

Atividade Matriz

Disciplina: Algoritmos II
Professora: Daniela Scherer dos Santos

Semestre Letivo: 2014/2
Data de entrega: 27/08/2014

1. Escreva um programa para ler as notas de uma turma com 30 alunos considerando 3 notas por aluno. Utilizar uma matriz para armazenar estes valores, considerando que em cada linha da matriz estarão armazenadas as 3 notas de cada aluno. O algoritmo deverá calcular e mostrar a média aritmética de cada aluno e a média geral da turma. No final mostrar a matriz.

```
static void Main(string[] args)
{
    const int LINHAS = 4;
    const int COLUNAS = 4;
    double[,] matriz = new double[LINHAS, COLUNAS];
    int x, y;
    double somaNotas, mediaTurma = 0;
    //leitura da matriz
    for (x = 0; x < LINHAS; x++)
    {
        Console.WriteLine("Informe as {0} notas do aluno {1}", COLUNAS - 1, x);
        somaNotas = 0;
        for (y = 0; y < COLUNAS; y++)
        {
            if (y < COLUNAS - 1)
            {
                matriz[x, y] = Convert.ToDouble(Console.ReadLine());
                somaNotas = somaNotas + matriz[x, y];
            }
            else
            {
                matriz[x, y] = somaNotas / (COLUNAS - 1);
                mediaTurma = mediaTurma + matriz[x, y];
            }
        }
    }
    //mostra a matriz
    for (x = 0; x < LINHAS; x++)
    {
        for (y = 0; y < COLUNAS; y++)
        {
            Console.Write(matriz[x, y] + " ");
        }
        Console.WriteLine();
    }
    Console.WriteLine("A média da turma é " + mediaTurma / LINHAS);
    Console.ReadKey();
}
```

2. Faça um programa para ler uma matriz M (5 x 5) e criar dois vetores SL e SC que contenham, respectivamente, as somas das linhas e das colunas de M. No final o programa deve escrever os vetores e a matriz.

//funciona apenas para matrizes quadradas
static void Main(string[] args)

```
{
    const int LINHAS = 4;
    const int COLUNAS = 4;
    int[,] matriz = new int[LINHAS, COLUNAS];
    int[] sL = new int[LINHAS];
    int[] sC = new int[COLUNAS];
    int x, y, somaLinhas, somaColunas;
```

```

//leitura da matriz
Console.WriteLine("Informe os {0} valores da matriz", COLUNAS*LINHAS);
for (x = 0; x < LINHAS; x++)
{
    for (y = 0; y < COLUNAS; y++)
    {
        matriz[x, y] = Convert.ToInt16(Console.ReadLine());
    }
}
//gera vetor sL e o vetor sC
for (x = 0; x < LINHAS; x++)
{
    somaLinhas = 0;
    somaColunas = 0;
    for (y = 0; y < COLUNAS; y++)
    {
        somaLinhas = somaLinhas + matriz[x, y];
        somaColunas = somaColunas + matriz[y, x];
    }
    sL[x] = somaLinhas;
    sC[x] = somaColunas;
}
//imprime os valores e as matrizes
Console.WriteLine("Matriz:");
for (x = 0; x < LINHAS; x++)
{
    for (y = 0; y < COLUNAS; y++)
    {
        Console.Write(matriz[x, y] + "\t");
    }
    Console.WriteLine();
}
Console.WriteLine("Vetor com a soma das linhas:");
for (x = 0; x < LINHAS; x++)
    Console.Write(sL[x] + "\t");
Console.WriteLine();
Console.WriteLine("Vetor com a soma das colunas:");
for (x = 0; x < COLUNAS; x++)
    Console.Write(sC[x] + "\t");
Console.ReadKey();
}

```

```

//funciona para qualquer tamanho de matriz
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            const int LINHAS = 4;
            const int COLUNAS = 5;
            int[,] matriz = new int[LINHAS, COLUNAS];
            int[] sL = new int[LINHAS];
            int[] sC = new int[COLUNAS];
            int x, y, somaLinhas, somaColunas;

            //leitura da matriz
            Console.WriteLine("Informe os {0} valores da matriz", COLUNAS * LINHAS);
            for (x = 0; x < LINHAS; x++)
            {
                for (y = 0; y < COLUNAS; y++)

```

```

        {
            matriz[x, y] = Convert.ToInt16(Console.ReadLine());
        }
    }
    //gera vetor sL
    for (x = 0; x < LINHAS; x++)
    {
        somaLinhas = 0;
        for (y = 0; y < COLUNAS; y++)
        {
            somaLinhas = somaLinhas + matriz[x, y];
        }
        sL[x] = somaLinhas;
    }
    //gera vetor sC
    for (x = 0; x < COLUNAS; x++)
    {
        somaColunas = 0;
        for (y = 0; y < LINHAS; y++)
        {
            somaColunas = somaColunas + matriz[y, x];
        }
        sC[x] = somaColunas;
    }
    //imprime os valores e as matrizes
    Console.WriteLine("Matriz:");
    for (x = 0; x < LINHAS; x++)
    {
        for (y = 0; y < COLUNAS; y++)
        {
            Console.Write(matriz[x, y] + "t");
        }
        Console.WriteLine();
    }
    Console.WriteLine("Vetor com a soma das linhas:");
    for (x = 0; x < LINHAS; x++)
        Console.Write(sL[x] + "t");
    Console.WriteLine();
    Console.WriteLine("Vetor com a soma das colunas:");
    for (x = 0; x < COLUNAS; x++)
        Console.Write(sC[x] + "t");
    Console.ReadKey();
}
}
}

```

3. Escreva um programa que leia um vetor de 25 elementos inteiros e, a partir desse vetor, crie uma matriz 5 x 5 preenchida da seguinte forma: os cinco primeiros elementos do vetor formarão a primeira linha, os próximos 5 elementos formarão a segunda linha e assim por diante. Imprimir o vetor e a matriz no final.

```

static void Main(string[] args)
{
    const int LINHAS = 3;
    const int COLUNAS = 3;
    int[,] matriz = new int[LINHAS, COLUNAS];
    int[] vetor = new int[LINHAS*COLUNAS];
    int x, y, indiceVetor;

    //leitura do vetor
    Console.WriteLine("Informe os {0} valores do vetor", COLUNAS * LINHAS);
    for (x = 0; x < LINHAS*COLUNAS; x++)
    {
        vetor[x] = Convert.ToInt16(Console.ReadLine());
    }
    //gera matriz
    indiceVetor = 0;
    for (x = 0; x < LINHAS; x++)

```

```

{
    for (y = 0; y < COLUNAS; y++)
    {
        matriz[x, y] = vetor[indiceVetor];
        indiceVetor++;
    }
}
//mostra matriz gerada
for (x = 0; x < LINHAS; x++)
{
    for (y = 0; y < COLUNAS; y++)
    {
        Console.Write(matriz[x, y] + "\t");
    }
    Console.WriteLine();
}

Console.ReadKey();
}

```

4. Escreva um programa que lê uma matriz A e a escreve. A seguir o programa deve verificar quais os elementos de A que estão repetidos e quantas vezes cada um está repetido. Escrever cada elemento repetido com uma mensagem dizendo que o elemento aparece x vezes em A.

```

static void Main(string[] args)
{
    const int LINHAS = 3;
    const int COLUNAS = 3;
    int[,] matriz = new int[LINHAS, COLUNAS];
    int[] valores = new int[LINHAS * COLUNAS];
    int[] ocorrencias = new int[LINHAS * COLUNAS];

    int x, y, indiceVetor, i, achou, cont, w, z;

    //leitura da matriz
    Console.WriteLine("Informe os {0} valores da matriz", COLUNAS * LINHAS);
    for (x = 0; x < LINHAS; x++)
    {
        for (y = 0; y < COLUNAS; y++)
        {
            matriz[x, y] = Convert.ToInt16(Console.ReadLine());
        }
    }
    //mostra matriz
    for (x = 0; x < LINHAS; x++)
    {
        for (y = 0; y < COLUNAS; y++)
        {
            Console.Write(matriz[x, y] + "\t");
        }
        Console.WriteLine();
    }
    /*para cada valor armazenado na matriz, o algoritmo
    * verifica se este valor já encontra-se no vetor de
    * valores. Se não estiver, armazena no vetor "valores"
    * o valor e no vetor "ocorrencias" a quantidade de
    * vezes que se repete. */
    indiceVetor = 0;
    cont = 0;
    achou = 0;
    for (x = 0; x < LINHAS; x++)
    {
        for (y = 0; y < COLUNAS; y++)
        {
            i = 0;

```

```

while (i < indiceVetor)//laço de repetição para percorrer o vetor "valores"
{
    if (matriz[x,y] == valores[i])//se o valor da matriz é igual ao valor armazenado na posição i do vetor
    {
        achou = 1;//indicando que o valor já encontra-se no vetor de valores
        i = indiceVetor;//para encerrar o laço de repetição pois o valor já foi encontrado
    }
    i++;
}
//fim do while
if (achou != 1)//se o elemento ainda não está no vetor de valores, procura ele dentro do restante da
matriz
{
    for (w = 0; w < LINHAS; w++)
    {
        for (z = 0; z < COLUNAS; z++)
        {
            if (matriz[x,y] == matriz[w,z])
            {
                cont++;//incrementa o contados do numero de repetições
            }
        }
    }
    valores[indiceVetor] = matriz[x,y];//armazena o valor no vetor valores
    ocorrencias[indiceVetor] = cont;//armazena no vetor ocorrencias a quantidade de vezes que o valor
aparece na matriz*/
    indiceVetor++;
}
achou = 0;
cont = 0;
}
}
//mostra vetor de valores e ocorrencias
for (x = 0; x < indiceVetor; x++)
{
    Console.WriteLine("O valor " + valores[x] + " aparece " + ocorrencias[x] + " vezes na matriz");
}
Console.ReadKey();
}

```

Observações sobre a solução do exercício 4:

- a lógica utilizada para a construção da solução do exercício 4 foi a seguinte:
 - o algoritmo utiliza 3 estruturas:
 - uma matriz "matriz" (onde os valores estão armazenados);
 - um vetor "valores" para armazenar todos os valores que constam na matriz;
 - um vetor "ocorrencias" para armazenar a quantidade de vezes que cada valor armazenado em "valores" aparece na matriz;
- No exemplo, o valor 50 encontra-se armazenado na posição 0 do vetor valores e o total de vezes que ele aparece na matriz encontra-se armazenado no vetor ocorrencias, também na posição 0.
- Exemplo:

	0	1	2	
0	50	45	50	➔ MATRIZ A
1	12	18	45	
2	50	12	50	

50	45	12	18						➔ VETOR valores
0	1	2	3	4	5	6	7	8	

4	2	2	1						➔ VETOR ocorrencias
0	1	2	3	4	5	6	7	8	