



## **UNIDAD 4.- CONCEPTOS DE DATOS ESTRUCTURADOS – ARREGLOS BIDIMENSIONALES: MATRICES**

### **OBJETIVOS:**

- Que comprenda el concepto de estructura de datos y describa las aplicaciones adecuadas de los arreglos unidimensionales, bidimensionales y multidimensionales.
- Que represente adecuadamente las estructuras de arreglo y registro.

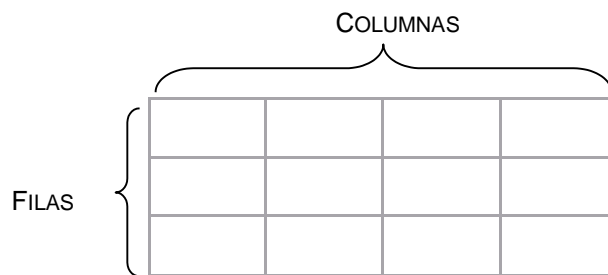
### **TEMAS:**

- 4.1.** Definición, lectura e impresión. Representación en lenguaje C.
- 4.2.** Operaciones (suma, resta, multiplicación de un escalar por una matriz, multiplicación de matrices).
- 4.3.** Operaciones por fila, operaciones por columna.
- 4.4.** Búsqueda.
- 4.5.** Ordenamiento.
- 4.6.** Tipos de matrices.
- 4.7.** Elementos característicos de una matriz.

#### 4.1. DEFINICIÓN. LECTURA E IMPRESIÓN. Representación en Lenguaje C.

Una **matriz** es un arreglo bidimensional, por lo tanto es un dato de tipo estructurado estático, con la particularidad de que al ser un arreglo todos los datos son de la misma naturaleza (son homogéneos), y además presenta la característica de que sus elementos se almacenan en la memoria RAM; por lo que al apagar la computadora o cuando termina el programa éstos se pierden.

Las dos dimensiones que poseen son: filas y columnas.



Aquí es fundamental poder distinguir:

- **Orden de la Matriz:** indica la cantidad de elementos.
- **Nombre de la Matriz:** indica la forma de distinguir una matriz de otra; por lo que dos matrices serán distintas si poseen nombres diferentes y no subíndices diferentes.
- **Subíndices:** indican la posición de un elemento, en Lenguaje C, pueden comenzar en el valor 0 (cero).

En este caso se posee:

- Subíndice para las filas: que le llamaremos **i**.
- Subíndice para las columnas: que le llamaremos **j**.
- Si se necesitan indicar dos elementos, estos subíndices pueden representarse con: números, variables, o con fórmulas.

Por ejemplo:

- $A[i][j]$  representa el elemento de la  $i$ -ésima fila y de la  $j$ -ésima columna de la matriz A.
- $A[2][3]$  representa el elemento de la fila 2 y de la columna 3 de la matriz A.

Por ejemplo si se tiene una matriz como la siguiente:



**N=3** cantidad de columnas

**M=5** cantidad de filas

**A[i][j]** elemento de la matriz

- ✓ M es la cantidad de las filas de la matriz.
- ✓ N es la cantidad de columnas de la matriz.
- ✓ A es el nombre de la matriz.
- ✓ Con **i** se indicará el subíndice de las filas.
- ✓ Con **j** se indicará el subíndice de las columnas.
- ✓ **A[i][j]** cada elemento de la matriz.

Otro concepto a tener en claro, es que al ser una matriz un conjunto de datos, tanto su lectura como su impresión se realizan con estructuras de iteración anidadas cuyos cortes de control serán uno para las filas y uno para las columnas.

Sólo se podrá realizar la impresión o la lectura fuera del ciclo de repetición si lo que se pretende leer o imprimir es un único valor (correspondiente a un elemento de la matriz).

### **LECTURA E IMPRESIÓN POR FILAS**

Para leer una matriz por filas el primer ciclo de repetición debe ser para las filas, y el segundo e interno será para las columnas.

Es necesario aclarar que no importa si se lee la matriz por filas o por columnas, cuando se lee el elemento de la matriz siempre el primer subíndice es el que corresponde a las filas y el segundo es el que corresponde a las columnas.

### **Datos**

**M:** cantidad de filas de la matriz.

**N:** cantidad de columnas de la matriz.

**A:** nombre de la matriz.

**i:** subíndice de las filas de la matriz.

**j:** subíndice de las columnas de la matriz.

**A[i][j]:** cada uno de los elementos de la matriz.



### **PROGRAMA PARA LECTURA E IMPRESIÓN POR FILAS**

```
#include<stdio.h>

main()
{
    int i,j,n,m;
    int a[100][100];

    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);
    printf("ingrese la cantidad de columnas: ");
    scanf("%d",&n);

    //primero ciclo de i --> por filas
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("a[%d][%d]=",i,j);
            scanf("%d",&a[i][j]);
        }
    }
}
```

### **LECTURA E IMPRESIÓN POR COLUMNAS**

Para leer una matriz por columnas el primer ciclo de repetición debe ser para las columnas, y el segundo e interno será para las filas.

#### **Datos**

**M:** cantidad de filas de la matriz.

**N:** cantidad de columnas de la matriz.

**A:** nombre de la matriz.

**i:** subíndice de las filas de la matriz.

**j:** subíndice de las columnas de la matriz.

**A[i][j]:** cada uno de los elementos de la matriz.



### **PROGRAMA PARA LECTURA E IMPRESIÓN POR COLUMNAS**

```
#include<stdio.h>

main()
{

    int i,j,n,m;
    int a[100][100];

    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);
    printf("ingrese la cantidad de columnas: ");
    scanf("%d",&n);

    //primero ciclo de j --> por columnas
    for(j=0;j<n;j++)
    {
        for(i=0;i<m;i++)
        {
            printf("a[%d][%d]=",i,j);
            scanf("%d",&a[i][j]);
        }
    }
}
```

### **4.2. OPERACIONES CON MATRICES**

Los trabajos que se pueden desarrollar con los elementos de las matrices pueden ser los siguientes:

- Suma de los elementos de una matriz, de todos, por fila o por columnas.
- Promedio de los elementos de una matriz, de todos, por fila o por columnas.
- Porcentaje de los elementos de una matriz, de todos, por fila o por columnas.
- Mayor elemento de la matriz, por fila o por columnas y la posición del mismo.
- Menor elemento de la matriz, por fila o por columnas y la posición del mismo.
- Generación de matrices.
- Matrices especiales: nula, identidad, triangular superior, triangular inferior,....
- Elementos característicos de las matrices: elementos de la cruz, diagonal principal, diagonal secundaria,....
- Ordenamiento de los elementos de la matriz.
- Búsqueda de un elemento en la matriz.



Las operaciones que se pueden realizar son:

- Suma de dos matrices.
- Resta de dos matrices.
- Multiplicación de un escalar por una matriz.
- Multiplicación de dos matrices.

## 1. SUMA DE DOS O MÁS MATRICES

Para sumar dos o más matrices, las mismas deben ser del mismo orden, es decir deben tener la misma cantidad de elementos.

Supongamos dos matrices A y B de orden MxN, se ingresan los elementos de la matriz A y de la matriz B, luego se realiza la suma en el correspondiente elemento de la matriz C.

Por ejemplo:

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1+1 & 3+0 & 2+5 \\ 1+7 & 0+5 & 0+0 \\ 1+2 & 2+1 & 2+1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 7 \\ 8 & 5 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

En informática en forma genérica:  $C[i][j]=A[i][j]+B[i][j]$

## PROGRAMA EN LENGUAJE C PARA INGRESAR Y SUMAR DOS MATRICES

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

```
main ()
```

```
{
```

```
    int i,j,n,m;
```

```
    int a[10][10],b[10][10],c[10][10];
```

```
    printf("ingrese la cantidad de filas: ");
```

```
    scanf("%d",&m);
```

```
    printf("ingrese la cantidad de columnas: ");
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<m;i++)
```

```
    {
```

```
        for(j=0;j<n;j++)
```

```
{
    printf("Ingrese el elemento a[%d][%d] = ",i,j);
    scanf("%d",&a[i][j]);
    printf("Ingrese el elemento b[%d][%d] = ",i,j);
    scanf("%d",&b[i][j]);
    c[i][j]=a[i][j]+b[i][j];
}
}

printf("\nLa matriz resultante de la suma es:\n ");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("Elemento c[%d][%d] = %d\n",i,j,c[i][j]);
    }
}
```

## 2. RESTA DE DOS O MÁS MATRICES

Para restar dos o más matrices, las mismas deben ser del mismo orden, es decir deben tener la misma cantidad de elementos.

Supongamos dos matrices A y B de orden MxN, se ingresan los elementos de la matriz A y de la matriz B, luego se realiza la suma en el correspondiente elemento de la matriz C.

En informática en forma genérica:  $C[i][j] = A[i][j] - B[i][j]$

### Programa en Lenguaje C para ingresar y restar dos matrices

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

```
main ()
{
    int i,j,n,m;
    int a[10][10],b[10][10],c[10][10];

    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);

    printf("ingrese la cantidad de columnas: ");
```

```
scanf("%d",&n);

for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("Ingrese el elemento a[%d][%d] = ",i,j);
        scanf("%d",&a[i][j]);
        printf("Ingrese el elemento b[%d][%d] = ",i,j);
        scanf("%d",&b[i][j]);
        c[i][j]=a[i][j]-b[i][j];
    }
}

printf("\nLa matriz resultante de la suma es:\n ");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("Elemento c[%d][%d] = %d\n",i,j,c[i][j]);
    }
}
}
```

### 3. MULTIPLICACIÓN DE UN ESCALAR POR UNA MATRIZ

Para multiplicar una escalar **k** por una matriz A, se debe multiplicar cada elemento de la matriz por el escalar, generando de esta forma una nueva matriz C con el resultado de la información.

Por ejemplo:

$$2 \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 5 \end{bmatrix} = \begin{bmatrix} 2 \times 1 & 2 \times 8 & 2 \times -3 \\ 2 \times 4 & 2 \times -2 & 2 \times 5 \end{bmatrix} = \begin{bmatrix} 2 & 16 & -6 \\ 8 & -4 & 10 \end{bmatrix}$$

En forma genérica:  $C[i][j] = k * A[i][j]$



**PROGRAMA EN LENGUAJE C PARA INGRESAR Y MULTIPLICAR UN ESCALAR POR UNA MATRIZ**

```
#include<stdio.h>

main ()
{
    int i,j,n,m,k;
    int a[10][10],c[10][10];

    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);
    printf("ingrese la cantidad de columnas: ");
    scanf("%d",&n);
    printf("ingrese el valor de k: ");
    scanf("%d",&k);
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("Ingrese el elemento a[%d][%d] = ",i,j);
            scanf("%d",&a[i][j]);
            c[i][j]=k*a[i][j];
        }
    }

    printf("\nLa matriz resultante de la multiplicación por el escalar es:\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("Elemento c[%d][%d] = %d\n",i,j,c[i][j]);
        }
    }
}
```

#### **4. MULTIPLICACIÓN DE MATRICES**

Dadas la matriz A de orden  $M \times P$  y B de orden  $Q \times N$ , la multiplicación de ambas sólo es posible si  $P=Q$ , y donde la matriz resultante C será de orden  $M \times N$ ; en donde el elemento  $c[0][0]$  será la suma de los productos de los elementos de la primera fila de A por la primera columna de B.

Por ejemplo:

$$\begin{pmatrix} 1 & 1 & 0 \\ 2 & -1 & 2 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 5 \\ 3 & -1 & 3 \\ 5 & 2 & 1 \end{pmatrix} =$$
$$= \begin{pmatrix} 1+3+0 & 2-1+0 & 5+3+0 \\ 2-3+10 & 4+1+4 & 10-3+2 \\ 0+9+0 & 0-3+0 & 0+9+0 \end{pmatrix} =$$
$$= \begin{pmatrix} 4 & 1 & 8 \\ 9 & 9 & 9 \\ 9 & -3 & 9 \end{pmatrix}$$

En forma genérica:  $C[i][j] = \sum A[i][k] * B[k][j]$  para  $k=1..P$

### **PROGRAMA EN LENGUAJE C PARA INGRESAR Y MULTIPLICAR DOS MATRICES**

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

```
main ()
```

```
{  
    int i,j,k,m,p,q,n;  
    int a[10][10],b[10][10],c[10][10];  
    printf("ingrese la cantidad de filas de la matriz A: ");  
    scanf("%d",&m);  
    printf("ingrese la cantidad de columnas de la matriz A: ");  
    scanf("%d",&p);  
    printf("ingrese la cantidad de filas de la matriz B: ");  
    scanf("%d",&q);  
    printf("ingrese la cantidad de columnas de la matriz B: ");  
    scanf("%d",&n);  
  
    for(i=0;i<m;i++)  
    {  
        for(j=0;j<p;j++)  
        {  
            printf("Ingrese el elemento a[%d][%d] = ",i,j);  
            scanf("%d",&a[i][j]);  
        }  
    }  
  
    printf("\n\n");  
    for(i=0;i<q;i++)  
    {  
        for(j=0;j<n;j++)  
        {  
            printf("Ingrese el elemento b[%d][%d] = ",i,j);
```

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

printf("\n\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        c[i][j]=0;
        for(k=0;k<p;k++)
        {
            c[i][j]=c[i][j]+a[i][k]*b[k][j];
        }
    }
}

printf("\nLa matriz resultante de la multiplicacion es:\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("Elemento c[%d][%d] = %d\n",i,j,c[i][j]);
    }
}
}
```

#### 4.3. OPERACIONES POR FILA Y POR COLUMNA.

##### OPERACIONES CON TODOS LOS ELEMENTOS DE LA MATRIZ

Cuando **se desea trabajar con todos los elementos de la matriz** la **inicialización** de las variables **se realiza antes de los ciclos de repetición** y los **resultados se muestran una vez que finalizan ambos**.

##### Ejemplo:

Dada la matriz A de orden MxN, ingrese los valores de la misma y realice la suma de los mismos. Muestre este resultado.

*En este caso se necesita un acumulador que se inicializará antes de los ciclos y el resultado se mostrará al final.*



### **PROGRAMA PARA SUMAR TODOS LOS ELEMENTOS DE UNA MATRIZ**

```
#include<stdio.h>

main()
{
    int i,j,n,m,s;

    int a[100][100];

    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);
    printf("ingrese la cantidad de columnas: ");
    scanf("%d",&n);

    //trabajar con TODOS los elementos
    s=0;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("a[%d][%d]=",i,j);
            scanf("%d",&a[i][j]);
            s=s+a[i][j];
        }
    }

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

### **OPERACIONES POR FILA**

Cuando **se desea trabajar con los elementos de las filas** la **inicialización** de las variables **se realiza entre los ciclos de repetición** y los **resultados se muestran una vez que finaliza el ciclo interno**.

#### **Ejemplo:**

Dada la matriz A de orden MxN, ingrese los valores de la misma y realice la suma de los elementos de cada fila. Muestre los resultados.



En este caso se necesita un acumulador por cada fila, éste se **inicializará entre los dos ciclos** y los **resultados se mostrarán cuando termine el ciclo interno**.

PROGRAMA PARA SUMAR LOS ELEMENTOS DE CADA FILA

```
#include<stdio.h>

main()
{
    int i,j,n,m,s;

    int a[100][100];

    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);
    printf("ingrese la cantidad de columnas: ");
    scanf("%d",&n);

    for(i=0;i<m;i++)
    {
        //trabajar POR FILA
        s=0;
        for(j=0;j<n;j++)
        {
            printf("a[%d][%d]=",i,j);
            scanf("%d",&a[i][j]);
            s=s+a[i][j];
        }
        printf("\nLa suma de los elementos de la fila %d es: %d\n\n",i,s);
    }
}
```

En este caso también se puede realizar la suma los elementos de cada fila generando un vector donde cada elemento contendrá la suma de los elementos de cada fila. Este vector tendrá orden M.

3	2
6	5
3	20

5
11
23



Ejemplo:

Dada la matriz A de orden MxN, ingrese los valores de la misma y realice la suma de los elementos de cada fila mediante un vector. Muestre los resultados.

*En este caso se necesita un acumulador por cada fila, éste será un vector que se **inicializará cada elemento del mismo entre los dos ciclos** y los **resultados se mostrarán cuando termine el ciclo interno**.*

**PROGRAMA PARA SUMAR LOS ELEMENTOS DE CADA FILA**

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

```
main()
```

```
{
```

```
    int i,j,n,m;
```

```
    int a[100][100],v[100];
```

```
    printf("ingrese la cantidad de filas: ");
```

```
    scanf("%d",&m);
```

```
    printf("ingrese la cantidad de columnas: ");
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<m;i++)
```

```
    {
```

```
        //trabajar POR FILA con un VECTOR
```

```
        v[i]=0;
```

```
        for(j=0;j<n;j++)
```

```
        {
```

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

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

```
            v[i]=v[i]+a[i][j];
```

```
        }
```

```
        printf("\nLa suma de los elementos de la fila %d es: %d\n\n",i,v[i]);
```

```
    }
```

```
}
```



### **OPERACIONES POR COLUMNA**

Cuando **se desea trabajar con los elementos de las columnas** la **inicialización** de las variables **se realiza entre los ciclos de repetición** y los **resultados se muestran una vez que finaliza el ciclo interno**.

#### **Ejemplo:**

Dada la matriz A de orden MxN, ingrese los valores de la misma y realice la suma de los elementos de cada columna. Muestre los resultados.

*En este caso se necesita un acumulador por cada columna, éste se **inicializará entre los dos ciclos** y los **resultados se mostrarán cuando termine el ciclo interno**.*

#### **PROGRAMA PARA SUMAR LOS ELEMENTOS DE CADA COLUMNA**

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

```
main()  
{
```

```
    int i,j,n,m,s;
```

```
    int a[100][100];
```

```
    printf("ingrese la cantidad de filas: ");
```

```
    scanf("%d",&m);
```

```
    printf("ingrese la cantidad de columnas: ");
```

```
    scanf("%d",&n);
```

```
    for(j=0;j<n;j++)
```

```
    {
```

```
        //trabajar por coulumna
```

```
        s=0;
```

```
        for(i=0;i<m;i++)
```

```
        {
```

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

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

```
            s=s+a[i][j];
```

```
        }
```

```
        printf("\nLa suma de los elementos de la columna %d es: %d\n\n",j,s);
```

```
    }
```

```
}
```

En este caso también se puede realizar la suma los elementos de cada columna generando un vector donde cada elemento contendrá la suma de los elementos de cada



columna. Este vector tendrá orden N.

3	2
6	5
3	20

12	27
----	----

**Ejemplo:**

Dada la matriz A de orden MxN, ingrese los valores de la misma y realice la suma de los elementos de cada columna mediante un vector. Muestre los resultados.

*En este caso se necesita un acumulador por cada columna, éste será un vector que se **inicializará cada elemento del mismo entre los dos ciclos** y los **resultados se mostrarán cuando termine el ciclo interno**.*

**PROGRAMA PARA SUMAR LOS ELEMENTOS DE CADA COLUMNA**

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

```
main()  
{
```

```
    int i,j,n,m;
```

```
    int a[100][100],v[100];
```

```
    printf("ingrese la cantidad de filas: ");  
    scanf("%d",&m);  
    printf("ingrese la cantidad de columnas: ");  
    scanf("%d",&n);
```

```
    for(j=0;j<n;j++)  
    {  
        //trabajar por coulumna con un Vector  
  
        v[j]=0;  
        for(i=0;i<m;i++)  
        {  
            printf("a[%d][%d]=",i,j);  
            scanf("%d",&a[i][j]);
```



```
        v[j]=v[j]+a[i][j];
    }
    printf("\nLa suma de los elementos de la columna %d es: %d\n\n",j,v[j]);
}
}
```

#### 4.4. MÉTODO DE BÚSQUEDA.

Dada una matriz A de orden MxN y un elemento k, el método de búsqueda de determinar si dicho elemento se encuentra en la matriz o no. Para este fin se utiliza una variable bandera que se inicializará con el valor 0 (cero) y cambiará si el valor k buscado se encuentra dentro de los elementos de la matriz.

Si la matriz **contiene elementos repetidos** cuando se cambia el valor de la bandera también **se debe** mostrar la posición donde se encuentra cada elemento igual a k.

Si la matriz contiene **NO contiene elementos repetidos** cuando se cambia el valor de la bandera también **se puede** mostrar la posición donde se encuentra el elemento encontrado, o se puede mostrarlo al final previamente guardado en una variable.

#### PROGRAMA PARA BUSCAR EL VALOR K DENTRO DE LOS ELEMENTOS DE LA MATRIZ SIN ELEMENTOS REPETIDOS

```
#include<stdio.h>

main ()
{
    int i,j,n,m,b,k;
    int a[100][100];

    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);
    printf("ingrese la cantidad de columnas: ");
    scanf("%d",&n);

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("Ingrese el elemento a[%d][%d] = ",i,j);
            scanf("%d",&a[i][j]);
        }
    }
}
```



```
printf("\n\n");
printf("ingrese el valor k que desea buscar: ");
scanf("%d",&k);

b=0;
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        if (a[i][j]==k)
        {
            b=1;
        }
    }
}

if (b==1)
{
    printf("\n\nEl valor %d se encuentra en la matriz",k);
}
else
{
    printf("\n\nEl valor %d NO se encuentra en la matriz",k);
}
}
```

**PROGRAMA PARA BUSCAR EL VALOR K DENTRO DE LOS ELEMENTOS DE LA MATRIZ CON ELEMENTOS REPETIDOS**

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

```
main ()
{
```

```
    int i,j,n,m,b,k;
    int a[100][100];
```

```
    printf("ingrese la cantidad de filas: ");
    scanf("%d",&m);
    printf("ingrese la cantidad de columnas: ");
    scanf("%d",&n);
```

```
    for(i=0;i<m;i++)
```

```
{
    for(j=0;j<n;j++)
    {
        printf("Ingrese el elemento a[%d][%d] = ",i,j);
        scanf("%d",&a[i][j]);
    }
}

printf("\n\n");
printf("ingrese el valor k que desea buscar: ");
scanf("%d",&k);

b=0;
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        if (a[i][j]==k)
        {
            printf("\n\nEl valor %d se encuentra en la fila:%d columna:%d\n",k,i,j);
            b=1;
        }
    }
}

if (b==0)
{
    printf("\n\nEl valor %d NO se encuentra en la matriz",k);
}
}
```

#### 4.5. MÉTODO DE ORDENAMIENTO.

Dada una matriz A de orden MxN, se pueden ordenar los valores de la matriz.

##### **ORDENAR TODOS LOS ELEMENTOS DE LA MATRIZ**

Dada una matriz A de orden MxN, se puede ordenar teniendo en cuenta los elementos de la primera fila, en este caso los elementos quedarán ordenados por columnas. Para ordenar los elementos de cada columna se utilizará un método de ordenamiento de vectores, en este caso se utiliza el método de la burbuja.

Se debe tener en cuenta también que se puede ordenar en forma ascendente o descendente teniendo en cuenta los operadores de relación que se deben usar en cada caso.

Ejemplo:



3	8	6	11
16	15	13	5
1	4	9	23

Ordenada:

1	3	4	5
6	8	9	11
13	15	16	23

**PROGRAMA PARA ORDENAR TODOS LOS ELEMENTOS DE LA MATRIZ**

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

```
main ()
```

```
{    int i,j,n,m,b,aux;
```

```
    int a[100][100];
```

```
    printf("ingrese la cantidad de filas: ");
```

```
    scanf("%d",&m);
```

```
    printf("ingrese la cantidad de columnas: ");
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<m;i++)
```

```
    {
```

```
        for(j=0;j<n;j++)
```

```
        {
```

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

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

```
        }
```

```
    }
```

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

```
    printf("COMIENZA EL ORDENAMIENTO\n\n");
```

```
    do
```

```
    {
```

```
        b=0;
```

```
for(i=0;i<m;i++)
{
    for(j=0;j<n-1;j++)
    {
        if (a[i][j]>a[i][j+1])
        {
            aux=a[i][j];
            a[i][j]=a[i][j+1];
            a[i][j+1]=aux;
            b=1;
        }
    }

    if (i<m-1)
    {
        if (a[i][n-1]>a[i+1][0])
        {
            aux=a[i][n-1];
            a[i][n-1]=a[i+1][0];
            a[i+1][0]=aux;
            b=1;
        }
    }
}
while (b==1);

printf("La matriz ordenada es:\n\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("Elemento a[%d][%d] = %d\n",i,j,a[i][j]);
    }
}
}
```

#### 4.6. TIPOS DE MATRICES.

##### MATRIZ FILA

**Matriz fila** es toda matriz rectangular con una sola fila de dimensión  $1 \times n$ .

Condiciones:

$i=0$

$j=0..n-1$

$$H = (2, 3, 4, 0, 76)$$

##### MATRIZ COLUMNA

**Matriz columna** es toda matriz rectangular con una sola columna de dimensión  $m \times 1$ .

Condiciones:

$i=0..m-1$

$j=0$

$$K = \begin{pmatrix} 5 \\ 22 \\ -3 \\ 12 \end{pmatrix}$$

##### MATRIZ NULA

Una **matriz nula** es una matriz rectangular con todos sus elementos nulos. Se denota por  $0$ .

Condiciones:

$a[i][j]=0 \quad \forall i, \forall j$

##### MATRIZ TRIANGULAR SUPERIOR

**Matriz triangular superior** es toda matriz cuadrada en la que todos los términos situados por debajo de la diagonal principal son ceros.

Condiciones:

$a[i][j]=0 \quad \forall i > j$

$a[i][j] \neq 0 \quad \forall i \leq j$

$$K = \begin{pmatrix} 1 & 5 & -2 & 11 \\ 0 & 3 & -1 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$



### **MATRIZ TRIANGULAR INFERIOR**

**Matriz triangular inferior** es toda matriz cuadrada en la que todos los términos situados por encima de la diagonal principal son ceros.

Condiciones:

$$a[i][j] \neq 0 \quad \forall i > j$$

$$a[i][j] = 0 \quad \forall i < j$$

$$K = \begin{pmatrix} 1 & 5 & -2 & 11 \\ 0 & 3 & -1 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

### **MATRIZ DIAGONAL**

**Matriz diagonal** es toda matriz cuadrada en la que todos los términos no situados en la diagonal principal son ceros.

Condiciones:

$$a[i][j] = 0$$

$$i \neq j$$

$$D = \begin{pmatrix} 7 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 8 \end{pmatrix}$$

### **MATRIZ ESCALAR**

**Matriz escalar** es toda matriz diagonal en la que todos los términos de la diagonal principal son iguales.

Condiciones:

$$a[i][j] = 0 \quad \forall i \neq j$$

$$a[i][j] = k \quad \forall i = j$$

$$F = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{pmatrix}$$

### **MATRIZ UNIDAD**

**Matriz unidad o identidad** es la matriz escalar cuyos elementos de la diagonal principal son todos 1.



Condiciones:

$$a[i][j]=0 \quad \forall i \neq j$$

$$a[i][j]=1 \quad \forall i=j$$

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

### MATRIZ TRANSPUESTA

**Matriz transpuesta** se obtiene al intercambiar los elementos por medio de su posición dentro de la matriz. Las filas de una matriz pasan a ser las columnas en la matriz transpuesta.

Condiciones:

$$a[i][j] = a[j][i]$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^t = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

### MATRIZ SIMÉTRICA

**Matriz simétrica** es aquella matriz cuadrada ( $m = n$ ) en la que  $a_{ij} = a_{ji}$  para todo  $i, j = 1, 2, 3, 4, \dots, n$ .

Condiciones:

$$a[i][j] == a[j][i] \quad \forall i \quad \forall j$$

$$H = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 8 \end{pmatrix}$$

### MATRIZ ANTISIMÉTRICA

**Matriz antisimétrica** es aquella matriz cuadrada ( $m = n$ ) en la que

Condiciones:

$$a[i][j] == -a[j][i] \quad \forall i \quad \forall j$$

$$D = \begin{pmatrix} 0 & -3 & -2 \\ 3 & 0 & -4 \\ 2 & 4 & 0 \end{pmatrix}$$



#### 4.7. ELEMENTOS CARACTERÍSTICOS DE UNA MATRIZ.

Los elementos característicos de una matriz son:

1. Diagonal principal.
2. Diagonal secundaria.
3. Cruz de la matriz.
4. Cuadro de la matriz.

##### 1. DIAGONAL PRINCIPAL

Para que una matriz contenga diagonal principal la misma debe ser una matriz cuadrada, es decir de orden  $M \times M$ .

Los elementos de esta diagonal cumplen con la siguiente condición:

$$i == j \quad \forall i \forall j$$

Ejemplo:

$$A = \begin{pmatrix} 5 & 1 & 5 & -6 \\ 1 & 2 & 0 & 3 \\ 7 & -3 & 4 & 11 \\ 1 & 9 & 3 & 8 \end{pmatrix}$$

##### 2. DIAGONAL SECUNDARIA

Para que una matriz contenga diagonal secundaria la misma debe ser una matriz cuadrada, es decir de orden  $M \times M$ .

Los elementos de esta diagonal cumplen con la siguiente condición:

$$j == M - (i+1) \quad \forall i \forall j$$

Ejemplo:

$$= \begin{pmatrix} 5 & 1 & 5 & -6 \\ 1 & 2 & 0 & 3 \\ 7 & -3 & 4 & 11 \\ 1 & 9 & 3 & 8 \end{pmatrix}$$

### 3. CRUZ DE LA MATRIZ

Para una matriz de orden  $M \times N$ , la cruz de la matriz está formada por las filas y columnas centrales.

Según sea la cantidad de filas o columnas par o impar se tendrá una o dos filas o columnas pertenecientes a la cruz de la matriz.

Para M impar – N impar

<b>i == M/2</b>		X	
	X	X	X
		X	

**j == N/2**

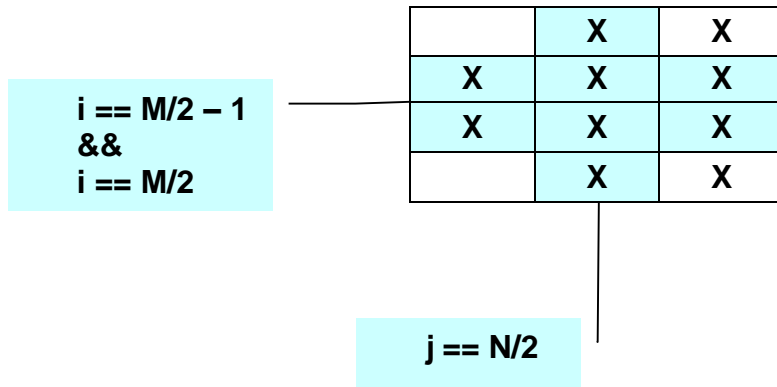
Para M par – N par

<b>i == M/2 - 1 &amp;&amp; i == M/2</b>		X	X	
	X	X	X	X
	X	X	X	X
		X	X	

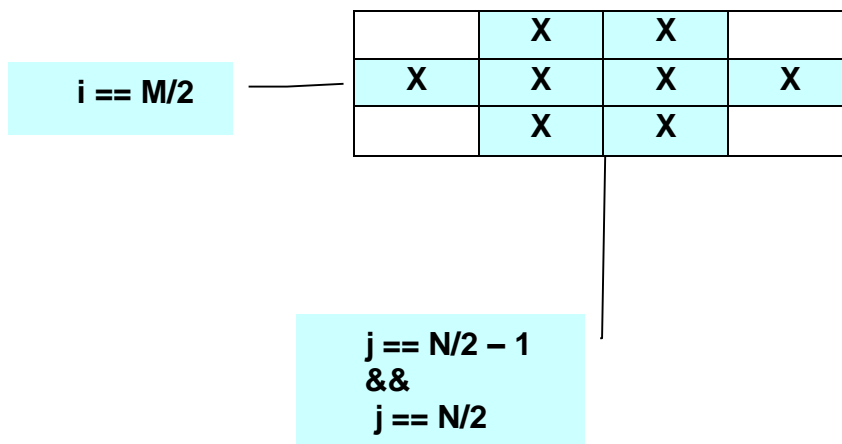
**j == N/2 - 1  
&&  
j == N/2**



Para M par – N impar



Para M impar – N par



#### 4. CUADRO DE LA MATRIZ

Para una matriz de orden MxN, la cruz de la matriz está formada por la primera y última fila, primera y última columna.

