LISTA AEDS 6 – ESTRUTURAS HOMOGÊNEAS

EXERCÍCIO NÚMERO 1

#include <stdio.h>

int sum = 0; //sum= soma

int rep = 0, app = 0; //rep= reprovados, app= aprovados

void notas(int v[10], int n) {

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

printf("digite a nota do aluno %d: ",i+1); //i+1 é para nao começar no aluno 0

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

sum += v[i];

}

}

void media(int v[10],int n){

int me;

me = sum/n;

printf("\na media das notas dos alunos é: %d\n",me);

for(int j = 0; j < n; j++){ //isso aqui é para saber se a pessoa foi aprovada ou reprovada

if(v[j] < me) {

rep++;

} else {

app++;

}

}

}

int main (void){

int n=10; //n= numero alunos

int v[10];

//chama os dois procedimentos

notas(v,n);

media(v,n);

//coloca os app e os rep aqui, ao inves do procedimento, para ficar organziadao bonito no final do codigo

printf("\n%d alunos foram aprovados\n",app);

printf("%d alunos foram reprovados.\n",rep);

}

EXERCÍCIO NÚMERO 2

#include <stdio.h>

#include <stdlib.h>//biblioteca para o rand dar certo

void notas(int v[10]) {

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

v[i] = -1 \* rand();

}

}

int \*vetor(int v[]){

int x = 0;

int v2[10];

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

v2[i] = 0;

}

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

if (v[i] < 0){

v2[x] = v[i];

printf("numero %d\n", v2[x]);

x++;

}

}

return v2;

}

void printaVetor(int\* vet, int size){

for(int i = 0; i < size; i++){

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

}

}

int main(void) {

int vetor\_notas[10] = {1,2,-5,2,-1,0,-7,2,5,-9};

int\* vetorNotasNeg;

//notas(vetor\_notas);

vetorNotasNeg = vetor(vetor\_notas);

printaVetor(vetorNotasNeg, 10);

return 0;

}

EXERCÍCIO NÚMERO 3

#include <stdio.h>

#include <stdlib.h>

void vetor\_x\_y(int vx[10], int vy[10]) {

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

vx[i]=rand()%10;

vy[i]=rand()%10;

}

}

void novo\_veto(int v1[10], int v2[10], int v3[10]) {

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

if (v3[i]%2!=0) {

v3[i]=v1[i];

} else {

v3[i]=v2[i];

}

}

}

void imprimir\_vetor (int vet[10]) {

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

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

}

}

int main(void) {

int vetor1[10];

int vetor2[10];

int vetor3[10];

vetor\_x\_y(vetor1, vetor2);

novo\_veto(vetor1, vetor2, vetor3);

imprimir\_vetor(vetor3);

return 0;

}

EXERCÍCIO NÚMERO 4

#include <stdio.h>

int main(void) {

int temp[31] = {20, 24, 15, 17, 18, 19, 20, 25, 28, 29, 25, 34, 32, 39, 38, 34, 36, 17, 18, 34, 32, 40, 30, 35, 37, 23, 22, 21, 20, 31, 38};

int tmin=40, tmax=0, soma=0, inf = 0;

float media;

for (int i=0; i < 31; i++) {

if (temp[i] < tmin){

tmin = temp[i];

}

if (temp[i] > tmax){

tmax = temp[i];

}

soma += temp[i];

}

media = soma/31;

for (int i=0; i < 31; i++) {

if (temp[i] < media){

inf++;

}

}

printf("a maior temperatura foi de: %d\n", tmax);

printf("a menor temperatura foi de: %d\n", tmin);

printf("a media das temperaturas foi de: %.1f\n", media);

printf("número de dias que a temperatura foi menor que a média: %d\n", inf);

return 0;

}

EXERCÍCIO NÚMERO 5

#include <stdio.h>

int m = 5;

int M[5][5];

int sum = 0;

float soma\_linha\_4(float M[5][5]) {

sum = 0;

for (int j = 0; j < m; j++) {

sum += M[3][j]; //como é 0, 1, 2, 3, 4 ai tem que colocar 3 e nao 4, porque tem q contar o 0 tambem

}

return sum;

}

float soma\_coluna\_2(float M[5][5]) {

sum = 0;

for (int i = 0; i < m; i++) {

sum += M[1][i]; //como é 0, 1, 2, ai tem que colocar 1 e nao 2, porque tem q contar o 0 tambem

}

return sum;

}

float soma\_diagonal\_principal(float M[5][5]) {

sum = 0;

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

if (i == j)

sum += M[i][j];

}

}

return sum;

}

float soma\_diagonal\_secundaria(float M[5][5]) {

sum = 0;

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

if (i + j == 4)

sum += M[i][j];

}

}

return sum;

}

float soma\_todos\_os\_elemenos(float M[5][5]) {

sum = 0;

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

sum += M[i][j];

}

}

return sum;

}

int main(void) {

float M[m][m];

float somali, somacol, somadiag, somadiagsec, somatudo;

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

M[i][j] = rand() % 10;

printf("%.0f ", M[i][j]);

}

printf("\n");

}

float soma = soma\_linha\_4(M);

printf("\na soma da linha 4 é: %.0f\n", soma);

soma = soma\_coluna\_2(M);

printf("a soma da coluna 2 é: %.0f\n", soma);

soma = soma\_diagonal\_principal(M);

printf("a soma da diagonal princial é: %.0f\n", soma);

soma = soma\_diagonal\_secundaria(M);

printf("a soma da diagonal secundária é: %.0f\n", soma);

soma = soma\_todos\_os\_elemenos(M);

printf("a soma de todos os elementos é: %.0f\n", soma);

return 0;

}

EXERCÍCIO NÚMERO 6

#include <stdio.h>

int main(void) {

int m = 4;

int sum = 0;

int M[4][4];

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

M[i][j] = rand() % 100;

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

}

printf("\n"); // para pular linha e separar

}

printf("\n"); // para pular linha e separar

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

if (i == j)

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

else

printf(" ");

}

printf("\n");

}

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

if (i > j)

sum += M[i][j];

}

}

printf("\na soma dos elementos abaixo da diagonal pricipal é: %d", sum); //colocar fora do for, porqeu se colocar dentro repete umas mil vezes

}

EXERCÍCIO NÚMERO 7

#include <stdio.h>

int m = 4;

int n = 6;

int M[4][6];

int sum = 0;

int dif = 0;

void matriz\_a(int M[4][6]) {

printf("MATRIZ A\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

M[i][j] = rand() % 100;

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

}

printf("\n");

}

}

void matriz\_b(int M[4][6]) {

printf("MATRIZ B\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

M[i][j] = rand() % 100;

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

}

printf("\n");

}

}

void matriz\_s(int M1[4][6], int M2[4][6], int M3[4][6]) {

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

M3[i][j] = M1[i][j] + M2[i][j];

}

}

}

void matriz\_d(int M1[4][6], int M2[4][6], int M3[4][6]) {

dif = 0;

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

M3[i][j] = M1[i][j] - M2[i][j];

}

}

}

int main(void) {

int matrizA[4][6], matrizB[4][6], matrizS[4][6], matrizD[4][6];

matriz\_a(matrizA);

printf("\n");

matriz\_b(matrizB);

//soma

matriz\_s(matrizA, matrizB, matrizS);

printf("\na soma das matrizes A e B é:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

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

}

printf("\n");

}

//diferença

matriz\_d(matrizA, matrizB, matrizD);

printf("\na diferença das matrizes A e B é:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

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

}

printf("\n");

}

return 0;

}

EXERCÍCIO NÚMERO 8

#include <stdio.h>

int i;

int j;

int m = 10;

void matriz\_princ(int M[10][10]) {

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

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

M[i][j] = rand() % 10;

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

}

printf("\n");

}

}

// linha 2 com a linha 8

void troca\_1(int M[10][10]) {

int troca;

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

troca = M[1][j];

M[1][j] = M[7][j];

M[7][j] = troca;

}

}

// coluna 4 com a coluna 10

void troca\_2(int M[10][10]) {

int troca;

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

troca = M[i][3];

M[i][3] = M[i][9];

M[i][9] = troca;

}

}

// a diagonal principal com a diagonal secundária

void troca\_3(int M[10][10]) {

int troca;

for (i = 0; i < m; i++) { // linha

for (j = 0; j < m; j++) { // coluna

troca = M[i][i];

M[i][i] = M[i][j + 1 - i];

M[i][j + 1 - i] = troca;

}

}

}

// a linha 5 com a coluna 10

void troca\_4(int M[10][10]) {

int troca;

for (i = 0; i < m; i++) { // linha

for (j = 0; j < m; j++) { // coluna

troca = M[4][j];

M[4][j] = M[i][9];

M[i][9] = troca;

}

}

}

void imprimir(int M[10][10]) {

printf("\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

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

}

printf("\n");

}

}

int main(void) {

int mp[10][10];

printf("\nMATRIZ PRINCIPAL\n");

matriz\_princ(mp);

printf("\nPRIMEIRA TROCA");

troca\_1(mp);

imprimir(mp);

printf("\nSEGUNDA TROCA");

troca\_2(mp);

imprimir(mp);

printf("\nTERCEIRA TROCA");

troca\_3(mp);

imprimir(mp);

printf("\nQUARTA TROCA");

troca\_4(mp);

imprimir(mp);

return 0;

}