**FEMA/IMESA – Análise e Desenvolvimento de Sistemas**

**Algoritmos e Estruturas de Dados I**

**Matriz – Revisão**

1. Escreva um algoritmo que leia uma matriz de 4x4, do tipo inteiro. A seguir transforme a mesma matriz numa matriz binária, da seguinte forma:
2. A cada elemento **par** que encontrar, grave 0 em seu lugar;
3. A cada elemento **ímpar** que encontrar, grave 1 em seu lugar.
4. Mostre a matriz gerada.

Algoritmo um;

Var

m[4][4], q, r, i, j: inteiro;

Início

Para (i = 0; i < 4; i = i +1)

Para (j = 0; j < 4; j = j + 1)

Imprima “Digite o elemento: “;

Leia m[i][j];

q ← m[i][j] / 2;

r ← m[i][j] – q \* 2;

Se (r = 0)

Então m[i][j] ← 0;

Senão m[i][j] ← 1;

Imprima “A matriz resultante: “;

Para (i = 0; i < 4; i = i +1)

Para (j = 0; j < 4; j = j + 1)

Imprima m[i][j];

Fim.

**Outra Solução:**

Algoritmo um\_um;

Var

m[4][4], q, r, i, j: inteiro;

Início

Para (i = 0; i < 4; i = i +1)

Para (j = 0; j < 4; j = j + 1)

Imprima “Digite o elemento: “;

Leia m[i][j];

Para (i = 0; i < 4; i = i +1)

Para (j = 0; j < 4; j = j + 1)

q ← m[i][j] / 2;

r ← m[i][j] – q \* 2;

Se (r = 0)

Então m[i][j] ← 0;

Senão m[i][j] ← 1;

Imprima “A matriz resultante: “;

Para (i = 0; i < 4; i = i +1)

Para (j = 0; j < 4; j = j + 1)

Imprima m[i][j];

Fim.

1. Leia o programa abaixo e escreva o Teste de Mesa mostrando, passo a passo, o que ocorrerá após a execução dos dois laços:

#define t 4

main(){

int i, j, s=0, a[t][t] = { 3, 5, 9, 4,

7, 6, 3, 8,

12, 8, 2, 9,

-7, 0, 4, 7};

a[3][0] = 0;

for (i=0; i<t-1; i++){

for (j=0; j<t; j++){

if (a[i][j] % 3 == 0){

a[3][0] = a[3][0] + a[i][j];

}

}

}

cout << "O resultado: " << a[3][0];

}

1. Faça um algoritmo que leia uma matriz M de 9 elementos, inteiros, e a seguir calcule e mostre:
2. A quantidade de elementos pares na diagonal principal (i = j);
3. A soma entre os ímpares na diagonal secundária (i + j = T – 1);
4. Os elementos que estão acima da diagonal principal (i < j).

Algoritmo três;

Var

m[3][3], q, r, i, j, cont, s: inteiro;

Início

Para (i = 0; i < 3; i = i +1)

Para (j = 0; j < 3; j = j + 1)

Imprima “Digite o elemento: “;

Leia m[i][j];

cont ← 0; s ← 0;

Para (i = 0; i < 3; i = i +1)

Para (j = 0; j < 3; j = j + 1)

q ← m[i][j] / 2;

r ← m[i][j] – q \* 2;

Se (r = 0)

Então Se (i = j) //(a)

Então cont ← cont + 1;

Senão Se (i + j = 2) //(b)

Então s ← s + m[i][j];

Se (i < j) //(c)

Então imprima “Elemento está acima da DP: “, m[i][j];

Imprima “A quantidade de elementos pares na diagonal principal: “, cont;

Imprima “A soma entre os ímpares na diagonal secundária: “, s;

Fim.

1. Escreva um algoritmo que leia uma matriz de 2x3, do tipo inteiro, calcule e mostre:
2. O menor elemento da matriz;
3. O maior elemento da matriz;
4. A média entre os elementos >= 5.

Algoritmo quatro;

Var

a[3][3], i, j, maior, s, menor, c: inteiro;

Início

//Leitura da Matriz

Para (i = 0; i < 2; i = i +1)

Para (j = 0; j < 3; j = j + 1)

Imprima “Digite o elemento: “;

Leia a[i][j];

//Maior elemento da Matriz

maior ← a[0][0];

Para (i = 0; i < 2; i = i +1)

Para (j = 0; j < 3; j = j + 1)

Se (a[i][j] > maior)

Então maior ← a[i][j];

Imprima “O maior elemento: “, maior;

//Menor elemento da Matriz

menor ← a[0][0];

Para (i = 0; i < 2; i = i +1)

Para (j = 0; j < 3; j = j + 1)

Se (a[i][j] < menor)

Então menor ← a[i][j];

Imprima “O menor elemento: “, menor;

//Média entre os elementos >= 5

s ← 0; c ← 0;

Para (i = 0; i < 2; i = i +1)

Para (j = 0; j < 3; j = j + 1)

Se (a[i][j] >= 5)

Então s ← s + a[i][j];

c ← c + 1;

Se (c = 0)

Então Imprima “Não foram encontrados elementos >= 5”;

Senão Imprima “A média entre os elementos >= 5: “, s / c;

Fim.

1. Leia o programa abaixo e escreva o *Teste de Mesa* mostrando, passo a passo, o que ocorrerá após a execução do programa:

#define t 4

main(){

int i, j, s;

int a[t][t] = **{** 8, 11, 0, 20,

12, 14, 9, 30,

18, 6, 10, 40,

5, 20, 3, 1**}**;

for(j = 0; j < t; j++)**{**

s = 0;

for(i = 0; i < t; i++)**{**

if(a[i][j] > 10)**{**

s += a[i][j];**}**

**}**

cout << "\nO valor é: " << s;

**}**

**}**