

# UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE



Nombre: Criollo Solano Jahi

NRC: 29583

Fecha: 06/12/2025

Tema: Ejercicios de MTZ



## 1. Suma de componentes de una matriz.

Halle la suma de las componentes de una matriz, cuya dimensión y componentes se han de leer por teclado y 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 interior lee todas las columnas para una fila dada. El bucle externo va cambiando la fila que hay que leer. A medida que se leen las componentes se irán acumulando en la variable res, la cual se debe iniciar a 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 desde 1 al número de columnas; sin embargo, en C los índices de vectores y matrices comienzan en 0.

La imagen muestra un editor de código con el archivo MTZ1.c y una terminal ejecutando el programa MTZ1.exe. El código en C define una matriz de enteros, la lee por filas y calcula la suma de todos sus elementos. La terminal muestra la entrada de 2 filas y 3 columnas, la matriz ingresada (todas las celdas contienen el valor 2) y el resultado final de la suma, que es 12.

```
int matriz[N][M];

printf("\nIngrese los elementos de la matriz por filas:");
for (i = 0; i < N; i++) {
    for (j = 0; j < M; j++) {
        printf("Elemento [%d][%d]: ", i + 1, j + 1);
        scanf("%d", &matriz[i][j]);
        suma += matriz[i][j];
    }
}

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

printf("\nLa suma de todos los elementos es: %d\n", suma);
return 0;
```

Terminal output:

```
"C:\Users\Jahir\Documents\Deberes U\1er Semestre\Fund.Programaci\4n\MTZ 1.exe"
Ingrese el numero de filas (N <= 10): 2
Ingrese el numero de columnas (M <= 10): 3
Ingrese los elementos de la matriz por filas:
Elemento [1][1]: 2
Elemento [1][2]: 2
Elemento [1][3]: 2
Elemento [2][1]: 2
Elemento [2][2]: 2
Elemento [2][3]: 2
Matriz ingresada:
2 2 2
2 2 2
La suma de todos los elementos es: 12
Process returned 0 (0x0)   execution time : 11.966 s
Press any key to continue.
```

Online GDB: <https://onlinegdb.com/DzGEtv7xEf>

## 2. Escritura de matriz en sentido inverso

Dada una matriz de  $N \times N$  elementos, realice un algoritmo que recorra la por filas desde la última a la primera y cada fila en sentido inverso, y de la 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 las variables  $i$  y  $j$  comienzan en la última fila/columna de la matriz. La condición de permanencia en los bucles es ahora  $i \geq 1$  o  $j \geq 1$  (en C  $i \geq 0$  o  $j \geq 0$ ).

variables se decrementan en cada iteración. A continuación, se muestra el diagrama de flujo de la solución en la figura como la tabla de objetos y codificación en C3

The image shows a C++ IDE with two windows. The left window displays the source code for a program that swaps two rows in a matrix. The right window shows the program's execution output.

**Source Code (MTZ 2.c):**

```

}

int a = i - 1;
int b = j - 1;

if (a != b) {
    for (int c = 0; c < M; c++)
        int aux = mat[a][c];
        mat[a][c] = mat[b][c];
        mat[b][c] = aux;
    }
    printf("\nSe han intercambiado las filas i y j\n");
} else {
    printf("\nLas filas i y j son iguales\n");
}

printf("\nMatriz despues del intercambio:\n");
for (int f = 0; f < N; f++) {
    for (int c = 0; c < M; c++)
        printf("%4d", mat[f][c]);
    printf("\n");
}

return 0;
}

```

**Execution Output:**

```

Ingrese el numero de filas N (1..10): 4
Ingrese el numero de columnas M (1..10): 4

Ingrese los elementos de la matriz (4 x 4):
mat[1][1]: 1
mat[1][2]: 2
mat[1][3]: 3
mat[1][4]: 4
mat[2][1]: 5
mat[2][2]: 6
mat[2][3]: 7
mat[2][4]: 8
mat[3][1]: 9
mat[3][2]: 10
mat[3][3]: 11
mat[3][4]: 12
mat[4][1]: 13
mat[4][2]: 14
mat[4][3]: 15
mat[4][4]: 16

Matriz original:
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16

Ingrese la fila i a intercambiar (1..4): 2
Ingrese la fila j a intercambiar (1..4): 3

Se han intercambiado las filas 2 y 3.

Matriz despues del intercambio:
1  2  3  4
9 10 11 12
5  6  7  8
13 14 15 16

```

Online GDB: [https://onlinegdb.com/\\_tcA2dqjs](https://onlinegdb.com/_tcA2dqjs)

### 3. Maximo 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  $\text{max} = -1000$ .

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, se muestra a continuación la solución planteada en el diagrama de flujo de la figura.

```
for (int f = 0; f < N; f++) {
    for (int c = 0; c < M; c++)
        printf("mat[%d][%d]: ", f, c);
    scanf("%d", &mat[f][c]);
}

printf("\nMatriz ingresada:\n");
for (int f = 0; f < N; f++) {
    for (int c = 0; c < M; c++)
        printf("%4d", mat[f][c]);
    printf("\n");
}

int fila;
printf("\nIngrese el numero de la fila para buscar el maximo (1..3): ");
scanf("%d", &fila);

if (fila < 1 || fila > N) {
    printf("Error: fila fuera de rango\n");
    return 1;
}
```

Output:

```
Ingrese el numero de filas N (1..10): 3
Ingrese el numero de columnas M (1..10): 2
Ingrese los elementos de la matriz (3 x 2):
mat[1][1]: 3
mat[1][2]: 44
mat[2][1]: 2
mat[2][2]: 2
mat[3][1]: 88
mat[3][2]: 2
Matriz ingresada:
 3 44
 2  2
88  2
Ingrese el numero de la fila para buscar el maximo (1..3): 1
El valor maximo en la fila 1 es: 44
Process returned 0 (0x0)   execution time : 18.605 s
Press any key to continue.
```

Online GDB: <https://onlinegdb.com/rOc-SaCF8>

#### 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 NxN

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 las 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 mata a6

```
Aux = mat(a)[6];
```

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

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

```
MTZ 1.c x MTZ 2.c x MTZ 3.c x MTZ 4.c x "C:\Users\Jahir\Documents\Deberes U\1er Semestre\Fund.Programaci3n\MTZ 4.exe

// CRIOLLO JAHIR
//MTZ4
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    int N;

    printf("Ingrese N (dimension de
    if (scanf("%d", &N) != 1) {
        printf("Error.\n");
        return 1;
    }
    if (N < 1 || N > 10) {
        printf("Error: N fuera de r
        return 1;
    }

    int mat[10][10];

    printf("\nIngrese los elementos
    for (int f = 0; f < N; f++) {
        for (int c = 0; c < N; c++) {
            printf("mat[%d][%d]: ",
            if (scanf("%d", &mat[f]
                printf("Error.\n");
                return 1;
            }
        }
    }

    Ingrese los elementos de la matriz (3 x 3):
    mat[1][1]: 1
    mat[1][2]: 2
    mat[1][3]: 3
    mat[2][1]: 44
    mat[2][2]: 55
    mat[2][3]: 66
    mat[3][1]: 7
    mat[3][2]: 8
    mat[3][3]: 9

    Matriz original:
    1  2  3
    44 55 66
    7  8  9

    Ingrese la fila i a intercambiar (1..3): 1
    Ingrese la fila j a intercambiar (1..3): 2

    Se intercambiaron las filas 1 y 2.

    Matriz despues del intercambio:
    44 55 66
    1  2  3
    7  8  9

    Process returned 0 (0x0)   execution time : 11.776 s
    Press any key to continue.
```

Online GDB: <https://onlinegdb.com/-RUdkq4WF>