

TP N°10

ejercicio 1:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void random(int matriz[4][4]){
    for(int i=0; i<4; i++){
        for(int j=0; j<4; j++){
            matriz[i][j] = rand()%10;
        }
    }
}

int main() {

    int matriz[4][4], i, j, k, cont0 = 0;
    srand(time(NULL));
    random(matriz);
    for(int i=0; i<4; i++){
        for(int j=0; j<4; j++){
            printf("%d\t", matriz[i][j]);
            if(matriz[i][j] == 0){
                cont0++;
            }
        }
        printf("\n");
    }

    printf("hay %d 0", cont0);

    return 0;
}
```

ejercicio 2:

```
#include <stdio.h>

#define CHOAFERES 20
#define DIAS_SEMANA 7

int main() {
    int kiloPorChofer[CHOAFERES][DIAS_SEMANA];

    // Capturar la información de los kilómetros recorridos
    for (int i = 0; i < CHOAFERES; i++) {
        printf("Ingrese los kilómetros recorridos por el chofer %d durante la semana:\n", i + 1);
        for (int j = 0; j < DIAS_SEMANA; j++) {
```

```

        printf("Día %d: ", j + 1);
        scanf("%d", &kiloPorChofer[i][j]);
    }
}

// Generar informe
printf("\nInforme de Kilómetros Recorridos:\n");
for (int j = 0; j < DIAS_SEMANA; j++) {
    int totalKilometrosDia = 0;
    printf("Día %d:\n", j + 1);
    for (int i = 0; i < CHOAFERES; i++) {
        printf("Chofer %d: %d kilómetros\n", i + 1, kiloPorChofer[i][j]);
        totalKilometrosDia += kiloPorChofer[i][j];
    }
    printf("Total Kilómetros Recorridos en el Día %d: %d kilómetros\n\n", j + 1, totalKilometrosDia);
}

// Calcular el total de kilómetros recorridos durante la semana
int totalKilometrosSemana = 0;
for (int i = 0; i < CHOAFERES; i++) {
    for (int j = 0; j < DIAS_SEMANA; j++) {
        totalKilometrosSemana += kiloPorChofer[i][j];
    }
}

// Mostrar el total de kilómetros recorridos durante la semana
printf("Total Kilómetros Recorridos durante la Semana: %d kilómetros\n", totalKilometrosSemana);

return 0;
}

```

ejercicio 3:

```
#include <stdio.h>
```

```

int main() {

    int ventas[5][5];
    int ventamayor = 0;
    int empleadomayor, diamayor;

    for(int empleado=0; empleado<5; empleado++){
        for(int dia=0; dia<5; dia++){
            printf("ingresa la venta del empleado %d para el dia %d:", empleado+1, dia+1);
            scanf("%d", &ventas[empleado][dia]);
            if(ventas[empleado][dia] > ventamayor){
                ventamayor = ventas[empleado][dia];
                empleadomayor = empleado;
                diamayor = dia;
            }
        }
    }
}

```

```
    }  
}
```

```
printf("la venta mayor fue de %d realizada por el empleado %d el dia %d.\n", ventamayor,  
empleadomayor+1, diamayor+1);
```

```
    return 0;  
}
```

ejercicio 4:

```
#include <stdio.h>
```

```
void intercambiar(int *a, int *b){  
    int temp=*a;  
    *a=*b;  
    *b=temp;  
}
```

```
void girarderecha(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;  
}
```

```
int main() {  
    int longitud;  
  
    printf("ingresa la longitud del vector:");  
    scanf("%d", &longitud);  
  
    int vector[longitud];  
  
    printf("ingresa los elementos del vector:\n");  
  
    for(int i=0; i<longitud; i++){  
        scanf("%d", &vector[i]);  
    }  
  
    girarderecha(vector,longitud);  
  
    printf("vector rotado hacia la derecha:\n");  
  
    for(int i=0; i<longitud; i++){  
        printf("%d\t", vector[i]);  
    }  
}
```

```

    }

    return 0;
}

```

ejercicio 5:

```
#include <stdio.h>
```

```
#define FILAS 15
```

```
#define COLUMNAS 12
```

```
int main() {
```

```
    // Declaración e inicialización del arreglo
```

```
    int arreglo[FILAS][COLUMNAS];
```

```
    // Lectura del arreglo
```

```
    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]);
```

```
        }
```

```
    }
```

```
    // Calcular el menor elemento del arreglo
```

```
    int menorElemento = arreglo[0][0];
```

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

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

```
            if (arreglo[i][j] < menorElemento) {
```

```
                menorElemento = arreglo[i][j];
```

```
            }
```

```
        }
```

```
    }
```

```
    // Calcular la suma de los elementos de las cinco primeras filas
```

```
    int sumaFilas = 0;
```

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

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

```
            sumaFilas += arreglo[i][j];
```

```
        }
```

```
    }
```

```
    // Calcular el total de elementos negativos en las columnas de la quinta a la nueve
```

```
    int totalNegativos = 0;
```

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

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

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

```
                totalNegativos++;
```

```
            }
```

```
        }
```

```
    }
```

```

// Mostrar resultados
printf("\nMenor elemento del arreglo: %d\n", menorElemento);
printf("Suma de los elementos de las cinco primeras filas: %d\n", sumaFilas);
printf("Total de elementos negativos en las columnas de la quinta a la nueve: %d\n", totalNegativos);

return 0;
}

```

ejercicio 6:

```

#include <stdio.h>
#include <stdbool.h>

```

```

void leermatriz(int matriz[12][12]){
    printf("ingresa los elementos de la matriz (12x12):\n");
    for(int i=0; i<12; i++){
        for(int j=0; j<12; j++){
            printf("matriz[%d][%d]:",i,j);
            scanf("%d", &matriz[i][j]);
        }
    }
}

```

```

bool comparardiagonales(int matriz1[12][12], int matriz2[12][12]){
    for(int i=0; i<12; i++){
        if(matriz1[i][i] != matriz2[i][i]){
            return false;
        }
    }

    return true;
}

```

```

int main() {

    int matrizA[12][12];
    int matrizB[12][12];

    leermatriz(matrizA);
    leermatriz(matrizB);

    bool soniguales = comparardiagonales(matrizA, matrizB);

    if(soniguales){
        printf("las diagonales principales de ambas matrices son iguales.\n");
    }else{
        printf("las diagonales principales de ambas matrices no son iguales.\n");
    }
}

```

```
        return 0;
    }
}
```

ejercicio 7:

```
#include <stdio.h>
```

```
#define FILAS 12
```

```
#define COLUMNAS 19
```

```
// Función para cambiar los valores negativos por cero en la matriz
```

```
void cambiarNegativosACero(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;
            }
        }
    }
}
```

```
// Función para mostrar la matriz
```

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

```
    for (int i = 0; i < FILAS; i++) {
        for (int j = 0; j < COLUMNAS; j++) {
            printf("%d\t", matriz[i][j]);
        }
        printf("\n");
    }
}
```

```
int main() {
```

```
    // Declaración e inicialización de la matriz
```

```
    int matriz[FILAS][COLUMNAS];
```

```
    // Lectura de la matriz
```

```
    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]);
        }
    }
}
```

```
    // Cambiar los valores negativos por cero
```

```
    cambiarNegativosACero(matriz);
```

```
    // Mostrar la matriz resultante
```

```
    printf("\nMatriz resultante:\n");
```

```
    mostrarMatriz(matriz);
```

```
    return 0;
```

```
}
```

ejercicio 8:

```
#include <stdio.h>
```

```
#define FILAS 5
```

```
#define COLUMNAS 6
```

```
// Función para contar los elementos negativos en la matriz
```

```
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;  
}
```

```
// Función para contar los elementos de la diagonal principal que son igual a cero
```

```
int contarCerosDiagonal(int matriz[FILAS][COLUMNAS]) {  
    int contadorCerosDiagonal = 0;  
    for (int i = 0; i < FILAS; i++) {  
        if (matriz[i][i] == 0) {  
            contadorCerosDiagonal++;  
        }  
    }  
    return contadorCerosDiagonal;  
}
```

```
// Función para mostrar la matriz
```

```
void mostrarMatriz(int matriz[FILAS][COLUMNAS]) {  
    for (int i = 0; i < FILAS; i++) {  
        for (int j = 0; j < COLUMNAS; j++) {  
            printf("%d\t", matriz[i][j]);  
        }  
        printf("\n");  
    }  
}
```

```
int main() {
```

```
    // Declaración e inicialización de la matriz
```

```
    int matriz[FILAS][COLUMNAS];
```

```
    // Lectura de la matriz
```

```
    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]);
    }
}

// Mostrar la matriz
printf("\nMatriz ingresada:\n");
mostrarMatriz(matriz);

// Contar elementos negativos
int cantidadNegativos = contarNegativos(matriz);
printf("\nCantidad de elementos negativos en la matriz: %d\n", cantidadNegativos);

// Contar elementos de la diagonal principal que son igual a cero
int cantidadCerosDiagonal = contarCerosDiagonal(matriz);
printf("Cantidad de elementos de la diagonal principal que son igual a cero: %d\n",
cantidadCerosDiagonal);

return 0;
}

```

ejercicio 9:

```
#include <stdio.h>
```

```
#define CHOAFERES 5
```

```
#define DIAS_SEMANA 6
```

```
// Función para calcular el total de horas trabajadas a la semana para un chofer
```

```
int calcularTotalHorasSemana(int horasTrabajadas[DIAS_SEMANA]) {
    int totalHoras = 0;
    for (int i = 0; i < DIAS_SEMANA; i++) {
        totalHoras += horasTrabajadas[i];
    }
    return totalHoras;
}

```

```
// Función para calcular el sueldo semanal para un chofer
```

```
float calcularSueldoSemanal(int horasTrabajadas[DIAS_SEMANA], float sueldoPorHora) {
    return sueldoPorHora * calcularTotalHorasSemana(horasTrabajadas);
}

```

```
int main() {
```

```
    // Declaración e inicialización de datos
```

```
    int legajos[CHOAFERES] = {101, 102, 103, 104, 105};
```

```
    int horasTrabajadas[CHOAFERES][DIAS_SEMANA] = {
        {8, 8, 7, 9, 8, 6},
        {7, 6, 8, 9, 7, 8},
        {8, 8, 8, 8, 8, 8},
        {9, 9, 9, 9, 9, 9},
        {8, 8, 8, 8, 8, 8}
    };

```

```
    float sueldoPorHora[CHOAFERES] = {10.5, 11.0, 10.0, 9.5, 10.5};

```



```

// Calcular el total que pagará la empresa y el chofer que labora más horas el día lunes
float totalEmpresa = 0;
int maxHorasLunes = 0;
int choferMaxHorasLunes = -1;

// Imprimir encabezado del reporte
printf("Legajo\tTotal Horas\tSueldo Semanal\n");

// Calcular el total de horas y sueldo semanal para cada chofer
for (int i = 0; i < CHOAFERES; i++) {
    int totalHoras = calcularTotalHorasSemana(horasTrabajadas[i]);
    float sueldoSemanal = calcularSueldoSemanal(horasTrabajadas[i], sueldoPorHora[i]);
    totalEmpresa += sueldoSemanal;

    // Verificar si el chofer trabaja más horas el día lunes
    if (horasTrabajadas[i][0] > maxHorasLunes) {
        maxHorasLunes = horasTrabajadas[i][0];
        choferMaxHorasLunes = i;
    }

    // Imprimir información del chofer en el reporte
    printf("%d\t%d\t%.2f\n", legajos[i], totalHoras, sueldoSemanal);
}

// Imprimir el total que pagará la empresa
printf("\nTotal que pagará la empresa: %.2f\n", totalEmpresa);

// Imprimir el nombre del chofer que labora más horas el día lunes
if (choferMaxHorasLunes != -1) {
    printf("El chofer que labora más horas el día lunes es el chofer con legajo %d\n",
legajos[choferMaxHorasLunes]);
} else {
    printf("No hay información sobre el chofer que labora más horas el día lunes.\n");
}

return 0;
}

```

ejercicio 10:

```
#include <stdio.h>
```

```

int main() {

    int matriz[10][10];
    int sumafilas[10] = {0};
    int sumacolumnas[10] = {0};

    printf("ingresa los valores de la matriz (10x10):\n");
    for(int i=0; i<10; i++){
        for(int j=0; j<10; j++){

```

```

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

for(int i=0; i<10; i++){
    for(int j=0; j<10; j++){
        sumafilas[i] += matriz[i][j];
    }
}

for(int i=0; i<10; i++){
    for(int j=0; j<10; j++){
        sumacolumnas[j] += matriz[i][j];
    }
}

printf("sumas de las filas:\n");
for(int i=0; i<10; i++){
    printf("fila %d: %d\n", i+1, sumafilas[i]);
}

printf("sumas de las columnas:\n");
for(int j=0; j<10; j++){
    printf("columna %d: %d\n", j+1, sumacolumnas[j]);
}

return 0;
}

```

ejercicio 11:

```

#include <stdio.h>
#include <stdlib.h>

```

```

void habitantes(int torres[7][20][6]){
    int i,j,k;
    for(i=0; i<7; i++){
        for(j=0; j<20; j++){
            for(k=0; k<6; k++){
                printf("%d\t",torres[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }
}

int main() {

    int torres[7][20][6]={0},i,j,k,totalhabitantes=0,acu=0;

```

```

float promedio[7][20],totaltorres[7]={0};

for(i=0; i<7; i++){
    for(j=0; j<20; j++){
        for(k=0; k<6; k++){
            printf("ingrese la cantidad de habitantes de la torre %d:, del piso %d: y del
departamento %d:",i+1,j+1,k+1);
            scanf("%d", &torres[i][j][k]);
            totalhabitantes += torres[i][j][k];
            totaltorres[i] += torres[i][j][k];
            acu += torres[i][j][k];
        }

        promedio[i][j]=acu;
        acu=0;

    }
}

habitantes(torres);

printf("cantidad de habitantes del complejo: %d\n", totalhabitantes);
printf("cantidad de habitantes por torre: \n");

for(i=0; i<7; i++){
    printf("torre %d: %.2f\n",i+1,totaltorres[i]);
}

printf("promedio de habitantes por torre: \n");

for(i=0; i<7; i++){
    printf("torre %d: %.2f\n",i+1,totaltorres[i]/7);
}
for(i=0; i<2; i++){
    for(j=0; j<2; j++){
        printf("torre %d:, piso %d: %.2f\n",i+1,j+1,promedio[i][j]/6);
    }
}

return 0;
}

```

ejercicio 12:

```
#include <stdio.h>
```

```

int main() {

    int ventas[4][7];
    int totalventas[4]={0};
    float promedioventas[4];

```

```

int semanamayorventa = 0;
int maxventa = 0;

printf("ingresa las ventas de cada dia (4 semanas x 7 dias):\n");
for(int i=0; i<4; i++){
    printf("semana %d:\n", i+1);
    for(int j=0; j<7; j++){
        printf("dia %d: $", j+1);
        scanf("%d", &ventas[i][j]);
        totalventas[i] += ventas[i][j];
    }
}

for(int i=0; i<4; i++){
    promedioventas[i] = (float)totalventas[i]/7;
    if(totalventas[i] > maxventa){
        maxventa = totalventas[i];
        semanamayorventa = i;
    }
}

printf("resultados:\n");
for(int i=0; i<4; i++){
    printf("semana %d - total de ventas: $%d - promedio de ventas : $%.2f\n",
i+1,totalventas[i],promedioventas[i]);
}

printf("la semana con las mayores ventas es la semana %d con un total de $%d.\n",
semanamayorventa +1, maxventa);

return 0;
}

```