Universidad de las FF AA “ESPE”

Integrantes: Leandro Safla, Viviana Bonilla, Marco Chuquisala

Grupo: N# 5

Fecha: 06/06/2025

NRC: 20823

Trabajo en clase 2

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 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 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;

Requisitos funcionales:

1. El programa debe solicitar el tamaño n de una matriz cuadrada.
2. El programa debe permitir al usuario ingresar todos los elementos de la matriz.
3. El programa debe mostrar la matriz original.
4. El programa debe solicitar al usuario dos índices de fila i y j ($0 \leq i, j < n$).
5. El programa debe validar que i y j estén dentro de los límites válidos.
6. El programa debe intercambiar correctamente las filas i y j sin perder información.
7. El programa debe mostrar la matriz después del intercambio.

Tabla de objetos:

Objeto	Nombre	Valor	Tipo
Dato 1	n	Variable	Entero
Matriz	matriz	Variable	Entero (matriz)
Fila	i	Variable	Entero
Columna	j	Variable	Entero
Matriz intercambiada	K	Variable	Entero
Fila	r	Variable	Entero
Columna	c	Variable	Entero
Aux	aux	Variable	Entero

Código Pseint:

```
Proceso IntercambiarFilasMatriz
    Definir n, i, j, r, c, k, aux Como Entero
    Escribir "Introduce el tamaño de la matriz (n x n): "
    Leer n
    Dimension matriz[n, n]
            Dimension matriz[r, c]
    // Ingreso de los elementos de la matriz
    Escribir "Introduce los elementos de la matriz (", n, " x ", n, "):"
    Para r <- 1 Hasta n
        Para c <- 1 Hasta n
            Escribir "Elemento [", r, "][", c, "]: "
            Leer matriz[r, c]
        FinPara
    FinPara
    Escribir ""
    Escribir "Matriz original:"
    Para r <- 1 Hasta n
        Para c <- 1 Hasta n
            Escribir matriz[r, c], " ", Sin Saltar
        FinPara
        Escribir ""
    FinPara
```

```
    Escribir ""
    Escribir "Introduce los números de las dos filas a intercambiar (0 a ", n - 1, "): "
    Leer i, j
    Si i < 0 O i >= n O j < 0 O j >= n Entonces
        Escribir "Índices fuera de rango. Terminando el programa."
FinSi
Para k <- 0 Hasta n - 1
            aux <- matriz[i, k]
            matriz[i, k] <- matriz[j, k]
            matriz[j, k] <- aux
FinPara
Escribir ""
Escribir "Matriz después de intercambiar las filas ", i, " y ", j, ":"
Para r <- 0 Hasta n - 1
            Para c <- 0 Hasta n - 1
                Escribir matriz[r, c], Sin Saltar
            FinPara
            Escribir ""
FinPara
FinProceso
```

Código en C:

```c
#include <stdio.h>

int main() {
    int n;

    printf("Introduce el tamano de la matriz (n x n): "); // Solicitar el tamaño de la matriz cuadrada
    scanf("%d", &n);

    int matriz[n][n];

    printf("Introduce los elementos de la matriz (%d x %d):\n", n, n);// Ingreso de los elementos de la matriz
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("Elemento [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    printf("\nMatriz original:\n"); // Mostrar la matriz original
    for (int r = 0; r < n; r++) {
        for (int c = 0; c < n; c++) {
            printf("%4d", matriz[r][c]);
        }
        printf("\n");
    }

    int i, j;

    printf("\nIntroduce los numeros de las dos filas a intercambiar (0 a %d): ", n - 1);// Solicitar las filas a intercambiar
    scanf("%d %d", &i, &j);

    if (i < 0 || i >= n || j < 0 || j >= n) { // Validar índices
        printf("Indices fuera de rango. Terminando el programa.\n");
        return 1;
    }

    for (int k = 0; k < n; k++) { // Intercambiar las filas i y j
        int aux = matriz[i][k];
        matriz[i][k] = matriz[j][k];
        matriz[j][k] = aux;
    }

    printf("\nMatriz despues de intercambiar las filas %d y %d:\n", i, j); // Mostrar la matriz después del intercambio
    for (int r = 0; r < n; r++) {
        for (int c = 0; c < n; c++) {
            printf("%4d", matriz[r][c]);
        }
        printf("\n");
    }

    return 0;
}
```

Prueba de escritorio:

```
Introduce el tamano de la matriz (n x n): 2
Introduce los elementos de la matriz (2 x 2):
Elemento [0][0]: 15
Elemento [0][1]: 69
Elemento [1][0]: 45
Elemento [1][1]: 36

Matriz original:
  15  69
  45  36

Introduce los numeros de las dos filas a intercambiar (0 a 1): 1 0

Matriz despues de intercambiar las filas 1 y 0:
  45  36
  15  69

Process returned 0 (0x0)   execution time : 13.678 s
Press any key to continue.
```

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 que tenga el 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 que el valor inicial se le da a la variable que va a almacenar el máximo (max). Imagine que se asume que 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, cuyo caso el valor del máximo calculo 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 inicializar) y proceder a continuación con las comparaciones como se ha indicado. De este modo no se fuerza ninguna suposición sobre el rango de los valores donde se encuentran las componentes del vector.

Requisitos funcionales

1. El programa debe permitir ingresar el tamaño N de la matriz cuadrada N x N con valores enteros.
2. El programa debe permitir validar que el tamaño N sea un número entero positivo antes de continuar.
3. El programa debe permitir al usuario ingresar los elementos de la matriz fila por fila y columna por columna.
4. El programa debe permitir al usuario ingresar el número de una fila específica para ser analizada.
5. El programa debe permitir validar que el número de fila ingresado esté dentro del rango válido (0 a N-1).
6. El programa debe permitir calcular el valor máximo de la fila indicada sin hacer suposiciones sobre el rango de los valores, inicializando la variable max con la primera componente de dicha fila.
7. El programa debe permitir mostrar por pantalla el valor máximo encontrado en la fila seleccionada.

Tabla de objetos

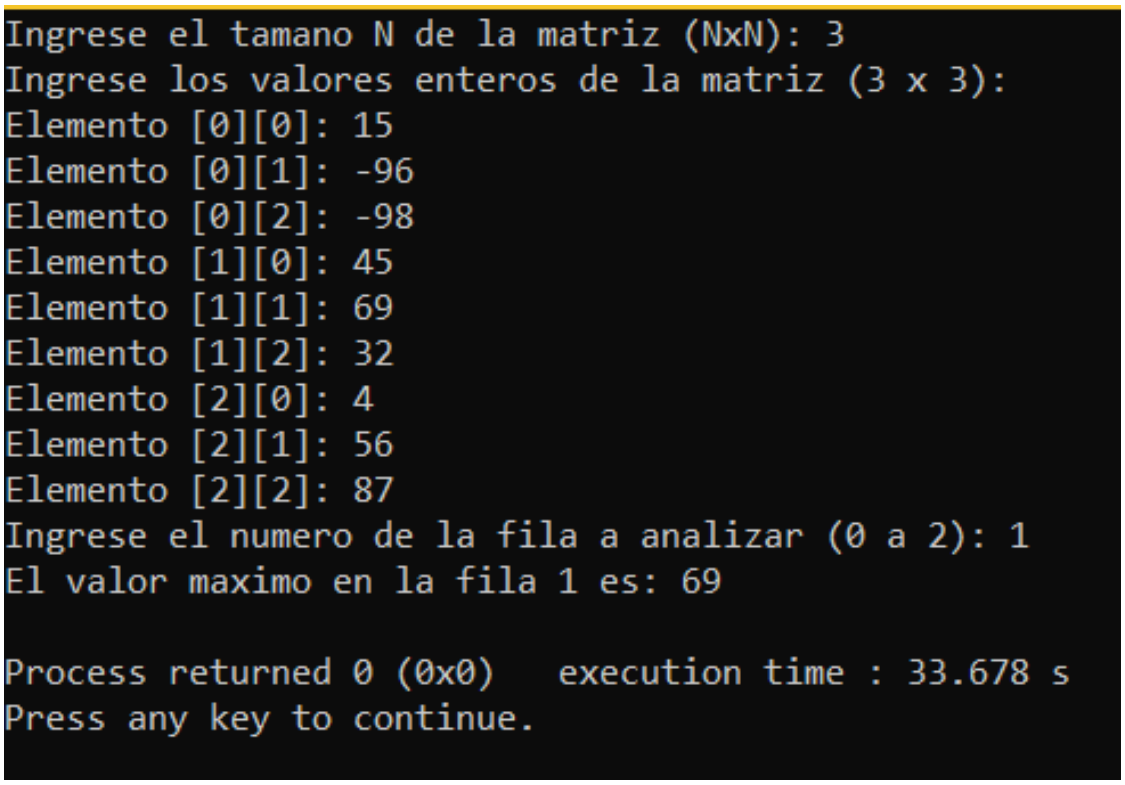
Objeto	Nombre	Valor	Tipo
N	Tamaño de la matriz	variable	Entero
Matriz [][]	Matriz de números	Variable	Entero
i	Índice de fila (bucle externo)	Variable	Entero
j	Índice de columna (bucle)	Variable	Entero
fila	Fila seleccionada por el usuario	Variable	Entero
max	Máximo valor de una fila	Variable	Entero

Código en C

```c
#include <stdio.h>
int main() {
    int N;

    // RF1: El programa debe permitir ingresar el tamaño N de la matriz
    printf("Ingrese el tamano N de la matriz (NxN): ");
    scanf("%d", &N);

    // RF2: El programa debe permitir validar que el tamaño sea un entero positivo
    if (N <= 0) {
        printf("El tamano debe ser un numero entero positivo.\n");
        return 1;
    }

    int matriz[N][N];

    // RF3: El programa debe permitir ingresar los elementos de la matriz
    printf("Ingrese los valores enteros de la matriz (%d x %d):\n", N, N);
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("Elemento [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    int fila;

    // RF4: El programa debe permitir ingresar el número de una fila
    printf("Ingrese el numero de la fila a analizar (0 a %d): ", N - 1);
    scanf("%d", &fila);

    // RF5: El programa debe permitir validar que la fila esté en el rango valido
    if (fila < 0 || fila >= N) {
        printf("Número de fila invalido.\n");
        return 1;
    }
```

```c
    // RF6: El programa debe permitir calcular el valor maximo de la fila
    int max = matriz[fila][0];
    for (int j = 1; j < N; j++) {
        if (matriz[fila][j] > max) {
            max = matriz[fila][j];
        }
    }

    // RF7: El programa debe permitir mostrar el resultado
    printf("El valor maximo en la fila %d es: %d\n", fila, max);

    return 0;
}
```

Prueba de escritorio



Código Pseint

```
Proceso MaximoValorFila
   Definir N, i, j, fila, max Como Entero
   Definir matriz Como Entero
   Escribir "Ingrese el tamaño de la matriz N x N:"
   Leer N
   Dimension matriz[N, N]

   // Leer la matriz
   Para i <- 0 Hasta N - 1
      Para j <- 0 Hasta N - 1
         Escribir "Ingrese el valor en la posición [", i, ",", j, "]:"
         Leer matriz[i, j]
      FinPara
   FinPara
   // Solicitar el número de fila (verificar que esté en rango)
```

```
    Repetir
        Escribir "Ingrese el número de fila que desea consultar (de 0 a ", N - 1, "):"
        Leer fila
    Hasta Que fila >= 0 Y fila < N
    // Inicializar max con el primer valor de la fila
    max <- matriz[fila, 0]
    // Recorrer la fila para encontrar el máximo
    Para j <- 1 Hasta N - 1
        Si matriz[fila, j] > max Entonces
            max <- matriz[fila, j]
        FinSi
    FinPara
    // Mostrar el resultado
    Escribir "El valor máximo en la fila ", fila, " es: ", max
FinProceso
```

Dada una matriz de N × N elementos, realice un algoritmo que recorra la matriz por filas desde la última a la primera y cada fila en sentido inverso, 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 de las variables i y j comienzan en la última fila/columna de la matriz. La condición de permanencia en los bucles es ahora

i >= 0 o j >= 0 (en C: i >= 0, j >= 0)

Las variables se decrementan en cada iteración.

A continuación, se muestra el diagrama de flujo de la solución en la figura 2.11 así como la tabla de objetos y codificación en C³.

Requisitos funcionales

1. El programa debe solicitar al usuario ingresar el tamaño de una matriz cuadrada
2. El programa debe solicitar al usuario que ingrese cada elemento de la matriz empezando debe la última fila y columna hasta la primera.
3. El programa debe mostrar en pantalla la matriz ingresada

Prueba de escritorio

Objeto	**Nombre**	**Valor**	**Tipo**
N	Tamaño de la matriz	variable	Entero
Matriz [][]	Matriz de números	Variable	Entero
i	Índice de fila	Variable	Entero
j	Índice de columna	Variable	Entero

[*] Matriz en sentido inverso.cpp

```c
#include <stdio.h>

int main() {
    int n;

    // Solicitar el tamano de la matriz cuadrada
    printf("Ingrese el tamano de la matriz cuadrada (n x n): ");
    scanf("%d", &n);

    int matriz[n][n];

    printf("Ingrese los elementos de la matriz desde la ultima fila y columna hasta la primera:\n");

    // Entrada de la matriz en orden inverso (de abajo hacia arriba y de derecha a izquierda)
    for (int i = n - 1; i >= 0; i--) {
        for (int j = n - 1; j >= 0; j--) {
            printf("Elemento [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    // Mostrar la matriz
    printf("\nMatriz ingresada:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d\t", matriz[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Código c

```
Ingrese el tamano de la matriz cuadrada (n x n): 4
Ingrese los elementos de la matriz desde la ultima fila y columna hasta la primera:
Elemento [3][3]: 23
Elemento [3][2]: 5
Elemento [3][1]: 34
Elemento [3][0]: 64
Elemento [2][3]: 4
Elemento [2][2]: 6
Elemento [2][1]: 23
Elemento [2][0]: 6
Elemento [1][3]: 23
Elemento [1][2]: 43
Elemento [1][1]: 3
Elemento [1][0]: 34
Elemento [0][3]: 6
Elemento [0][2]: 5
Elemento [0][1]: 56
Elemento [0][0]: 7

Matriz ingresada:
7       56      5       6
34      3       43      23
6       23      6       4
64      34      5       23

--------------------------------
Process exited after 12.65 seconds with return value 0
Presione una tecla para continuar . . .
```

Prueba de escritorio

código Pseint

```
Proceso Matriz_Inversa
   Definir n, i, j Como Entero
   Definir matriz Como Entero
   Escribir "Ingrese el tamaño de la matriz cuadrada (n x n): "
   Leer n
   Dimension matriz[n, n]
   Escribir "Ingrese los elementos de la matriz desde la ultima fila y columna hasta la primera:
   Para i <- n - 1 Hasta 0 Con Paso -1
      Para j <- n - 1 Hasta 0 Con Paso -1
         Escribir "Elemento [", i, "][", j, "]: "
         Leer matriz[i, j]
      FinPara
   FinPara

   Escribir ""
   Escribir "Matriz ingresada:"
   Para i <- 0 Hasta n - 1
      Para j <- 0 Hasta n - 1
         Escribir Sin Saltar matriz[i, j], "  "
      FinPara
      Escribir ""
   FinPara
FinProceso
```