

ejercicio 1:

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

```
#define FILAS 2
```

```
#define COLUMNAS 3
```

```
int main() {
```

```
    // Declaración de la matriz
```

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

```
    // Ingreso de datos
```

```
    printf("Ingrese los valores para la matriz de %d filas por %d columnas:\n", FILAS, COLUMNAS);
```

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

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

```
            printf("Ingrese el valor para la posición [%d][%d]: ", i, j);
```

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

```
        }
```

```
    }
```

```
    // Mostrar el contenido de la matriz
```

```
    printf("\nContenido de la matriz:\n");
```

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

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

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

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

ejercicio 2:

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

```
#define FILAS 2
```

```
#define COLUMNAS 3
```

```
int main() {
```

```
    // Declaración de la matriz
```

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

```
    // Ingreso de datos
```

```
    printf("Ingrese los valores para la matriz de %d filas por %d columnas:\n", FILAS, COLUMNAS);
```

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

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

```
            printf("Ingrese el valor para la posición [%d][%d]: ", i, j);
```

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

```
        }
```

```
    }
```

```
    // Multiplicar por 4 todos los elementos de la primera fila
```

```

for (int j = 0; j < COLUMNAS; j++) {
    matriz[0][j] *= 4;
}

// Multiplicar por 3 todos los elementos de la tercera columna
for (int i = 0; i < FILAS; i++) {
    matriz[i][2] *= 3;
}

// Mostrar el contenido de la matriz modificada
printf("\nContenido de la matriz modificada:\n");
for (int i = 0; i < FILAS; i++) {
    for (int j = 0; j < COLUMNAS; j++) {
        printf("%d ", matriz[i][j]);
    }
    printf("\n");
}

return 0;
}

```

ejercicio 3:

```

#include <stdio.h>
#include <limits.h>

```

```

#define FILAS 2
#define COLUMNAS 3

```

```

int main() {
    // Declaración de la matriz
    int matriz[FILAS][COLUMNAS];

    // Ingreso de datos
    printf("Ingrese los valores para la matriz de %d filas por %d columnas:\n", FILAS, COLUMNAS);
    for (int i = 0; i < FILAS; i++) {
        for (int j = 0; j < COLUMNAS; j++) {
            printf("Ingrese el valor para la posición [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    // Multiplicar por 4 todos los elementos de la primera fila
    for (int j = 0; j < COLUMNAS; j++) {
        matriz[0][j] *= 4;
    }

    // Multiplicar por 3 todos los elementos de la tercera columna
    for (int i = 0; i < FILAS; i++) {
        matriz[i][2] *= 3;
    }
}

```

```

// Encontrar el menor valor de la matriz
int menorValor = INT_MAX;
for (int i = 0; i < FILAS; i++) {
    for (int j = 0; j < COLUMNAS; j++) {
        if (matriz[i][j] < menorValor) {
            menorValor = matriz[i][j];
        }
    }
}

// Sumar el contenido de todos los elementos de la matriz
int suma = 0;
for (int i = 0; i < FILAS; i++) {
    for (int j = 0; j < COLUMNAS; j++) {
        suma += matriz[i][j];
    }
}

// Mostrar el contenido de la matriz modificada
printf("\nContenido de la matriz modificada:\n");
for (int i = 0; i < FILAS; i++) {
    for (int j = 0; j < COLUMNAS; j++) {
        printf("%d ", matriz[i][j]);
    }
    printf("\n");
}

// Informar el menor valor y la suma de todos los elementos
printf("\nMenor valor de la matriz: %d\n", menorValor);
printf("Suma de todos los elementos de la matriz: %d\n", suma);

return 0;
}

```

ejercicio 4:

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

```
#define NUM_LINEAS 3
```

```
#define NUM_COCHES_POR_LINEA 12
```

```
int main() {
```

```
    // Declaración de variables
```

```
    float recaudacionPorLinea[NUM_LINEAS] = {0};
```

```
    float recaudacionPorCoche[NUM_COCHES_POR_LINEA] = {0};
```

```
    float recaudacionTotal = 0;
```

```
    // Ingreso de planillas hasta que el número de coche sea menor a 1
```

```
    while (1) {
```

```
        // Declaración de variables locales
```

```
        int numCoche, numLinea;
```

```
        float recaudacion;
```

```

// Ingreso de la planilla
printf("Ingrese el número de coche (o menor a 1 para finalizar): ");
scanf("%d", &numCoche);

// Verificar si se debe finalizar el ingreso de planillas
if (numCoche < 1) {
    break;
}

printf("Ingrese el número de línea (1 a %d): ", NUM_LINEAS);
scanf("%d", &numLinea);

printf("Ingrese la recaudación: ");
scanf("%f", &recaudacion);

// Validar el número de línea
if (numLinea < 1 || numLinea > NUM_LINEAS) {
    printf("Error: El número de línea debe estar entre 1 y %d. Ingrese la planilla nuevamente.\n",
NUM_LINEAS);
    continue;
}

// Actualizar las recaudaciones
recaudacionPorLinea[numLinea - 1] += recaudacion;
recaudacionPorCoche[(numLinea - 1) * NUM_COCHES_POR_LINEA + (numCoche - 1)] +=
recaudacion;
recaudacionTotal += recaudacion;
}

// Mostrar los resultados
printf("\nRecaudación total por línea:\n");
for (int i = 0; i < NUM_LINEAS; i++) {
    printf("Línea %d: %.2f\n", i + 1, recaudacionPorLinea[i]);
}

printf("\nRecaudación total por coche:\n");
for (int i = 0; i < NUM_LINEAS; i++) {
    for (int j = 0; j < NUM_COCHES_POR_LINEA; j++) {
        printf("Línea %d - Coche %d: %.2f\n", i + 1, j + 1, recaudacionPorCoche[i *
NUM_COCHES_POR_LINEA + j]);
    }
}

printf("\nRecaudación total general: %.2f\n", recaudacionTotal);

return 0;
}

```

ejercicio 5:

ejercicio 6:

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

```
#define NUM_PISOS 20
```

```
#define NUM_DEPARTAMENTOS_POR_PISO 6
```

```
int main() {
```

```
    // Declaración de variables
```

```
    int torre[NUM_PISOS][NUM_DEPARTAMENTOS_POR_PISO];
```

```
    int totalHabitantes = 0;
```

```
    // Ingreso de datos
```

```
    printf("Ingrese la cantidad de habitantes por departamento:\n");
```

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

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

```
            printf("Piso %d - Departamento %d: ", i + 1, j + 1);
```

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

```
            totalHabitantes += torre[i][j];
```

```
        }
```

```
    }
```

```
    // Cantidad total de habitantes
```

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

```
    // Cantidad promedio de habitantes por piso
```

```
    float promedioHabitantesPorPiso = (float)totalHabitantes / NUM_PISOS;
```

```
    printf("Cantidad promedio de habitantes por piso: %.2f\n", promedioHabitantesPorPiso);
```

```
    return 0;
```

```
}
```