



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

Nombre: Daniel Iza

NRC: 29583

Fecha: 07/12/2025

Tema: Problemas de matrices



1.

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

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

```
#define MAX_SIZE 10
```

```
int main() {
```

```
    int N, M;
```

```
    int matrix[MAX_SIZE][MAX_SIZE];
```

```
    long long sum = 0;
```

```
    printf("### Codificación en C del problema 2.2.1 ###\n");
```

```
    printf("Ingrese el numero de filas (N): ");
```

```
    if (scanf("%d", &N) != 1) {
```

```
        printf("Error al leer la entrada de N.\n");
```

```
        return 1;
```

```
    }
```

```
    printf("Ingrese el numero de columnas (M): ");
```

```
    if (scanf("%d", &M) != 1) {
```

```
        printf("Error al leer la entrada de M.\n");
```

```
        return 1;
```

```
    }
```

```
    if (N < 0 || N > MAX_SIZE || M < 0 || M > MAX_SIZE) {
```

```
        printf("Dimensiones invalidas. N y M deben estar entre 1 y %d.\n", MAX_SIZE);
```

```
        return 1;
```

```
}
```

```
printf("\nIngresando elementos de la matriz %dx%d:\n", N, M);
```

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

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

```
        printf("Ingrese el elemento matrix[%d][%d]: ", i, j);
```

```
        if (scanf("%d", &matrix[i][j]) != 1) {
```

```
            printf("Error al leer el elemento. Dimensiones invalidas o tipo de dato incorrecto.\n");
```

```
            return 1;
```

```
        }
```

```
    }
```

```
}
```

```
sum = 0;
```

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

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

```
        sum += matrix[i][j];
```

```
    }
```

```
}
```

```
printf("\nLa suma de los componentes de la matriz es: %lld\n", sum);
```

```
return 0;
```

```
}
```

2.

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

```
#define MAX_SIZE 10
```

```
int main() {
```

```
    int N;
```

```
    int matrix[MAX_SIZE][MAX_SIZE];
```

```
    printf("### Problema 2.2.2: Escritura de matriz en sentido inverso ###\n");
```

```
    printf("Ingrese el tamaño de la matriz cuadrada (N) [1-%d]: ", MAX_SIZE);
```

```
    if (scanf("%d", &N) != 1 || N <= 0 || N > MAX_SIZE) {
```

```
        printf("Error: Dimension de matriz N x N invalida.\n");
```

```
        return 1;
```

```
    }
```

```
    printf("\nIngresando elementos de la matriz %dx%d:\n", N, N);
```

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

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

```
            printf("Ingrese el elemento matrix[%d][%d]: ", i, j);
```

```
            if (scanf("%d", &matrix[i][j]) != 1) {
```

```
                printf("Error al leer el elemento.\n");
```

```
                return 1;
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("\nElementos de la matriz leídos en sentido inverso:\n");
```

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

```
        for (int j = N - 1; j >= 0; j--) {
```

```

        printf("%d ", matrix[i][j]);
    }

    printf("\n");
}

return 0; // Ejecución exitosa
}

```

3.

```

#include <stdio.h>

#define MAX_SIZE 10

int main() {
    int N, M;
    int matrix[MAX_SIZE][MAX_SIZE];
    int max;
    int fila_a_evaluar;

    printf("### Problema 2.2.3: Maximo de una fila ###\n");
    printf("Ingrese el numero de filas (N) [1-%d]: ", MAX_SIZE);
    if (scanf("%d", &N) != 1 || N <= 0 || N > MAX_SIZE) {
        printf("Error: Dimension de filas invalida.\n");
        return 1;
    }

    printf("Ingrese el numero de columnas (M) [1-%d]: ", MAX_SIZE);
    if (scanf("%d", &M) != 1 || M <= 0 || M > MAX_SIZE) {
        printf("Error: Dimension de columnas invalida.\n");
    }
}

```

```

        return 1;
    }

    printf("\nIngresando elementos de la matriz %dx%d:\n", N, M);

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            printf("Ingrese el elemento matrix[%d][%d]: ", i, j);

            if (scanf("%d", &matrix[i][j]) != 1) {
                printf("Error al leer el elemento.\n");
                return 1;
            }
        }
    }

    printf("\nIngrese el numero de la fila a evaluar (1 a %d): ", N);
    if (scanf("%d", &fila_a_evaluar) != 1 || fila_a_evaluar < 1 || fila_a_evaluar > N) {
        printf("Error: Numero de fila invalido.\n");
        return 1;
    }

    int indice_fila = fila_a_evaluar - 1;
    max = matrix[indice_fila][0];
    for (int j = 1; j < M; j++) {

        if (matrix[indice_fila][j] > max) {
            max = matrix[indice_fila][j];
        }

    }

    printf("\nEl valor maximo encontrado en la fila %d es: %d\n", fila_a_evaluar, max);

    return 0;

```

```
}
```

4.

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

```
#define MAX_ROWS 10
```

```
#define MAX_COLS 10
```

```
int main() {
```

```
    int N, M;
```

```
    int matrix[MAX_ROWS][MAX_COLS];
```

```
    int fila_i, fila_j;
```

```
    int aux;
```

```
    printf("Ingrese el numero de filas (N) [1-%d]: ", MAX_ROWS);
```

```
    if (scanf("%d", &N) != 1 || N <= 0 || N > MAX_ROWS) return 1;
```

```
    printf("Ingrese el numero de columnas (M) [1-%d]: ", MAX_COLS);
```

```
    if (scanf("%d", &M) != 1 || M <= 0 || M > MAX_COLS) return 1;
```

```
    printf("\nIngresando elementos de la matriz %dx%d:\n", N, M);
```

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

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

```
            printf("Ingrese el elemento matrix[%d][%d]: ", i, j);
```

```
            if (scanf("%d", &matrix[i][j]) != 1) return 1;
```

```
        }
```

```
    }
```

```
    printf("\nIngrese el numero de la primera fila a intercambiar (i, 1 a %d): ", N);
```

```
    if (scanf("%d", &fila_i) != 1 || fila_i < 1 || fila_i > N) return 1;
```

```
    printf("Ingrese el numero de la segunda fila a intercambiar (j, 1 a %d): ", N);
```

```

if (scanf("%d", &fila_j) != 1 || fila_j < 1 || fila_j > N) return 1;

int index_i = fila_i - 1;
int index_j = fila_j - 1;)
for (int k = 0; k < M; k++) {

    aux = matrix[index_i][k];
    matrix[index_i][k] = matrix[index_j][k];
    matrix[index_j][k] = aux;
}

printf("\nMatriz despues de intercambiar la fila %d y la fila %d:\n", fila_i, fila_j);
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        printf("%5d", matrix[i][j]);
    }
    printf("\n");
}

return 0;
}

```