1)

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define FILAS 4

#define COLUMNAS 4

int generarValorAleatorio() {

return rand() % 10;

}

void inicializarMatriz(int matriz[FILAS][COLUMNAS]) {

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

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

matriz[i][j] = generarValorAleatorio();

}

}

}

void mostrarMatriz(int matriz[FILAS][COLUMNAS]) {

printf("Matriz generada:\n");

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

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

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

}

printf("\n");

}

}

int contarCeros(int matriz[FILAS][COLUMNAS]) {

int contadorCeros = 0;

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

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

if (matriz[i][j] == 0) {

contadorCeros++;

}

}

}

return contadorCeros;

}

int main() {

srand(time(NULL));

int matriz[FILAS][COLUMNAS];

inicializarMatriz(matriz);

mostrarMatriz(matriz);

int cantidadCeros = contarCeros(matriz);

printf("\nCantidad de ceros en la matriz: %d\n", cantidadCeros);

return 0;

}

2)

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define FILAS 4

#define COLUMNAS 4

int generarValorAleatorio() {

return rand() % 10;

}

void inicializarMatriz(int matriz[FILAS][COLUMNAS]) {

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

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

matriz[i][j] = generarValorAleatorio();

}

}

}

void mostrarMatriz(int matriz[FILAS][COLUMNAS]) {

printf("Matriz generada:\n");

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

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

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

}

printf("\n");

}

}

int contarCeros(int matriz[FILAS][COLUMNAS]) {

int contadorCeros = 0;

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

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

if (matriz[i][j] == 0) {

contadorCeros++;

}

}

}

return contadorCeros;

}

int main() {

srand(time(NULL));

int matriz[FILAS][COLUMNAS];

inicializarMatriz(matriz);

mostrarMatriz(matriz);

int cantidadCeros = contarCeros(matriz);

printf("\nCantidad de ceros en la matriz: %d\n", cantidadCeros);

return 0;

}

3)

#include <stdio.h>

#define EMPLEADOS 5

#define DIAS\_SEMANA 5

void capturarVentas(int matriz[EMPLEADOS][DIAS\_SEMANA]) {

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

printf("Ingrese las ventas del empleado %d durante la semana:\n", i + 1);

for (int j = 0; j < DIAS\_SEMANA; j++) {

printf("Día %d: $", j + 1);

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

}

}

}

void determinarVentaMayor(int matriz[EMPLEADOS][DIAS\_SEMANA]) {

printf("\nVentas mayores por empleado:\n");

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

int ventaMayor = matriz[i][0];

int diaVentaMayor = 1;

for (int j = 1; j < DIAS\_SEMANA; j++) {

if (matriz[i][j] > ventaMayor) {

ventaMayor = matriz[i][j];

diaVentaMayor = j + 1;

}

}

printf("Empleado %d: Venta mayor de $%d fue el día %d\n", i + 1, ventaMayor, diaVentaMayor);

}

}

int main() {

int matrizVentas[EMPLEADOS][DIAS\_SEMANA];

capturarVentas(matrizVentas);

determinarVentaMayor(matrizVentas);

return 0;

}

4)

#include <stdio.h>

#define MAX\_ELEMENTOS 10

void leerVector(int vector[], int longitud) {

printf("Ingrese los elementos del vector:\n");

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

printf("Elemento %d: ", i + 1);

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

}

}

void rotarVector(int vector[], int longitud) {

int temp = vector[longitud - 1];

for (int i = longitud - 1; i > 0; i--) {

vector[i] = vector[i - 1];

}

vector[0] = temp;

}

void mostrarVector(int vector[], int longitud) {

printf("Vector resultante:\n");

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

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

}

printf("\n");

}

int main() {

int vector[MAX\_ELEMENTOS];

int longitud;

printf("Ingrese la longitud del vector (menor o igual a %d): ", MAX\_ELEMENTOS);

scanf("%d", &longitud);

if (longitud <= 0 || longitud > MAX\_ELEMENTOS) {

printf("Longitud no válida. El programa terminará.\n");

return 1;

}

leerVector(vector, longitud);

rotarVector(vector, longitud);

mostrarVector(vector, longitud);

return 0;

}

5)

#include <stdio.h>

#define FILAS 15

#define COLUMNAS 12

void leerArreglo(int arreglo[FILAS][COLUMNAS]) {

printf("Ingrese los elementos del arreglo:\n");

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

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

printf("Elemento [%d][%d]: ", i + 1, j + 1);

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

}

}

}

int encontrarMenorElemento(int arreglo[FILAS][COLUMNAS]) {

int menor = arreglo[0][0];

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

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

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

menor = arreglo[i][j];

}

}

}

return menor;

}

int calcularSumaFilas(int arreglo[FILAS][COLUMNAS]) {

int suma = 0;

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

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

suma += arreglo[i][j];

}

}

return suma;

}

int contarNegativosColumnas(int arreglo[FILAS][COLUMNAS]) {

int contadorNegativos = 0;

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

for (int j = 4; j < 9; j++) {

if (arreglo[i][j] < 0) {

contadorNegativos++;

}

}

}

return contadorNegativos;

}

int main() {

int arreglo[FILAS][COLUMNAS];

leerArreglo(arreglo);

int menorElemento = encontrarMenorElemento(arreglo);

printf("\nEl menor elemento del arreglo es: %d\n", menorElemento);

int sumaFilas = calcularSumaFilas(arreglo);

printf("La suma de los elementos de las cinco primeras filas es: %d\n", sumaFilas);

int negativosColumnas = contarNegativosColumnas(arreglo);

printf("El total de elementos negativos en las columnas de la quinta a la novena es: %d\n", negativosColumnas);

return 0;

}

6)

#include <stdio.h>

#define ORDEN\_MATRIZ 12

void leerMatrizCuadrada(int matriz[ORDEN\_MATRIZ][ORDEN\_MATRIZ]) {

printf("Ingrese los elementos de la matriz cuadrada:\n");

for (int i = 0; i < ORDEN\_MATRIZ; i++) {

for (int j = 0; j < ORDEN\_MATRIZ; j++) {

printf("Elemento [%d][%d]: ", i + 1, j + 1);

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

}

}

}

int diagonalPrincipalIgual(int matriz1[ORDEN\_MATRIZ][ORDEN\_MATRIZ], int matriz2[ORDEN\_MATRIZ][ORDEN\_MATRIZ]) {

for (int i = 0; i < ORDEN\_MATRIZ; i++) {

if (matriz1[i][i] != matriz2[i][i]) {

return 0;

}

}

return 1;

}

int main() {

int matriz1[ORDEN\_MATRIZ][ORDEN\_MATRIZ];

int matriz2[ORDEN\_MATRIZ][ORDEN\_MATRIZ];

leerMatrizCuadrada(matriz1);

leerMatrizCuadrada(matriz2);

if (diagonalPrincipalIgual(matriz1, matriz2)) {

printf("\nLa diagonal principal de las matrices es igual.\n");

} else {

printf("\nLa diagonal principal de las matrices no es igual.\n");

}

return 0;

}

7)

#include <stdio.h>

#define FILAS 12

#define COLUMNAS 19

void leerMatriz(int matriz[FILAS][COLUMNAS]) {

printf("Ingrese los elementos de la matriz:\n");

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

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

printf("Elemento [%d][%d]: ", i + 1, j + 1);

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

}

}

}

void cambiarNegativosACeros(int matriz[FILAS][COLUMNAS]) {

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

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

if (matriz[i][j] < 0) {

matriz[i][j] = 0;

}

}

}

}

void mostrarMatriz(int matriz[FILAS][COLUMNAS]) {

printf("Matriz resultante:\n");

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

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

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

}

printf("\n");

}

}

int main() {

int matriz[FILAS][COLUMNAS];

leerMatriz(matriz);

cambiarNegativosACeros(matriz);

mostrarMatriz(matriz);

return 0;

}

8)

#include <stdio.h>

#define FILAS 5

#define COLUMNAS 6

void leerMatriz(int matriz[FILAS][COLUMNAS]) {

printf("Ingrese los elementos de la matriz:\n");

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

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

printf("Elemento [%d][%d]: ", i + 1, j + 1);

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

}

}

}

int contarNegativos(int matriz[FILAS][COLUMNAS]) {

int contadorNegativos = 0;

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

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

if (matriz[i][j] < 0) {

contadorNegativos++;

}

}

}

return contadorNegativos;

}

int contarCerosDiagonalPrincipal(int matriz[FILAS][COLUMNAS]) {

int contadorCeros = 0;

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

if (matriz[i][i] == 0) {

contadorCeros++;

}

}

return contadorCeros;

}

int main() {

int matriz[FILAS][COLUMNAS];

leerMatriz(matriz);

int negativos = contarNegativos(matriz);

printf("\nEl total de elementos negativos en la matriz es: %d\n", negativos);

int cerosDiagonalPrincipal = contarCerosDiagonalPrincipal(matriz);

printf("El total de elementos de la diagonal principal que son igual a cero es: %d\n", cerosDiagonalPrincipal);

return 0;

}

9)

#include <stdio.h>

#include <string.h>

#define CHOEFERES 5

#define DIAS\_SEMANA 6

struct Chofer {

int legajo;

char nombre[50];

float horasTrabajadas[DIAS\_SEMANA];

float sueldoPorHora;

};

void calcularTotalHoras(struct Chofer choferes[CHOEFERES]) {

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

float totalHoras = 0;

for (int j = 0; j < DIAS\_SEMANA; j++) {

totalHoras += choferes[i].horasTrabajadas[j];

}

printf("Total de horas trabajadas para %s: %.2f horas\n", choferes[i].nombre, totalHoras);

}

}

void calcularSueldoSemanal(struct Chofer choferes[CHOEFERES]) {

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

float totalHoras = 0;

for (int j = 0; j < DIAS\_SEMANA; j++) {

totalHoras += choferes[i].horasTrabajadas[j];

}

float sueldoSemanal = totalHoras \* choferes[i].sueldoPorHora;

printf("Sueldo semanal para %s: $%.2f\n", choferes[i].nombre, sueldoSemanal);

}

}

void calcularTotalEmpresa(struct Chofer choferes[CHOEFERES]) {

float totalEmpresa = 0;

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

float totalHoras = 0;

for (int j = 0; j < DIAS\_SEMANA; j++) {

totalHoras += choferes[i].horasTrabajadas[j];

}

totalEmpresa += totalHoras \* choferes[i].sueldoPorHora;

}

printf("Total que pagará la empresa: $%.2f\n", totalEmpresa);

}

void trabajadorMasHorasLunes(struct Chofer choferes[CHOEFERES]) {

float maxHoras = 0;

char nombre[50];

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

if (choferes[i].horasTrabajadas[0] > maxHoras) {

maxHoras = choferes[i].horasTrabajadas[0];

strcpy(nombre, choferes[i].nombre);

}

}

printf("El trabajador que labora más horas el día lunes es: %s\n", nombre);

}

void imprimirReporte(struct Chofer choferes[CHOEFERES]) {

printf("\n--- Reporte ---\n");

calcularTotalHoras(choferes);

calcularSueldoSemanal(choferes);

calcularTotalEmpresa(choferes);

trabajadorMasHorasLunes(choferes);

}

int main() {

struct Chofer choferes[CHOEFERES] = {

{1, "Esteban", {8, 7, 6, 8, 7, 6}, 10.0},

{2, "Juan", {9, 8, 7, 6, 5, 8}, 12.0},

{3, "Pablo", {7, 7, 8, 6, 7, 8}, 11.0},

{4, "Mariana", {8, 7, 8, 7, 6, 5}, 10.5},

{5, "Luna", {7, 6, 5, 6, 7, 8}, 11.5},

};

imprimirReporte(choferes);

return 0;

}

10)

#include <stdio.h>

#define FILAS 10

#define COLUMNAS 10

void leerMatriz(int matriz[FILAS][COLUMNAS]) {

printf("Ingrese los elementos de la matriz:\n");

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

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

printf("Elemento [%d][%d]: ", i + 1, j + 1);

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

}

}

}

void calcularSumaFilasColumnas(int matriz[FILAS][COLUMNAS], int sumaFilas[FILAS], int sumaColumnas[COLUMNAS]) {

// Inicializar los vectores de suma en 0

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

sumaFilas[i] = 0;

}

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

sumaColumnas[j] = 0;

}

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

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

sumaFilas[i] += matriz[i][j];

sumaColumnas[j] += matriz[i][j];

}

}

}

void mostrarVector(int vector[], int longitud, char nombre[]) {

printf("%s: [", nombre);

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

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

if (i < longitud - 1) {

printf(", ");

}

}

printf("]\n");

}

int main() {

int matriz[FILAS][COLUMNAS];

int sumaFilas[FILAS];

int sumaColumnas[COLUMNAS];

leerMatriz(matriz);

calcularSumaFilasColumnas(matriz, sumaFilas, sumaColumnas);

mostrarVector(sumaFilas, FILAS, "Suma de Filas");

mostrarVector(sumaColumnas, COLUMNAS, "Suma de Columnas");

return 0;

}

11)

#include <stdio.h>

#define TORRES 7

#define PISOS\_POR\_TORRE 20

#define DEPARTAMENTOS\_POR\_PISO 6

int calcularCantidadTotalHabitantes(int complejo[TORRES][PISOS\_POR\_TORRE][DEPARTAMENTOS\_POR\_PISO]) {

int totalHabitantes = 0;

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

for (int j = 0; j < PISOS\_POR\_TORRE; j++) {

for (int k = 0; k < DEPARTAMENTOS\_POR\_PISO; k++) {

totalHabitantes++;

}

}

}

return totalHabitantes;

}

int calcularCantidadHabitantesPorTorre(int complejo[TORRES][PISOS\_POR\_TORRE][DEPARTAMENTOS\_POR\_PISO]) {

int totalHabitantesPorTorre[TORRES] = {0};

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

for (int j = 0; j < PISOS\_POR\_TORRE; j++) {

for (int k = 0; k < DEPARTAMENTOS\_POR\_PISO; k++) {

totalHabitantesPorTorre[i]++;

}

}

}

int totalHabitantes = 0;

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

totalHabitantes += totalHabitantesPorTorre[i];

printf("Total de habitantes en la torre %d: %d\n", i + 1, totalHabitantesPorTorre[i]);

}

return totalHabitantes;

}

float calcularPromedioHabitantesPorTorre(int complejo[TORRES][PISOS\_POR\_TORRE][DEPARTAMENTOS\_POR\_PISO]) {

int totalHabitantes = calcularCantidadTotalHabitantes(complejo);

int totalHabitantesPorTorre = calcularCantidadHabitantesPorTorre(complejo);

return (float)totalHabitantesPorTorre / TORRES;

}

float calcularPromedioHabitantesPorPiso(int complejo[TORRES][PISOS\_POR\_TORRE][DEPARTAMENTOS\_POR\_PISO]) {

int totalHabitantes = calcularCantidadTotalHabitantes(complejo);

return (float)totalHabitantes / (TORRES \* PISOS\_POR\_TORRE);

}

int main() {

int complejo[TORRES][PISOS\_POR\_TORRE][DEPARTAMENTOS\_POR\_PISO];

int totalHabitantes = calcularCantidadTotalHabitantes(complejo);

printf("\nCantidad total de habitantes del complejo: %d\n", totalHabitantes);

int totalHabitantesPorTorre = calcularCantidadHabitantesPorTorre(complejo);

printf("\nCantidad promedio de habitantes por torre: %.2f\n", calcularPromedioHabitantesPorTorre(complejo));

printf("\nCantidad promedio de habitantes por piso: %.2f\n", calcularPromedioHabitantesPorPiso(complejo));

return 0;

}

12)

#include <stdio.h>

#define SEMANAS 4

#define DIAS\_POR\_SEMANA 7

void leerVentas(int ventas[SEMANAS][DIAS\_POR\_SEMANA]) {

printf("Ingrese las ventas diarias de cada semana:\n");

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

printf("Semana %d:\n", i + 1);

for (int j = 0; j < DIAS\_POR\_SEMANA; j++) {

printf("Día %d: ", j + 1);

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

}

}

}

void calcularTotalVentas(int ventas[SEMANAS][DIAS\_POR\_SEMANA], int totalVentas[SEMANAS]) {

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

totalVentas[i] = 0;

for (int j = 0; j < DIAS\_POR\_SEMANA; j++) {

totalVentas[i] += ventas[i][j];

}

}

}

void calcularPromedioVentas(int totalVentas[SEMANAS], float promedioVentas[SEMANAS]) {

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

promedioVentas[i] = (float)totalVentas[i] / DIAS\_POR\_SEMANA;

}

}

int determinarSemanaMayorVenta(int totalVentas[SEMANAS]) {

int semanaMayorVenta = 0;

for (int i = 1; i < SEMANAS; i++) {

if (totalVentas[i] > totalVentas[semanaMayorVenta]) {

semanaMayorVenta = i;

}

}

return semanaMayorVenta;

}

void mostrarVector(int vector[], int longitud, char nombre[]) {

printf("%s: [", nombre);

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

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

if (i < longitud - 1) {

printf(", ");

}

}

printf("]\n");

}

int main() {

int ventas[SEMANAS][DIAS\_POR\_SEMANA];

int totalVentas[SEMANAS];

float promedioVentas[SEMANAS];

leerVentas(ventas);

calcularTotalVentas(ventas, totalVentas);

calcularPromedioVentas(totalVentas, promedioVentas);

int semanaMayorVenta = determinarSemanaMayorVenta(totalVentas);

mostrarVector(totalVentas, SEMANAS, "Total de Ventas");

mostrarVector(promedioVentas, SEMANAS, "Promedio de Ventas");

printf("La semana con la mayor venta fue la semana %d\n", semanaMayorVenta + 1);

return 0;

}