|  |  |
| --- | --- |
| Universidade Positivo  Tecnologia em Análise e Desenvolvimento de Sistemas  **Disciplina: Algoritmos de Programação**  **Referência: Profa. Fernanda Hembecker**  **Professor: Kristian Capeline** | Aula  **02** |

# Exercícios

* Matrizes

1. Realize o teste de mesa dos algoritmos a seguir, indique o valor das variáveis e o que será apresentado na tela:

**ALGORITMO Matriz1**

VAR

INTEIRO: mat[3][2] = {{7,6},{8,4},{1,10}};

INTEIRO: i, j;

INÍCIO

i🡨0;

j🡨1;

mat[i][j]🡨mat[i][j]+j;

mat[j][i]🡨mat[j][i]+mat[i][j];

mat[i][j-1]🡨mat[j][j];

mat[j+1][j]🡨i+j;

mat[j][j]🡨mat[i][j]\*mat[j][i];

PARA(i 🡨 0; i < 3; i 🡨 i + 1) FAÇA

PARA(j 🡨 0; j < 2; j 🡨 j + 1) FAÇA

ESCREVA(mat[i][j]);

FIM\_PARA

FIM\_PARA

FIM.

Resposta: 4 7

15 105

1 1

**ALGORITMO Matriz2**

VAR

INTEIRO: mat[3][5]= {{1, 7, 3, 4, 5},{2, 6, 5, 1, 0},{4, 9, 0, 3, 6}};

INTEIRO: vet[3], i, j;

INÍCIO

PARA(i 🡨 0; i < 3; i 🡨 i + 1) FAÇA

vet[i]🡨0;

PARA(j 🡨 0; j < 5; j 🡨 j + 1) FAÇA

vet[i]🡨vet[i]+ mat[i][j];

FIM\_PARA

FIM\_PARA

PARA(i 🡨 0; i < 3; i 🡨 i + 1) FAÇA

PARA(j 🡨 0; j < 5; j 🡨 j + 1) FAÇA

ESCREVA(mat[i][j]);

FIM\_PARA;

ESCREVA(vet[i]);

FIM\_PARA

FIM.

Resposta: 1 7 3 4 5 Vetor: 20

2 6 5 1 0 Vetor: 14

4 9 0 3 6 Vetor: 22

1. Elabore um algoritmo que leia uma matriz A(5 x 5) do teclado e crie uma matriz B onde cada elemento é o triplo do elemento correspondente de A.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int matA[5][5], matB[5][5];

int i, j;

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

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

printf("Digite os dados da matriz, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &matA[i][j]);

}

}

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

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

matB[i][j] = 3 \* matA[i][j];

}

}

printf("Matriz formada: \n");

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

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

printf("[%d][%d] = %d\t", i, j, matB[i][j]);

}

printf("\n");

}

}

1. Faça um algoritmo que leia uma matriz de 10 x 8 e apresente na tela o maior e o menor elemento da matriz.

#include <stdio.h>

#include <stdlib.h>

int main(){

int matA[10][8];

int i, j, maior, menor;

for (i = 0; i < 10; i++){

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

printf("Digite os dados da matriz, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &matA[i][j]);

if (i == 0 && j == 0){

maior = matA[0][0];

menor = matA[0][0];

}

if (matA[i][j] > maior){

maior = matA[i][j];

}

if (matA[i][j] < menor){

menor = matA[i][j];

}

}

}

printf("Matriz formada: \n");

for (i = 0; i < 10; i++){

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

printf("[%d][%d] = %d\t", i, j, matA[i][j]);

}

printf("\n");

}

printf("Maior: %d \n", maior);

printf("Menor: %d \n", menor);

}

1. Crie um algoritmo que leia uma matriz de dimensão (10x6) e apresente na tela a quantidade de elementos pares desta matriz.

#include <stdio.h>

#include <stdlib.h>

int main(){

int matA[5][2];

int i, j, par;

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

for (j = 0; j < 2; j++){

printf("Digite os dados da matriz, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &matA[i][j]);

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

par++;

}

}

}

printf("Matriz formada: \n");

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

for (j = 0; j < 2; j++){

printf("[%d][%d] = %d\t", i, j, matA[i][j]);

}

printf("\n");

}

printf("Quantidade de elementos pares: %d \n", par);

}

1. Faça um algoritmo que leia uma matriz A (5 x 5). Construa uma matriz B de mesma dimensão onde cada elemento de B deverá ser o dobro de cada elemento correspondente de A, exceto para os valores da diagonal principal (números 1 na matriz **de exemplo** abaixo), cujos valores deverão ser o triplo de cada elemento correspondente da matriz A.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 |

#include <stdio.h>

#include <stdlib.h>

int main(){

int matA[5][5], matB[5][5];

int i, j;

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

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

printf("Digite os dados da matriz, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &matA[i][j]);

if(i == j){

matB[i][j] = 3 \* matA[i][j];

}

else{

matB[i][j] = 2 \* matA[i][j];

}

}

}

printf("Matriz formada: \n");

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

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

printf("[%d][%d] = %d\t", i, j, matB[i][j]);

}

printf("\n");

}

}

1. Crie um algoritmo que leia uma matriz A (5 x 2) e, em um matriz B do mesmo tamanho, armazene a soma dos elementos de A na primeira coluna e a multiplicação dos elementos de A na segunda coluna.

#include <stdio.h>

#include <stdlib.h>

int main(){

int matA[5][2], matB[5][2];

int i, j, soma = 0, multi = 1;

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

for (j = 0; j < 2; j++){

printf("Digite os dados da matriz, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &matA[i][j]);

}

}

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

for (j = 0; j < 2; j++){

if (j == 0){

matB[i][j] = matA[i][j];

soma += matB[i][j];

}

if (j == 1) {

matB[i][j] = matA[i][j];

multi \*= matB[i][j];

}

}

}

printf("Resultado Matriz: \n");

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

for (j = 0; j < 2; j++){

printf("[%d][%d] = %d \t", i, j, matA[i][j]);

}

printf("\n");

}

printf("Soma dos elementos da primeira coluna: %d\n", soma);

printf("Multiplicacao dos elementos da segunda coluna: %d", multi);

}

1. Construa um algoritmo que leia do teclado uma matriz quadrada (12 x 12) e verifique se ela é uma matriz identidade. Uma matriz quadrada possui o mesmo número de linhas e colunas. Uma matriz identidade possui 1 nos elementos da diagonal principal e 0 nos demais. Exemplo: matriz identidade de dimensão 3x3.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 |

#include <stdio.h>

#include <stdlib.h>

int main(){

int i, j, lin, col;

int mat[i][j];

printf("Digite o numero de linhas da sua matriz: \n");

scanf("%d", &lin);

printf("Digite o numero de colunas da sua matriz: \n");

scanf("%d", &col);

if (lin != col) {

printf("Matriz nao eh uma matriz identidade!!\n");

}

else

{

printf("\nMatriz formada: \n");

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

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

if (i == j){

mat[i][j] = 1;

}

else{

mat[i][j] = 0;

}

printf("[%d][%d] = %d\t", i, j, mat[i][j]);

}

printf("\n");

}

}

}

1. Construa um algoritmo que leia uma matriz de dimensão 7 x 4 e, em um vetor de 7 elementos, armazene o menor elemento de cada linha da matriz. Exemplo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 | 5 | 4 | 2 | 8 |
| 1 | 3 | 6 | 9 | -1 |
| 2 | 0 | -3 | 4 | 7 |
| ... | ... | ... | ... | ... |
| 6 | 4 | 9 | 7 | 6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | -1 | -3 | ... | ... | ... | 4 |

#include <stdio.h>

#include <stdlib.h>

int main(){

int mat[7][4], vet[7];

int i, j, maior = 0, menor;

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

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

printf("Digite os dados da matriz, linha %d coluna %d: \n", i, j);

scanf("%d", &mat[i][j]);

}

}

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

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

if(j == 0){

menor = mat[i][j];

}

if(mat[i][j] > maior){

maior = mat[i][j];

}

if(mat[i][j] < menor){

menor = mat[i][j];

}

}

vet[i] = menor;

}

printf("Matriz formada: \n");

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

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

printf("[%d][%d] = %d\t", i, j, mat[i][j]);

}

printf("\n");

}

printf("Vetor formado: \n");

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

printf("%d; ", vet[i]);

}

}

1. Elabore um algoritmo que leia dois vetores A e B de 15 números do teclado. Após a leitura, crie uma matriz de 15 linhas e 2 colunas onde a primeira coluna armazena os elementos de A e a segunda coluna armazena os elementos de B. Apresente a matriz na tela. Exemplo:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | ... | 14 |
| 10 | 20 | 30 | 40 | 50 | ... | 150 |

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 0 | 10 | 5 |
| 1 | 20 | 15 |
| 2 | 30 | 25 |
| ... | ... | ... |
| 14 | 150 | 145 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | ... | 14 |
| 5 | 15 | 25 | 35 | 45 | ... | 145 |

#include <stdio.h>

#include <stdlib.h>

int main(){

int vetA[15], vetB[15], mat[15][2];

int i, j;

for (i = 0; i < 15; i++){

printf("Digite 15 numeros para preencher o primeiro vetor: \n");

printf("Valor %d de 15: \n", i+1);

scanf("%d", &vetA[i]);

}

for (i = 0; i < 15; i++){

printf("Digite 15 numeros para preencher o segundo vetor: \n");

printf("Valor %d de 15: \n", i+1);

scanf("%d", &vetB[i]);

}

printf("Vetor A: \n");

for (i = 0; i < 15; i++)

printf("%d; ", vetA[i]);

printf("\n");

printf("Vetor B: \n");

for (i = 0; i < 15; i++)

printf("%d; ", vetB[i]);

printf("\n");

printf("Matriz formada a partir dos dois vetores: \n");

for (i = 0; i < 15; i++){

for (j = 0; j < 2; j++){

printf("[%d][%d] = %d \t", i, j, vetA[i]);

printf("[%d][%d] = %d \t", i, j, vetB[i]);

printf("\n");

}

}

printf("\n");

}

1. Elabore um algoritmo que leia do teclado uma matriz triangular superior de dimensão 20x20. Uma matriz triangular superior é quadrada e possui 0 nos elementos abaixo da diagonal principal e qualquer outro valor nos elementos da diagonal principal e acima dela. Exemplo: matriz triangular superior de tamanho 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 | x | x | x | x |
| 1 | 0 | x | x | x |
| 2 | 0 | 0 | x | x |
| 3 | 0 | 0 | 0 | x |

#include <stdio.h>

#include <stdlib.h>

int main(){

int mat[20][20];

int i, j;

for (i = 0; i < 20; i++){

for (j = 0; j < 20; j++){

if (i == j || j >= i){

printf("[%d][%d] = x ", i, j);

}

else{

printf("[%d][%d] = 0 ", i, j);

}

}

printf("\n");

}

}

1. Elabore um algoritmo que leia uma matriz A dimensão 3x4 e crie uma matriz B que representa a transposta de A. Apresente na tela a matriz B. Exemplo:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| 0 | 1 | 5 | 9 |
| 1 | 2 | 6 | 10 |
| 2 | 3 | 7 | 11 |
| 3 | 4 | 8 | 12 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 5 | 6 | 7 | 8 |
| 2 | 9 | 10 | 11 | 12 |

#include <stdio.h>

#include <stdlib.h>

int main (){

int mat[3][4];

int i, j;

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

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

printf("Digite os dados da matriz, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &mat[i][j]);

}

}

printf("Matriz formada: \n");

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

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

printf("[%d][%d] = %d\t", i, j, mat[i][j]);

}

printf("\n");

}

printf("Matriz transposta: \n");

for (i = 0; i < 4; i++){

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

printf("[%d][%d] = %d\t", i, j, mat[j][i]);

}

printf("\n");

}

}

1. Construa um algoritmo que leia duas matrizes do teclado de dimensão 5 x 3 e realize a troca dos elementos destas matrizes.

#include <stdio.h>

#include <stdlib.h>

int main (){

int matA[5][3], matB[5][3];

int i, j, aux;

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

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

printf("Digite os dados da matriz A, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &matA[i][j]);

}

}

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

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

printf("Digite os dados da matriz B, linha %d coluna %d: \n", i+1, j+1);

scanf("%d", &matB[i][j]);

}

}

printf("Matriz A Original: \n");

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

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

printf("[%d][%d] = %d\t", i, j, matA[i][j]);

}

printf("\n");

}

printf("\n");

printf("Matriz B Original: \n");

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

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

printf("[%d][%d] = %d\t", i, j, matB[i][j]);

}

printf("\n");

}

printf("\n");

aux = matA[i][j];

matA[i][j] = matB[i][j];

matB[i][j] = aux;

printf("Matriz A Trocada: \n");

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

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

printf("[%d][%d] = %d\t", i, j, matB[i][j]);

}

printf("\n");

}

printf("\n");

printf("Matriz B Trocada: \n");

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

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

printf("[%d][%d] = %d\t", i, j, matA[i][j]);

}

printf("\n");

}

}

1. O tempo que um determinado avião leva para percorrer o trecho entre duas localidades distintas está disponível através da seguinte tabela:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| cidades | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | - | 02 | 11 | 06 | 15 | 11 | 01 |
| 1 | 02 | - | 07 | 12 | 04 | 02 | 15 |
| 2 | 11 | 07 | - | 11 | 08 | 03 | 13 |
| 3 | 06 | 12 | 11 | - | 10 | 02 | 01 |
| 4 | 15 | 04 | 08 | 10 | - | 05 | 13 |
| 5 | 11 | 02 | 03 | 02 | 05 | - | 14 |
| 6 | 01 | 15 | 13 | 01 | 13 | 14 | - |

1. Construa um algoritmo que leia a tabela acima e informe ao usuário o tempo necessário para percorrer duas cidades por ele fornecidas, até o momento em que ele fornecer duas cidades iguais (fonte e destino). Não deve ser lido informações para a diagonal principal.
2. Elabore um algoritmo que imprima a tabela sem repetições (apenas o triângulo superior ou o triângulo inferior).
3. Desenvolva um algoritmo que permita ao usuário informar várias cidades, até inserir a cidade “7”, e que imprima o tempo total para cumprir todo o percurso especificado entre as cidades fornecidas. Ou seja:
   * O usuário deve informar as cidades que ele quer cadastrar a distância.
   * Notem que é um percurso, então, por exemplo, o segundo trajeto inicia da cidade destino do primeiro trajeto.

Exemplo do “c”:

Digite cidade de origem: 1

Digite cidade de destino: 3

Digite o tempo entre a cidade 1 e 3: 10

Digite cidade de destino: 5

Digite o tempo entre a cidade 3 e 5: 21

Digite cidade de destino: 2

Digite o tempo entre a cidade 5 e 2: 39

Digite cidade de destino: 7

O tempo total do seu trajeto é: 70 minutos

Destino

Origem

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| cidades | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | - |  |  |  |  |  |  |
| 1 |  | - |  | 10 |  |  |  |
| 2 |  |  | - |  |  |  |  |
| 3 |  |  |  | - |  | 21 |  |
| 4 |  |  |  |  | - |  |  |
| 5 |  |  | 39 |  |  | - |  |
| 6 |  |  |  |  |  |  | - |

#include <stdio.h>

#include <stdlib.h>

int main(){

int mat[7][7] ={{0, 2, 11, 6, 15, 11, 1},{2, 0, 7, 12, 4, 2, 15},{11, 7, 0, 11, 8, 3, 13},{6, 12, 11, 0, 10, 2, 1},{15, 4, 8, 10, 0, 5, 13},{11, 2, 3, 2, 5, 0, 14}, {1, 15, 13, 1, 13, 14, 0}};

int i, j, origem, destino, destinoB, soma = 0, tempo;

printf("Digite uma cidade de origem: \n");

scanf("%d", &origem);

printf("Digite uma cidade de destino: \n");

scanf("%d", &destino);

printf("Digite o tempo entre a cidade %d e %d: ", origem, destino);

scanf("%d", &tempo);

soma += tempo;

printf("\n");

if (origem > 7 || destino > 7){

printf("Cidades invalidas!!! Tente novamente");

}

else{

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

for(j = 1; j < 7; j++){

if(j > i){

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

}

else{

printf(" \t");

}

}

printf("\n\n");

}

while (destinoB != 7){

printf("Digite uma cidade de destino: \n");

scanf("%d", &destinoB);

if (destinoB > 7){

printf("Destino invalido!!! Tente novamente");

printf("\n");

}

else{

if (destinoB == 7){

continue;

}

printf("Digite o tempo entre a cidade %d e %d: ", destino, destinoB);

scanf("%d", &tempo);

printf("\n");

soma += tempo;

}

}

printf("Tempo total de trajeto: %d\n", soma);

}

}