

1. DATOS INFORMATIVOS

Carrera: Electronica y Automatizacion

Asignatura: Fundamentos de Programacion

Tema del taller: Ejercicios para github

Docente: Ing. Jenny Ruiz

Integrantes: Mendoza Cortez Christopher Alexander

Fecha: 08/12/2025 Paralelo: 29583

2. DESARROLLO

Problema 2.2.1 Suma de componentes de una matriz

Halle la suma de las componentes de una matriz cuya dimensión y componentes se leen por filas. Suponga que el número de filas N y columnas M son menores o iguales a 10.

Para leer una matriz por filas lo más sencillo es usar un doble bucle anidado. El bucle exterior lee las filas y el bucle interior lee las columnas para cada fila que queda vacía. El bucle externo va cambiando la variable i, la cual se debe iniciar en cero. Asimismo, observe que en el diagrama de flujo el índice de las filas se establece desde 1 al número de filas y el índice de las columnas comienza en 1 al número de columnas; sin embargo, en los índices de vectores y matrices comienzan en 0.

Código

```
#include <stdio.h>

int main() {
    int N, M;
    int matriz[10][10];
    int suma = 0;

    // Leer dimensiones
    printf("Ingrese el número de filas (N): ");
    scanf("%d", &N);
    printf("Ingrese el número de columnas (M): ");
    scanf("%d", &M);

    // Validar dimensiones
```

```

if (N > 10 || M > 10 || N <= 0 || M <= 0) {
    printf("Dimensiones inválidas. Deben ser entre 1 y 10.\n");
    return 1;
}

// Leer matriz por filas
printf("Ingrese los elementos de la matriz:\n");
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        printf("Elemento [%d][%d]: ", i, j);
        scanf("%d", &matriz[i][j]);
        suma += matriz[i][j]; // Acumular suma
    }
}

// Mostrar resultado
printf("La suma de los elementos de la matriz es: %d\n", suma);

return 0;
}

```

Corrida del programa

```

Ingrese el numero de filas (N): 2
Ingrese el numero de columnas (M): 2
Ingrese los elementos de la matriz:
Elemento [0][0]: 1
Elemento [0][1]: 2
Elemento [1][0]: 3
Elemento [1][1]: 1
La suma de los elementos de la matriz es: 7

Process returned 0 (0x0)  execution time : 5.418 s
Press any key to continue.

```

Link de onlinegdb

<https://onlinegdb.com/KaE2lUKgy>

Problema 2.2.2 Escritura de matriz en sentido inverso.

Dada una matriz de $N \times N$ elementos, realice un algoritmo que recorra la matriz por filas desde la última a la primera y cada fila en sentido inverso, y de la última columna a la primera, de modo que se vaya mostrando cada elemento.

La solución a este problema consiste en recorrer la matriz invirtiendo el sentido habitual de los bucles. Observe cómo, en este caso, los bucles de filas y columnas en las variables i y j comienzan en la última fila/columna de la matriz. La condición de permanencia en los bucles es ahora $i >= 1$ o $j >= 1$ (en C $i >= 0$ o $j >= 0$) y las variables se decrementan en cada iteración.

Código

```
#include <stdio.h>
#define N 4

void main(void)
{
    int mat[N][N] = {{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
    int i, j;

    printf("La matriz es:\n\n");
    for(i = 0; i < N; i++)
    {
        for(j = 0; j < N; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }

    printf("\nLa matriz a la inversa es:\n\n");
    for(i = N - 1; i >= 0; i--)
    {
        for(j = N - 1; j >= 0; j--)
            printf("%d ", mat[i][j]);
        printf("\n");
    }
}
```

Corrida del programa

```
La matriz es:
1 2 3 4
2 1 2 3
3 2 1 0
4 3 2 1
Introduzca la primera fila a cambiar
4
Introduzca la segunda fila a cambiar
1

La matriz alterada es:
4 3 2 1
2 1 2 3
3 2 1 0
1 2 3 4

Process returned 4 (0x4)    execution time : 2.070 s
Press any key to continue.
```

Link de onlinegdb

<https://onlinegdb.com/eicqd9yZe>

Problema 2.2.3 Máximo de una fila.

Escriba un programa que lea una matriz de N filas y N columnas de valores enteros. A

continuación, el programa debe pedir el número de una fila y mostrar por pantalla el valor de la mayor componente de esa fila.

Tal como ya se discutió en el problema 1.7, la dificultad de calcular el máximo valor de un vector (en este caso un vector fila de una matriz) reside en decidir qué valor inicial se le da a la variable que va a almacenar el máximo (*max*). Imagine que se asume que todos los números del vector son positivos y se inicializa *max* = -1000. Se procede entonces a comparar este valor con todas las componentes del vector *y*, si alguna es mayor, se actualiza el valor de *max* con el valor de esa componente. Podría ocurrir, sin embargo, que todas las componentes del vector sean menores que -1000, en cuyo caso el valor del máximo calculado sería erróneamente -1000.

Una forma sencilla de solucionar este problema es simplemente iniciar el valor de *max* con el valor de la primera componente del vector (cualquier componente del vector valdría en realidad para inicializar), y proceder a continuación con las comparaciones como se ha indicado. De este modo no se fuerza a ninguna suposición sobre el rango de valores donde se encuentran las componentes del vector.

Código

```
#include <stdio.h>
#define N 4

void main(void)
{
    int mat[N][N] = {{2,0,0,0},{2,1,0,0},{3,2,1,0},{4,3,2,1}};
    int i, j, fil, max;

    printf("La matriz es:\n\n");
    for(i = 0; i < N; i++)
    {
        for(j = 0; j < N; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }

    printf("Introduzca la fila\n");
    scanf("%d", &fil);
    fil--; // Suponemos fila real (de 1 a N)

    max = mat[fil][0];
    for(i = 1; i < N; i++)
        if(max < mat[fil][i])
            max = mat[fil][i];
```

```

    printf("\nEl máximo de la fila %d es: %d\n", fil + 1, max);
}

```

Corrida del programa

```

La matriz es:
2 0 0 0
2 1 0 0
3 2 1 0
4 3 2 1
Introduzca la fila
4

El maximo de la fila 4 es: 4

Process returned 30 (0x1E)  execution time : 0.383 s
Press any key to continue.

```

Link de onlinegdb

<https://onlinegdb.com/YR-WIEPZFX>

Problema 2.2.4 Intercambiar las filas i, j de una matriz.

Escriba un programa que intercambie las filas i y j de una matriz de enteros de $N \times N$ componentes, siendo i y j dos valores introducidos por teclado.

La solución se muestra en el diagrama de la figura 2.13, junto con su tabla de objetos y codificación. La dificultad del problema reside en intercambiar las filas sin perder información, tal y como pasa aquí al intercambiar el elemento 7 de la filas a y b:

```
mat[a][6] = mat[b][6]; // Sobrescribe mat[a][6] y se pierde su valor
```

```
mat[b][6] = mat[a][6]; // Queda mat[a][6] con el mismo valor que mat[b][6]
```

Es necesario usar una variable auxiliar para no perder el valor original de `mat[a][6]`.

```
aux = mat[a][6];
```

```
mat[a][6] = mat[b][6];
```

```
mat[b][6] = aux;
```

codigo

```
#include <stdio.h>
#define N 4

void main(void)
{
    int mat[N][N] = {{1,2,3,4},{2,1,2,3},{3,2,1,0},{4,3,2,1}};
```

```

int aux, i, j, k;

printf("La matriz es:\n\n");
for(i = 0; i < N; i++)
{
    for(j = 0; j < N; j++)
        printf("%d ", mat[i][j]);
    printf("\n");
}

printf("Introduzca la primera fila a cambiar\n");
scanf("%d", &i);
printf("Introduzca la segunda fila a cambiar\n");
scanf("%d", &j);

for(k = 0; k < N; k++)
{
    aux = mat[i-1][k];
    mat[i-1][k] = mat[j-1][k];
    mat[j-1][k] = aux;
}

printf("\nLa matriz alterada es:\n\n");
for(i = 0; i < N; i++)
{
    for(j = 0; j < N; j++)
        printf("%d ", mat[i][j]);
    printf("\n");
}

```

Corrida del programa

```

La matriz es:

1 2 3 4
2 1 2 3
3 2 1 0
4 3 2 1
Introduzca la primera fila a cambiar
1
Introduzca la segunda fila a cambiar
3

La matriz alterada es:

3 2 1 0
2 1 2 3
1 2 3 4
4 3 2 1

Process returned 4 (0x4)  execution time : 6.149 s
Press any key to continue.

```

Link de onlinedb

https://onlinedb.com/2_m6BPxmq

3. CONCLUSIONES

Este ejercicio me ayudó a entender que para recorrer una matriz al revés no hace falta hacer cosas complicadas, solo cambiar cómo funcionan los bucles y los índices. Me di cuenta de que la lógica es lo más importante, porque si entiendes cómo se mueven las filas y columnas, puedes hacer que el programa haga lo que quieras. Creo que este tipo de problemas son buenos para aprender a pensar como programador.

4. RECOMENDACIONES

Yo recomiendo practicar mucho, no solo copiar el código. Primero intentar escribirlo uno mismo y después probar con otros casos, como matrices más grandes o que el usuario las escriba. También es bueno hacer un pseudocódigo antes, porque así tienes claro qué vas a hacer. Y algo que aprendí es que hay que validar lo que el usuario pone, porque si pone algo mal el programa puede fallar. Por último, comentar el código ayuda bastante para no perderse.

5. REFERENCIAS