

**Sistemas de Informação**

Algoritmos e Técnicas de Programação

**Roteiro de Exercícios 10**

Nome: Victor Magalhães de Souza - 1266914

Professor: Humberto Nigri

Exercício 1

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]M)

{

for (int l = 0; l < M.GetLength(0); l++){

for (int c = 0; c < M.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

M[l, c] = int.Parse(Console.ReadLine());

}

}

}

// Função que imprime matriz

static void ImprimeMatriz(int [,]M)

{

for(int l = 0; l < M.GetLength(0); l++){

for (int c = 0; c < M.GetLength(1); c++){

Console.Write(M[l,c] + "\t");

}

Console.Write("\n");

}

}

Exercício 2

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Função que vê maior número

static void MaiorElemento(int [,]matriz, string nomeMatriz)

{

int MaiorElemento = 0, linhaMaior = 0, colunaMaior = 0;

for(int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

if (l == 0 && c == 0){

MaiorElemento = matriz[l,c];

linhaMaior = l;

colunaMaior = c;

}

else{

if (matriz[l,c] > MaiorElemento){

MaiorElemento = matriz[l,c];

linhaMaior = l;

colunaMaior = c;

}

}

}

}

Console.Write("O maior elemento da matriz {0} foi o {1} na posição {2}:{3}", nomeMatriz, MaiorElemento, linhaMaior, colunaMaior);

Console.Write("\n");

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

int [ , ] matrizB = new int [4,4];

LeMatriz(matrizA);

LeMatriz(matrizB);

MaiorElemento(matrizA, "A");

MaiorElemento(matrizB, "B");

}

}

Exercício 3

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Soma as matrizes

static void SomaMatriz(int[,]matrizA, int[,]matrizB, int[,]matrizC)

{

for (int l = 0; l < matrizA.GetLength(0); l++){

for (int c = 0; c < matrizA.GetLength(1); c++){

matrizC[l,c] = matrizA[l,c] + matrizB[l,c];

}

}

}

// Função que imprime matriz

static void ImprimeMatriz(int [,]M)

{

for(int l = 0; l < M.GetLength(0); l++){

for (int c = 0; c < M.GetLength(1); c++){

Console.Write(M[l,c] + "\t");

}

Console.Write("\n");

}

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

int [ , ] matrizB = new int [4,4];

int [ , ] matrizC = new int [4,4];

LeMatriz(matrizA);

LeMatriz(matrizB);

SomaMatriz(matrizA, matrizB, matrizC);

Console.WriteLine("Soma das Matrizes A e B");

ImprimeMatriz(matrizC);

}

}

Exercício 4

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Média matriz

static void MediaMatriz(int[,]matriz, string nomeMatriz)

{

int soma = 0;

double media;

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

soma = soma + matriz[l,c];

}

}

media = soma / (matriz.GetLength(0) \* matriz.GetLength(1));

Console.Write("A média da matriz {0} é {1}", nomeMatriz, media);

Console.Write("\n");

}

// Função que imprime matriz

static void ImprimeMatriz(int [,]M)

{

for(int l = 0; l < M.GetLength(0); l++){

for (int c = 0; c < M.GetLength(1); c++){

Console.Write(M[l,c] + "\t");

}

Console.Write("\n");

}

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

int [ , ] matrizB = new int [4,4];

LeMatriz(matrizA);

LeMatriz(matrizB);

MediaMatriz(matrizA, "A");

MediaMatriz(matrizB, "B");

}

}

Exercício 5

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Soma as matrizes

static void SomaMatriz(int[,]matrizA, int[,]matrizB, int[,]matrizC)

{

for (int l = 0; l < matrizA.GetLength(0); l++){

for (int c = 0; c < matrizA.GetLength(1); c++){

matrizC[l,c] = matrizA[l,c] + matrizB[l,c];

}

}

}

// Soma das diagonais (principal e secundária)

static void SomaDiagonal(int[,] matrizC)

{

int somaPrincipal = 0, somaSecundaria = 0, c = 0;

// Principal

for(int l = 0; l < matrizC.GetLength(0); l++){

for (c = 0; c < matrizC.GetLength(1); c++){

if(l == c){

somaPrincipal += matrizC[l,c];

}

}

}

// Secundaria

c = matrizC.GetLength(1) - 1;

for(int l = 0; l < matrizC.GetLength(0); l++){

somaSecundaria += matrizC[l,c];

c--;

}

Console.Write("\nA soma da diagonal principal é {0}\n", somaPrincipal);

Console.Write("A soma da diagonal secundária é {0}\n", somaSecundaria);

}

// Função que imprime matriz

static void ImprimeMatriz(int [,]M)

{

for(int l = 0; l < M.GetLength(0); l++){

for (int c = 0; c < M.GetLength(1); c++){

Console.Write(M[l,c] + "\t");

}

Console.Write("\n");

}

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

int [ , ] matrizB = new int [4,4];

int [ , ] matrizC = new int [4,4];

LeMatriz(matrizA);

LeMatriz(matrizB);

SomaMatriz(matrizA, matrizB, matrizC);

Console.WriteLine("Soma das Matrizes A e B");

ImprimeMatriz(matrizC);

SomaDiagonal(matrizC);

}

}

Exercício 6

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Soma as matrizes

static void SomaMatriz(int[,]matrizA, int[,]matrizB, int[,]matrizC)

{

for (int l = 0; l < matrizA.GetLength(0); l++){

for (int c = 0; c < matrizA.GetLength(1); c++){

matrizC[l,c] = matrizA[l,c] + matrizB[l,c];

}

}

}

// Soma das diagonais (principal e secundária)

static void SomaDiagonal(int[,] matrizC)

{

int somaPrincipal = 0, somaSecundaria = 0, c = 0;

// Principal

for(int l = 0; l < matrizC.GetLength(0); l++){

for (c = 0; c < matrizC.GetLength(1); c++){

if(l == c){

somaPrincipal += matrizC[l,c];

}

}

}

// Secundaria

c = matrizC.GetLength(1) - 1;

for(int l = 0; l < matrizC.GetLength(0); l++){

somaSecundaria += matrizC[l,c];

c--;

}

Console.Write("\nA soma da diagonal principal é {0}\n", somaPrincipal);

Console.Write("A soma da diagonal secundária é {0}\n", somaSecundaria);

}

// Função que imprime matriz

static void ImprimeMatriz(int [,]M)

{

for(int l = 0; l < M.GetLength(0); l++){

for (int c = 0; c < M.GetLength(1); c++){

Console.Write(M[l,c] + "\t"); }

Console.Write("\n");

}

}

// Função que imprime matriz invertida

static void ImprimeMatrizInvertida(int [,]matrizC)

{

int [,] matrizD = new int [matrizC.GetLength(0), matrizC.GetLength(1)];

for(int l = 0; l < matrizC.GetLength(0); l++){

for (int c = 0; c < matrizC.GetLength(1); c++){

matrizD[c,l] = matrizC[l,c];

}

}

ImprimeMatriz(matrizD);

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

int [ , ] matrizB = new int [4,4];

int [ , ] matrizC = new int [4,4];

LeMatriz(matrizA);

LeMatriz(matrizB);

SomaMatriz(matrizA, matrizB, matrizC);

Console.WriteLine("Soma das Matrizes A e B");

ImprimeMatriz(matrizC);

SomaDiagonal(matrizC);

Console.WriteLine("Matriz C transposta");

ImprimeMatrizInvertida(matrizC);

}

}

Exercício 7

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Soma as matrizes

static void SomaMatriz(int[,]matrizA, int[,]matrizB, int[,]matrizC)

{

for (int l = 0; l < matrizA.GetLength(0); l++){

for (int c = 0; c < matrizA.GetLength(1); c++){

matrizC[l,c] = matrizA[l,c] + matrizB[l,c];

}

}

}

// Soma das diagonais (principal e secundária)

static void SomaDiagonal(int[,] matrizC)

{

int somaPrincipal = 0, somaSecundaria = 0, c = 0;

// Principal

for(int l = 0; l < matrizC.GetLength(0); l++){

for (c = 0; c < matrizC.GetLength(1); c++){

if(l == c){

somaPrincipal += matrizC[l,c];

}

}

}

// Secundaria

c = matrizC.GetLength(1) - 1;

for(int l = 0; l < matrizC.GetLength(0); l++){

somaSecundaria += matrizC[l,c];

c--;

}

Console.Write("\nA soma da diagonal principal é {0}\n", somaPrincipal);

Console.Write("A soma da diagonal secundária é {0}\n", somaSecundaria);

}

// Função que imprime matriz

static void ImprimeMatriz(int [,]M)

{

for(int l = 0; l < M.GetLength(0); l++){

for (int c = 0; c < M.GetLength(1); c++){

Console.Write(M[l,c] + "\t");

}

Console.Write("\n");

}

}

// Função que imprime matriz invertida

static void ImprimeMatrizInvertida(int [,]matrizC)

{

int [,] matrizD = new int [matrizC.GetLength(0), matrizC.GetLength(1)];

for(int l = 0; l < matrizC.GetLength(0); l++){

for (int c = 0; c < matrizC.GetLength(1); c++){

matrizD[c,l] = matrizC[l,c];

}

}

ImprimeMatriz(matrizD);

}

// Calculo da matriz E

static void MenorE(int[,] matrizA, int[,] matrizB)

{

int [ , ] matrizE = new int [4,4];

for(int l = 0; l < matrizA.GetLength(0); l++){

for(int c = 0; c < matrizB.GetLength(1); c++){

if(matrizA[l,c] < matrizB[l,c]){

matrizE[l,c] = matrizA[l,c];

}

else if(matrizA[l,c] > matrizB[l,c]){

matrizE[l,c] = matrizA[l,c];

}

else{

matrizE[l,c] = matrizA[l,c];

}

}

}

ImprimeMatriz(matrizE);

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

int [ , ] matrizB = new int [4,4];

int [ , ] matrizC = new int [4,4];

LeMatriz(matrizA);

LeMatriz(matrizB);

SomaMatriz(matrizA, matrizB, matrizC);

Console.WriteLine("Soma das Matrizes A e B");

ImprimeMatriz(matrizC);

SomaDiagonal(matrizC);

Console.WriteLine("\nMatriz C transposta");

ImprimeMatrizInvertida(matrizC);

Console.WriteLine("\nMatriz E (menor entre Matriz A e Matriz B)");

MenorE(matrizA, matrizB);

}

}

Exercício 8

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Soma das diagonais (principal e secundária)

static void ImprimePrincipal(int[,] matriz)

{

for(int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

if(l == c){

for(int a = 0; a <= c; a++){

Console.Write(matriz[l,c] + "\t");

}

}

}

Console.Write("\n");

}

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

LeMatriz(matrizA);

ImprimePrincipal(matrizA);

}

}

Exercício 9

class Program

{

// Função que lê matriz

static void LeMatriz(int [,]matriz)

{

for (int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

Console.Write("Posição {0}:{1} -> ", l+1, c+1);

matriz[l, c] = int.Parse(Console.ReadLine());

}

}

Console.Write("\n");

}

// Soma das diagonais (principal e secundária)

static void ImprimePrincipa l(int[,] matriz)

{

for(int l = 0; l < matriz.GetLength(0); l++){

for (int c = 0; c < matriz.GetLength(1); c++){

if(c >= l){

Console.Write(matriz[l,c] + "\t");

}

else{

Console.Write('\t');

}

}

Console.Write("\n");

}

}

static void Main(string[] args)

{

int [ , ] matrizA = new int [4,4];

LeMatriz(matrizA);

ImprimePrincipal(matrizA);

}

}