**Exercícios aplicando matrizes**

**Soluções**

1. Informe como ficará carregada a matriz ***m*** após execução do código a seguir (executar manualmente, sem uso de compilador!):

int m[5][4];

int i, j;

for(i = 0; i < 5; i++){

for(j = 0; j < 4; j++)

m[i][j] = i + j;

}

**0, 1, 2, 3,**

**1, 2, 3, 4,**

**2, 3, 4, 5,**

**3, 4, 5, 6,**

**4, 5, 6, 7,**

int m[5][4];

int i, j;

for(i = 0; i < 5; i++){

for(j = 0; j < 4; j++)

m[i][j] = 4\*i + j + 1;

}

**1, 2, 3, 4,**

**5, 6, 7, 8,**

**9, 10, 11, 12,**

**13, 14, 15, 16,**

**17, 18, 19, 20,**

int m[5][4];

int i, j;

for(i = 0; i < 5; i++){

for(j = 0; j < 4; j++)

m[i][j] = i \* (j % 2);

}

**0, 0, 0, 0,**

**0, 1, 0, 1,**

**0, 2, 0, 2,**

**0, 3, 0, 3,**

**0, 4, 0, 4,**

int m[3][6];

for(i = 0; i < 3; i++){

for(j = 0; j < 6; j++)

m[i][j] = 6\*i + j;

}

**0, 1, 2, 3, 4, 5,**

**6, 7, 8, 9, 10, 11,**

**12, 13, 14, 15, 16, 17,**

int m[5][5];

int i, j;

for(i = 0; i < 5; i++){

for(j = 0; j < 5; j++)

if (i == 0)

m[i][j] = j + 1;

else

m[i][j] = i + (j % i);

}

**1, 2, 3, 4, 5,**

**1, 1, 1, 1, 1,**

**2, 3, 2, 3, 2,**

**3, 4, 5, 3, 4,**

**4, 5, 6, 7, 4,**

1. Faça um programa em C que carregue uma matriz de [6x6] com valores reais aleatórios e depois calcule e mostre a soma da diagonal principal (coluna = linha) e da diagonal secundária (coluna = N – linha – 1) da matriz.

**#include<stdlib.h>**

**#include<stdio.h>**

**#include<time.h>**

**void main (){**

**int mat[6][6];**

**int i, j, soma;**

**srand(time(NULL));**

**// carrega a matriz mat com nr entre -10 e +10**

**for(i = 0; i < 6; i++){**

**for(j = 0; j < 6; j++)**

**mat[i][j] = (rand()%21) - 10;**

**}**

**// mostra o conteúdo de mat**

**printf("\n\n Conteúdo da matriz mat:\n");**

**for(i = 0; i < 6; i++){**

**for(j = 0; j < 6; j++){**

**printf("%3d, ", mat[i][j]);**

**}**

**printf("\n");**

**}**

**// calcula e mostra a soma das diagonais**

**int somaDP = 0, somaDS = 0;**

**for(i = 0; i < 6; i++){**

**somaDP += mat[i][i];**

**somaDS += mat[i][5-i];**

**}**

**printf("\n\n Soma da diagonal principal : %d.", somaDP);**

**printf("\n\n Soma da diagonal secundaria: %d.", somaDS);**

**}**

1. A soma de duas matrizes de mesma dimensão [NxM] é outra matriz também [NxM], onde cada elemento da matriz resultante é igual a soma dos elementos correspondentes nas duas matrizes. Faça um programa em C que calcule e mostre a matriz ***r*** resultante da soma de duas matrizes ***a*** e ***b***, de ordem [8x12], carregadas inicialmente com valores inteiros aleatórios.

**#include<stdlib.h>**

**#include<stdio.h>**

**#include<time.h>**

**void main (){**

**int a[8][12], b[8][12], r[8][12];**

**int i, j;**

**srand(time(NULL));**

**// carrega as matrizes a e b com nr entre 0 e 5**

**for(i = 0; i < 8; i++){**

**for(j = 0; j < 12; j++){**

**a[i][j] = rand()%6;**

**b[i][j] = rand()%6;**

**}**

**}**

**// calcula a matriz r**

**for(i = 0; i < 8; i++){**

**for(j = 0; j < 12; j++)**

**r[i][j] = a[i][j]+b[i][j];**

**}**

**// mostra as três matrizes**

**printf("\n\n");**

**for(i = 0; i < 8; i++){**

**for(j = 0; j < 12; j++)**

**printf("%3d ", a[i][j]);**

**printf("\n");**

**}**

**printf("\n\n");**

**for(i = 0; i < 8; i++){**

**for(j = 0; j < 12; j++)**

**printf("%3d ", b[i][j]);**

**printf("\n");**

**}**

**printf("\n\n");**

**for(i = 0; i < 8; i++){**

**for(j = 0; j < 12; j++)**

**printf("%3d ", r[i][j]);**

**printf("\n");**

**}**

**printf("\n\n ");**

**}**

1. Faça um programa em C que carregue uma matriz [6x8] com valores inteiros aleatórios entre 0 e 100, inclusive e depois mostre a quantidade de elementos com valores entre 10 e 20, inclusive, a soma dos elementos com valores ímpares e a média aritmética dos valores pares da matriz.

#include<stdlib.h>

#include<stdio.h>

#include<time.h>

void main (){

int matx[6][8];

int i, j, cont1020 = 0, somImpar = 0, somPar = 0, contPar = 0;

srand(time(NULL));

// CARREGA matx COM VALORES ENTRE 0 E 100

for(i = 0; i < 6; i++){

for(j = 0; j < 8; j++)

matx[i][j] = rand()%101;

}

// EFETUA OS CÁLCULOS ESTATÍSCOS

for(i = 0; i < 6; i++){

for(j = 0; j < 8; j++){

if(matx[i][j] >= 10 && matx[i][j] <= 20)

cont1020++;

if(matx[i][j]%2 == 0){

somPar += matx[i][j];

contPar++;

}

else

somImpar += matx[i][j];

}

}

// MOSTRA OS RESULTADOS

printf("\n\n CONTEUDO DE matx:\n");

for(i = 0; i < 6; i++){

for(j = 0; j < 8; j++)

printf("%3d ", matx[i][j]);

printf("\n");

}

printf("\n\tQuantidade de elementos entre 10 e 20: %d", cont1020);

printf("\n\tSoma dos elementos impares: %d", somImpar);

printf("\n\tMedia dos valores pares: %.2f\n", 1.0\*somPar/contPar);

}

1. Uma matriz identidade é uma matriz quadrada de ordem N, Mat[NxN], que atende a seguinte regra: Mat[lin, col] = 1 se lin = col e Mat[lin, col] = 0 se lin ≠ col. Faça um programa em C que leia pelo teclado o tamanho ***n*** de uma matriz quadrada e gere a matriz identidade correspondente.

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(){**

**int i, j, n;**

**printf("Informe o tamanho da matriz quadrada: ");**

**scanf("%d", &n);**

**int mat[n][n];**

**for(i = 0; i < n; i++)**

**for(j = 0; j < n; j++)**

**if (j == i)**

**mat[i][j] = 1;**

**else**

**mat[i][j] = 0;**

**printf("\n\n");**

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

**for(j = 0; j < n; j++)**

**printf("%d, ", mat[i][j]);**

**printf("\n");**

**}**

**printf("\n\n");**

**return 0;**

**}**

1. Faça um programa em C que, dada uma matriz quadrada de ordem 10 já carregada, verifique se esta é simétrica. Uma matriz quadrada M é simétrica se M[lin, col] = M[col, lin].

#include<stdio.h>

#include<stdlib.h>

#include<stdio.h>

#define TAM 10

int main(){

int i, j;

static int Matrix[TAM][TAM] =

{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

2, 3, 4, 5, 6, 7, 8, 9, 10, 11,

3, 4, 5, 6, 7, 8, 9, 10, 11, 12,

4, 5, 6, 7, 8, 9, 10, 11, 12, 13,

5, 6, 7, 8, 9, 10, 11, 12, 13, 14,

6, 7, 8, 9, 10, 11, 12, 13, 14, 15,

7, 8, 9, 10, 11, 12, 13, 14, 15, 16,

8, 9, 10, 11, 12, 13, 14, 15, 16, 17,

9, 10, 11, 12, 13, 14, 15, 16, 17, 18,

10, 11, 12, 13, 14, 15, 16, 17, 18, 19 };

/\* srand(time(NULL));

// inicia Matrix com valores aleatórios entre 0 e 1

for(i = 0; i < TAM; i ++)

for(j = 0; j < TAM; j++)

Matrix[i][j] = rand()%2;

\*/

**// VERIFICA SE Matrix É SIMÉTRICA**

int ehSimetr = 1;

for(i = 1; i < TAM; i ++){

for(j = 0; j < i; j++)

if(Matrix[i][j] != Matrix[j][i]){

ehSimetr = 0;

break;

}

}

printf("\n\n Conteúdo de Matrix:\n");

for(i = 0; i < TAM; i ++){

for(j = 0; j < TAM; j++)

printf("%2d, ", Matrix[i][j]);

printf("\n");

}

if (ehSimetr)

printf("\nMatrix e' simetrica.\n\n");

else

printf("\nMatrix nao e' simetrica.\n\n");

}

1. O produto matricial de uma matriz M1[IxJ] por outra matriz M2[JxK] é uma matriz M[IxK] (note que o tamanho de colunas de M1 tem que ser obrigatoriamente igual ao tamanho de linhas de M2). Cada elemento de M é calculado da seguinte forma:

M[i, k] =

Faça um programa em C que dadas as matrizes M1[6x5] e M2[5x8], já carregadas com valores inteiros aleatórios, calcule e mostre a matriz M[6x8] resultante do produto matricial de M1 x M2.

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define I 2

#define J 3

#define K 4

int main(){

int i, j, k;

static int M1[I][J], M2[J][K], M[I][K];

srand(time(NULL));

**// inicia a matriz M1 com valores aleatórios entre 0 e 4**

for(i = 0; i < I; i ++)

for(j = 0; j < J; j++)

M1[i][j] = rand()%5;

**// inicia a matriz M2 com valores aleatórios entre 0 e 4**

for(j = 0; j < J; j ++)

for(k = 0; k < K; k++)

M2[j][k] = rand()%5;

**// calcula M[i, k] = M1[i, j] X M2[j, k]**

for(i = 0; i < I; i ++)

for(k = 0; k < K; k++){

M[i][k] = 0;

for(j = 0; j < J; j++)

M[i][k] = M[i][k] + M1[i][j]\*M2[j][k];

}

**// print nas matrizes**

printf("\n\n Matriz M1:\n");

for(i = 0; i < I; i ++){

for(j = 0; j < J; j++)

printf("%2d, ", M1[i][j]);

printf("\n");

}

printf("\n\n Matriz M2:\n");

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

for(k = 0; k < K; k++)

printf("%2d, ", M2[j][k]);

printf("\n");

}

printf("\n\n Matriz M:\n");

for(i = 0; i < I; i ++){

for(k = 0; k < K; k++)

printf("%2d, ", M[i][k]);

printf("\n");

}

}

1. Denomina-se “ponto cela” de uma matriz a todo elemento da matriz que seja ao mesmo tempo o maior elemento de sua linha e também o maior elemento da sua coluna. Faça um programa em C em que, dada uma matriz [7x9], já carregada com valores inteiros aleatórios, determine e mostre a localização de todos os pontos de cela da matriz

#include<stdlib.h>

#include<stdio.h>

#include<time.h>

void main (){

int mat[7][9];

int lin, col, colMaior, i;

int achouMaiorCol;

srand(time(NULL));

**// CARREGA mat COM VALORES ENTRE 0 E 100**

for(lin = 0; lin < 7; lin++){

for(col = 0; col < 9; col++)

mat[lin][col] = rand()%101;

}

**// MOSTRA O CONTEÚDO DE mat**

printf("\n\n CONTEUDO DE mat:\n");

for(lin = 0; lin < 7; lin++){

for(col = 0; col < 9; col++)

printf("%3d ", mat[lin][col]);

printf("\n");

}

**// VERIFICA E MOSTRA OS PONTOS DE CELA.**

for(lin = 0; lin < 7; lin++){

**// ENCONTRA MAIOR DA LINHA**

colMaior = 0;

for(col = 1; col < 9; col++){

if(mat[lin][col] >= mat[lin][colMaior])

colMaior = col;

}

**// AQUI O MAIOR DA LINHA lin ESTA NA COLUNA colMaior**

**// PESQUISAR SE É TAMBÉM O MAIOR DESTA COLUNA**

achouMaiorCol = 0;

for(i = 0; i < 7; i++){

if(mat[i][colMaior] > mat[lin][colMaior]){

achouMaiorCol = 1;

break;

}

}

if( !achouMaiorCol )

printf("\n\t%d e' ponto de cela da linha %d e col %d.",

mat[lin][colMaior], lin, colMaior);

}

}