

# SEMANA 13

## MATRICES ARREGLOS BIDIMENSIONALES

# INTRODUCCIÓN

Supóngase que se tiene la siguientes calificaciones de los alumnos de Pseudocodigo de la sec. 107 *arreglados* de la siguiente manera:

| Alumno | Parcial 1 | Parcial 2 | Parcial 3 |
|--------|-----------|-----------|-----------|
| Luis   | S         | B         | E         |
| Carmen | NA        | S         | E         |
| Miguel | E         | E         | E         |

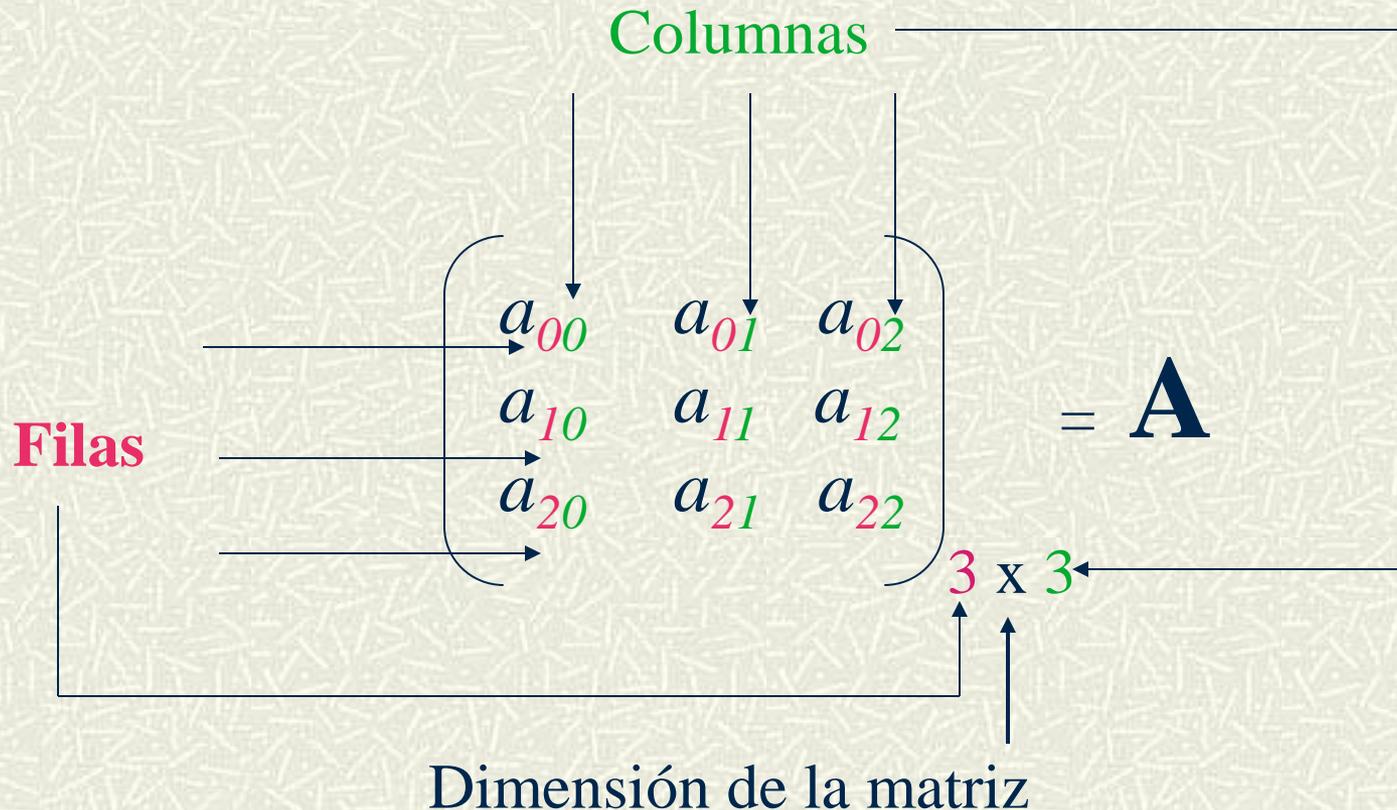
# DEFICIÓN

**MATRIZ** o arreglo bidimensional es un arreglo de arreglos, que representan una tabla, tiene filas y columnas. Los datos de una matriz son todos del mismo tipo y son accedidos mediante dos **índices** o posiciones: uno para filas y otro para columnas.

Por ejemplo:

|   | 1        | 2       | 3       |
|---|----------|---------|---------|
| 1 | S (1,1)  | B (1,2) | E (1,3) |
| 2 | NA (2,1) | S (2,2) | E (2,3) |
| 3 | E (3,1)  | E (3,2) | E (3,3) |

# Una Matriz desde el punto de vista matemático



# Vista conceptual de una matriz

|        | Columna 0 | Columna1 | Columna2 | Columna3 |
|--------|-----------|----------|----------|----------|
| Fila 0 |           |          |          |          |
| Fila 1 |           |          |          |          |
| Fila 2 |           |          |          |          |

En este caso tenemos una matriz de dimensión ( o tamaño)  $M \times N$ , donde  $M$  es el numero de filas y  $N$  el número de columnas. Aquí  $M=2$  y  $N=4$

De la misma forma que los arreglos, una matriz debe tener también un nombre, digamos  $M$ .

# Acceso a los elementos de una Matriz

- # Acceder a los elementos de una matriz significa, ser capaces de almacenar valores y recuperarlos de cada elemento de la matriz.
- # Cada elemento de la matriz tiene asignado una posición denotada por su fila y su columna. Por ejemplo:

$M[0][0]$

Se refiere al primer elemento de la Matriz M que está localizado en la fila 0 y columna 0

$M[1][2]$

¿ A cuál elemento se hace referencia?

# Acceso a los elementos de una Matriz. Continuación

Sea la matriz  $M[3][4]$

|        | Columna 0 | Columna1 | Columna2 | Columna3 |
|--------|-----------|----------|----------|----------|
| Fila 0 |           |          |          |          |
| Fila 1 |           |          |          |          |
| Fila 2 |           |          |          |          |

$M[0][0]$  points to the top-left cell.  $M[2][3]$  points to the cell at Row 2, Column 3.

- # La fila y la columna inician desde cero.
- # Tener cuidado de no exceder los límites de la matriz.
- # Cualquier elemento individual de una matriz, puede ser utilizado como una variable normal.

# Dimensionar y leer una matriz

- # Para dimensionar o declarar una matriz hacemos  $M [5][6]$ .
- # La lectura de una matriz se realiza por medio de dos ciclos anidados, uno que recorra la filas y otro las columnas, es decir,

```
para fila = 0 hasta 4 hacer
  para columna = 0 hasta 4 hacer
    Leer M[fila][columna]
  fin_para_columna
fin_para_fila
```

# Dimensionar y leer una matriz

## Cont.

El recorrido de la matriz  $M$  para su lectura se hace de la siguiente manera:

|   | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | → |   |   |   |   |   |
| 1 | ← |   |   |   |   |   |
| 2 | → |   |   |   |   |   |
| 3 | ← |   |   |   |   |   |
| 4 | → |   |   |   |   |   |

Si una matriz tiene igual número de filas y columnas decimos que es una matriz **cuadrada**.

# Procesar una matriz

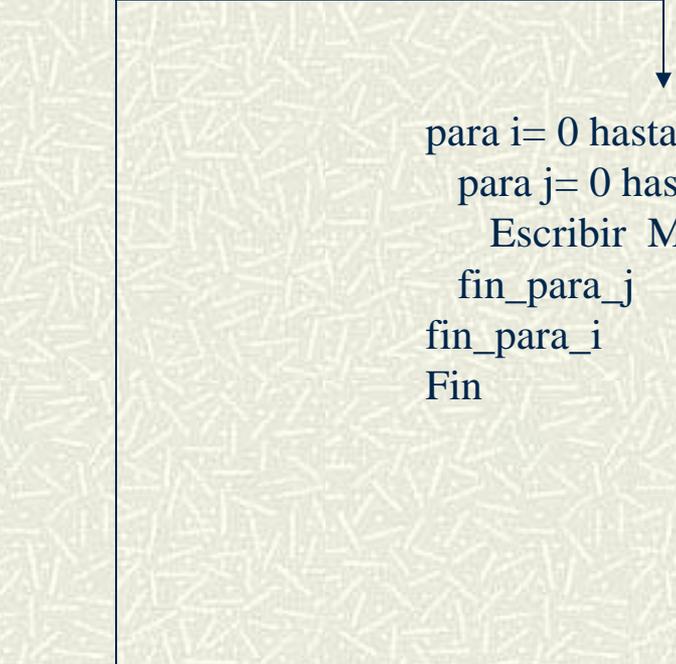
**Ejemplo:** Calcular el promedio de los elementos de una matriz.

Inicio

```
var suma, promedio,i,j  
dimensionar M[3][3]
```

```
para i= 0 hasta 2 hacer  
  para j= 0 hasta 2 hacer  
    Leer M[i][j]  
  fin_para_j  
fin_para_i
```

```
para i= 0 hasta 2 hacer  
  para j= 0 hasta 2 hacer  
    suma=suma+M[i][j]  
  fin_para_j  
fin_para_i  
promedio = suma/9
```

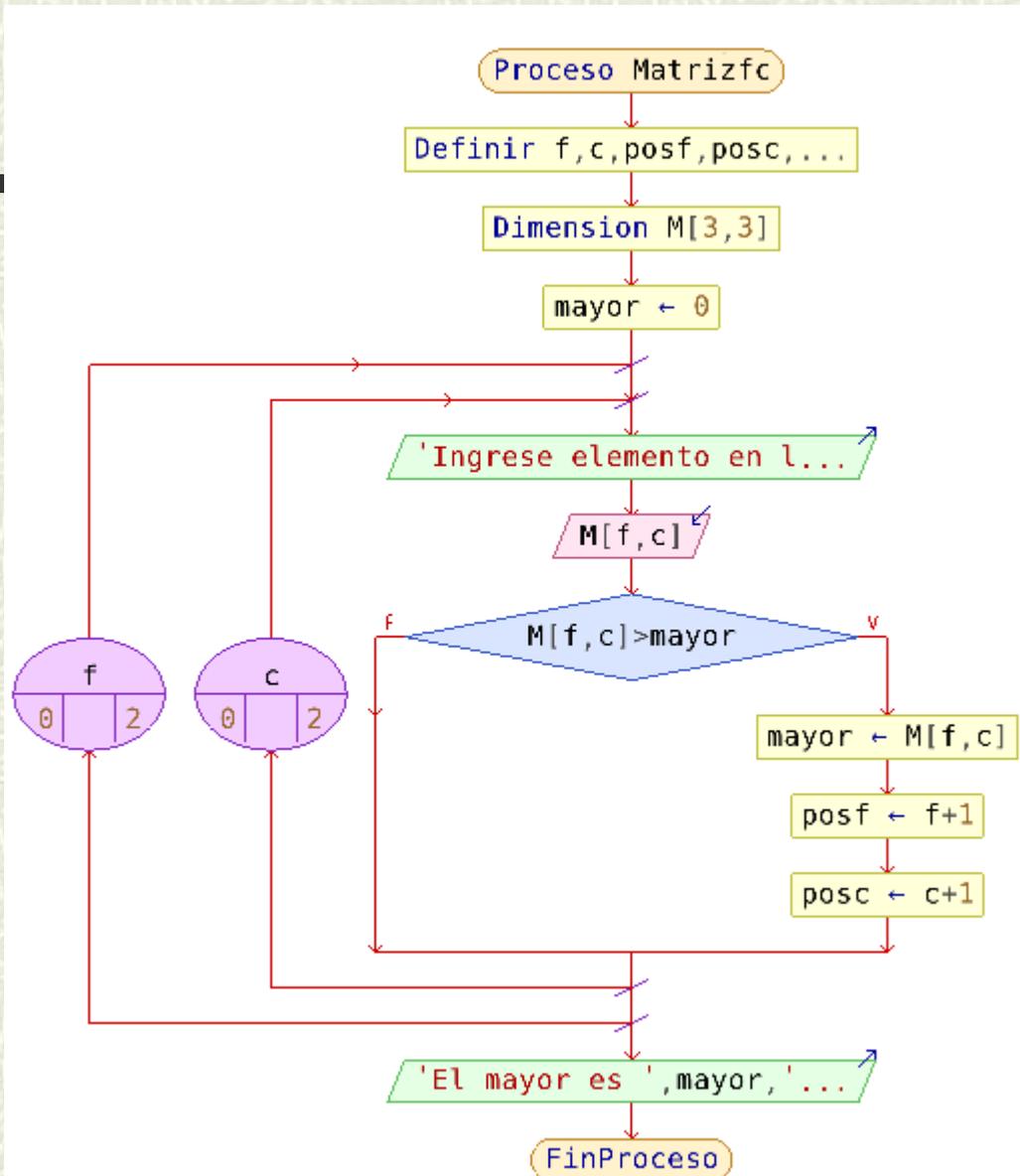


```
para i= 0 hasta 2 hacer  
  para j= 0 hasta 2 hacer  
    Escribir M[i][j]  
  fin_para_j  
fin_para_i  
Fin
```

**Ejemplo:** Algoritmo para leer una matriz de 10 x 10 y determinar la posición [f][c] del número mayor almacenado en la matriz. Se supone que todos los números son diferentes.

```
1  Proceso Matrizfc
2  Definir f,c,posf,posc,M,mayor Como Entero;
3  Dimension M[3,3];
4  mayor<-0;
5  Para f<-0 hasta 2 Hacer
6      Para c<-0 hasta 2 Hacer
7          Escribir "Ingrese elemento en la fila ",f+1," columna ",c+1;
8          Leer M[f,c];
9          Si M[f,c]>mayor Entonces
10             mayor<-M[f,c];
11             posf<-f+1;
12             posc<-c+1;
13             ;
14             FinSi
15         FinPara
16     FinPara
17     Escribir "El mayor es ", mayor," fila ",posf," columna ",posc;
18
19 FinProceso
```

# Diagrama de flujo de datos

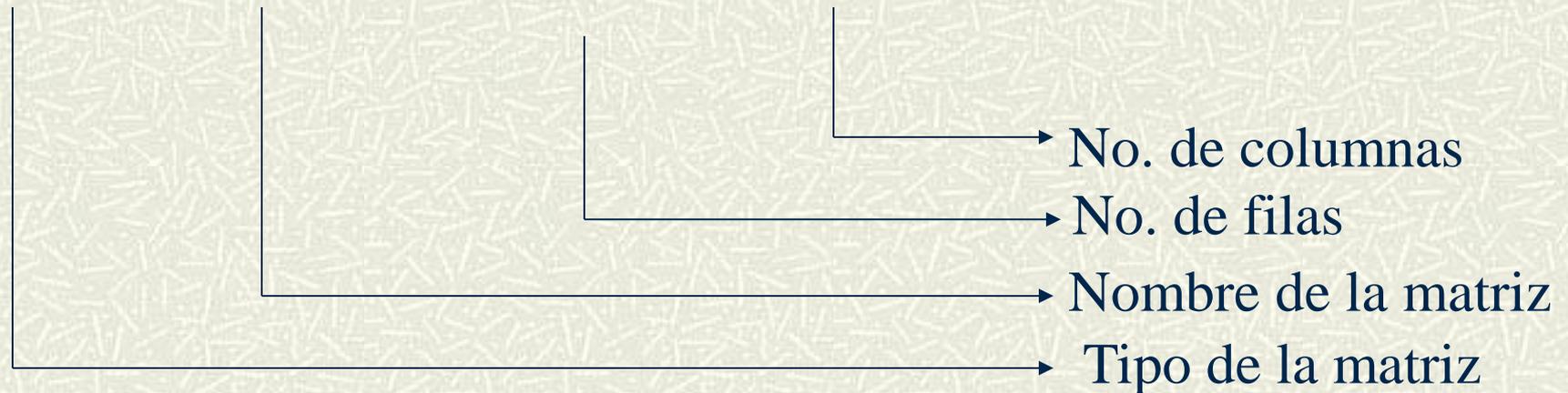


# Declaración de una matriz en C

**tipo NombreMatriz [ No. filas ][No. columnas];**

**Ejemplo:**

```
int Tabla [ 3 ] [ 5 ] ;
```



# Inicializar una matriz

Además de leer, escribir y procesar matrices, también podemos inicializarlas con valores para evitar leerlos desde el teclado u otro dispositivo de E/S.

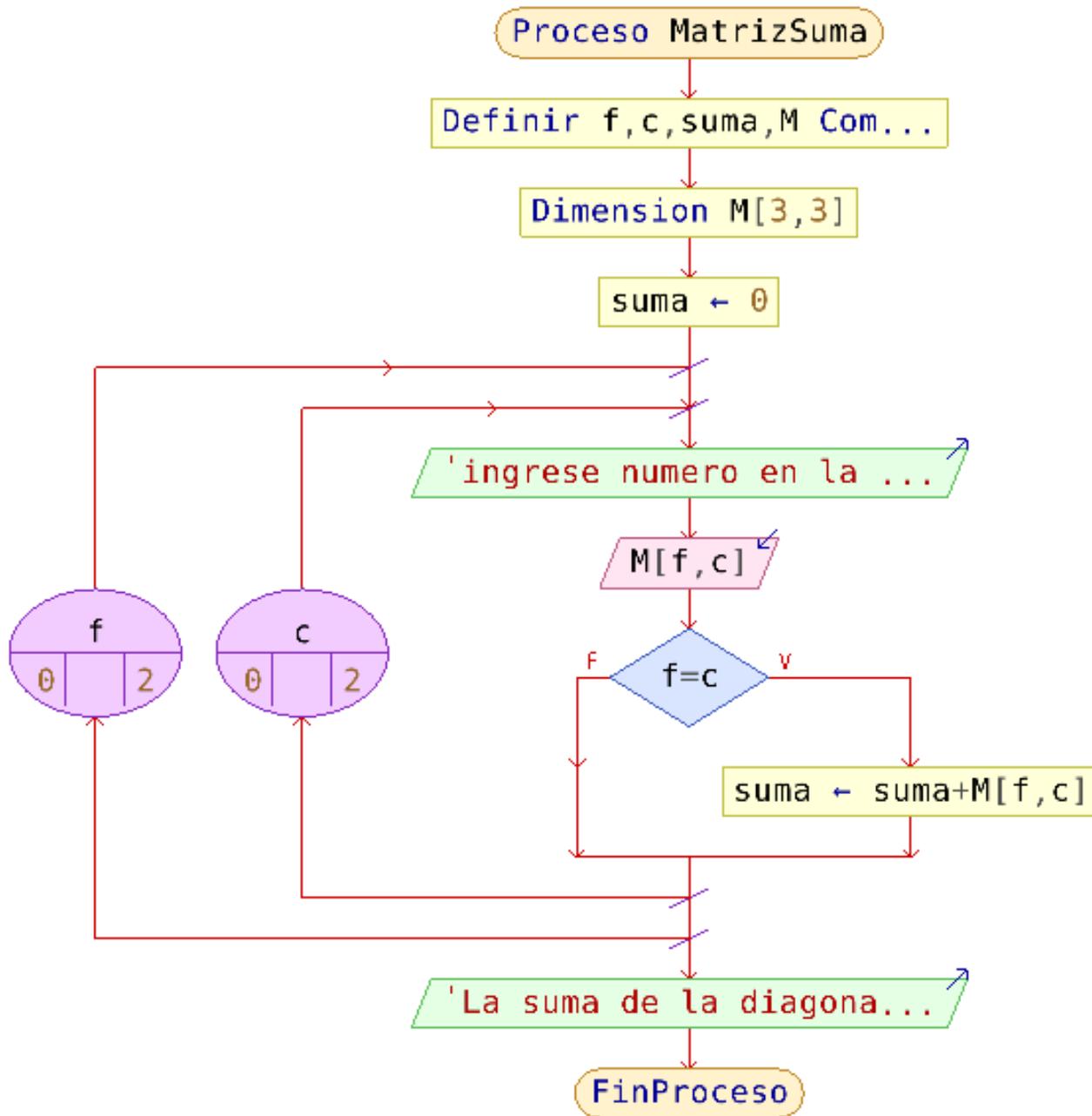
La inicialización se hace como sigue:

```
int tabla[2][3]={{ 10,20,30},{ 40,50,60}};
```

|        | Columna0 | columna1 | columna2 |
|--------|----------|----------|----------|
| Fila 0 | 10       | 20       | 30       |
| Fila 1 | 40       | 50       | 60       |

# Otro ejemplo de matrices en C

```
1  Proceso MarizSuma
2  Definir f,c,suma,M como entero;
3  Dimension M[3,3];
4  suma<-0;
5  para f<- 0 hasta 2 hacer
6      para c<- 0 hasta 2 hacer
7          Escribir "ingrese numero en la fila ",f+1," columna ",c+1;
8          Leer M[f,c];
9          si f=c entonces
10             suma<-suma+M[f,c];
11             finsi
12         finpara
13     finpara
14     Escribir "La suma de la diagonal es",suma;
15 FinProceso
```



# Codificación en C++ del ejemplo anterior

```
#include<stdio.h>
#include<conio.h>
#define FIL 10
#define COL 10

Main() {
    int f,c, M[FIL][COL];
    int suma=0;
    Clrscr();
    Printf("Dame los valores");
    For(f=0;f<FIL;f++){
        For(c=0;c<COL;c++){
            scanf("%d",&M[f][c]);
        }
    }
}
```

```
For (f=0;f<FIL;f++) { /*Se realiza la suma*/
    For (c=0;c<COL;c++) {
        suma=suma+M[f][f];
    }
}
For (f=0;f<FIL;f++) { /*Se escribe M*/
    For (c=0;c<COL;c++) {
        printf("%d ",M[f][c]);
    }
    printf("\n");
}
Printf("La suma de la
        diagonal es:%d",suma);
}
```

# Ejercicios

1.- Considere la siguiente declaración:

definir mensajes como cadena;

dimension mensajes[5]<-"Excelente","Bien","Ok","A casa"}; ¿Es correcto? Si, No  
¿Porqué? ¿Cuál es la salida de la siguiente sentencia?

Escribir mensajes[0];

2.- ¿Cuál es la salida del siguiente segmento de programa?

```
Proceso sin_titulo
  definir junk como cadena;
  Definir i,k como entero;
  dimension junk[4,4];
  para i<-0 hasta 3 hacer
    para k<-0 hasta 3 hacer
      si i mod 2 =0 entonces
        junk[i,k]<-"A";
      Sino
        junk[i,k]<-"B";
      FinSi
    FinPara
  FinPara
  escribir junk[i,k];
FinProceso
```

# Ejercicios. Continuación

---

- # 3. Escriba un algoritmo que lea una matriz de flotantes de dimensión  $3 \times 5$  y a continuación debería escribir la matriz recién leída.
- # 4. Elabore un algoritmo para leer una matriz de  $7 \times 7$ . Calcular la suma de cada renglón y almacenarla en un vector, la suma de cada columna y almacenarla en otro vector.
- # 5. Algoritmo para leer una matriz de  $20 \times 20$ , sumar las columnas e imprimir qué columna tuvo la máxima suma y la suma de esa columna.
- # 6. Algoritmo que asigne datos a una matriz de  $10 \times 10$  con 1's en la diagonal principal y 0 en las demás posiciones.

- # 7. Algoritmo para leer una matriz de 6x8 y que almacenen toda la matriz en un vector. Imprima el vector resultante.
- # 8. Algoritmo para leer una matriz de 5x6 y que imprima cuántos elementos almacenados en la matriz son 0 , cuántos son positivos y cuántos son negativos.
- # 9. Los resultados de las últimas elecciones a alcalde en el pueblo de Cacalotepec han sido los siguientes:

| Distrito | Condado A | Condado B   | Condado C | Condado D |
|----------|-----------|-------------|-----------|-----------|
| 1        | 194       | 48          | 206       | 45        |
| 2        | 180       | 20          | 320       | 16        |
| 3        | 221       | 90          | 140       | 20        |
| 4        | 432       | 50          | 821       | 14        |
| 5        | 820       | 61 Matrices | 946       | 18        |

# Ejercicio.

## # 9. Continuación

- a) Imprimir la tabla anterior con cabeceras.
- b) Calcule e imprima el número total de votos recibidos por cada candidato y el porcentaje total de votos emitidos. Así mismo visualizar el candidato más votado.
- c) Si algún candidato recibe más del 50% de los votos, el programa imprimirá un mensaje declarándolo ganador.
- d) Si ningún candidato recibe más del 50% de los votos, el programa deberá imprimir el nombre de los dos candidatos más votados, que serán los que pasen a la segunda ronda de las elecciones.

# Ejercicios

- # 10. Supóngase que los cuadrados de un juego del gato, se enumeran como en la figura adjunta y se leen los números de 3 cuadros  $N1, N2, N3$ . Sean  $N1 < N2 < N3$ . Si los tres cuadros así designados están en una línea, asígnele a la variable  $LINEA = 1$ , de otra manera asígnele a  $LINEA = 0$ . ¿Habría una manera más sencilla de enumerar los cuadros a fin de simplificar la prueba?

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

# Mas Ejercicios 11

6. Una cadena de ADN se representa como una secuencia circular de bases (adenina, timina, citosina y guanina) que es única para cada ser vivo, por ejemplo:

|   |   |   |
|---|---|---|
| A | T | G |
| T |   | C |
| A | T | G |

Dicha cadena se puede representar como un vector de caracteres recorriéndola en sentido horario desde la parte superior izquierda:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| A | T | G | C | G | T | A | T |
|---|---|---|---|---|---|---|---|

Se pide diseñar una clase que represente una secuencia de ADN e incluya un método booleano que nos devuelva `true` si dos cadenas de ADN coinciden.

**MUY IMPORTANTE:** La secuencia de ADN es cíclica, por lo que puede comenzar en cualquier posición. Por ejemplo, las dos secuencias siguientes coinciden:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| A | T | G | C | G | T | A | T |
| A | T | A | T | G | C | G | T |

# Ejercicios 12

9. Crear una clase `Matriz` para manipular matrices que encapsule un array bidimensional de números reales.
  - a. Incluya en la clase métodos que nos permitan acceder y modificar de forma segura los elementos de la matriz (esto es, las variables de instancia deben ser privadas y los métodos han de comprobar la validez de sus parámetros).
  - b. Escriba un método que nos permita sumar matrices.
  - c. Implemente un método que nos permita multiplicar matrices.
  - d. Cree un método con el que se obtenga la traspuesta de una matriz.

# Ejercicio 13

- # Escribir un programa que pida un número **n** e imprima por pantalla su tabla de sumar. Por ejemplo si el número fuera el 3 la tabla debería ser:

| + | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 1 | 1 | 2 | 3 | 4 |
| 2 | 2 | 3 | 4 | 5 |
| 3 | 3 | 4 | 5 | 6 |

# Ejercicio 14

- # Declare una matriz de número reales de tamaño DIMxDIM (DIM constante con valor 25). Escribir las siguientes funciones:
  - Función Traspuesta: Dada una matriz devolver su traspuesta.
- # void **traspuesta**(float mat[], float matras[]);
  - Función Simetrica: Dada una matriz indica si la matriz es simétrica (1) o si no lo es (0).
- # int **simetrica**(float mat[]);
  - Funcion Suma: Dadas dos matrices, calcular la suma de éstas.
- # void **sumaMat**(float mat1[],float mat2[], res[]);
  - Funcion Resta: Dadas dos matrices, calcular la resta de éstas. Piensa esta respuesta utilizando el procedimiento de suma de matrices, definido anteriormente.
  - Función Multiplica: Dadas dos matrices, devuelve el producto de ellas.
- # void **multiMat**(float mat1[],float mat2[], res[]);

# Ejercicio 15

## 1.2 Ejercicio 2: Producto de vector por matriz.

Para que no deje sin explotar todas las posibilidades de los bucles *for*, le proponemos este sencillo programa que multiplica un vector por una matriz (en este orden). La fórmula utilizada para realizar el producto matricial  $y = x * A$  ha sido la siguiente:

$$b_i = \sum_{j=0}^{n-1} x_j * a_{ji}$$

Escriba el siguiente programa y guárdelo como *VxMatriz.cpp* (no se olvide de crear el proyecto *Ejer2*):

# Ejercicio 16

## 1.3 Ejercicio 3: Producto de matriz por matriz.

Basándose en el producto de vector por matriz anterior, realice un proyecto llamado *Ejer3* que contenga un programa que multiplique dos matrices **A** y **B** de cualquier tamaño; llámelo *matxmat.cpp*.

Recordemos la fórmula para multiplicar dos matrices:

$$C = A * B \Rightarrow c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj} \quad , \text{donde } i = 1 \dots n, j = 1, \dots, n$$

Como se puede observar en la fórmula, intervienen 3 índices *i,j,k*, lo cual se traduce en que necesitaremos tres bucles *for* para realizar dicha operación.

# Ejercicio 17

## 1.4 Ejercicio 4: Resolución de un sistema lineal de 2 ecuaciones.

El ejercicio a realizar en este apartado debe permitir resolver un sistema lineal de 2 ecuaciones del tipo:

$$\begin{aligned}a_{11}X + a_{12}Y &= b_1 \\ a_{21}X + a_{22}Y &= b_2\end{aligned}$$

Donde el *determinante* viene dado por:

$$\det = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

Y los valores de las variables vienen dados por las siguientes expresiones:

$$x = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{\det}; y = \frac{\begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix}}{\det}$$

Crear el proyecto *Ejer4* y el fichero *Ecua2.cpp* para escribir el código.

# Cont.

Pasos a seguir para programar este ejercicio:

- a. Crear una matriz para albergar los coeficientes  $a_{11}$ ,  $a_{12}$ , etc.
- b. Crear un vector para albergar los términos independientes  $b_1$  y  $b_2$ .
- c. Pedir los datos al usuario.
- d. Realizar los cálculos (verificar que el determinante no es cero).
- e. Imprimir la matriz del determinante (con formato), así como su valor.
- f. Imprimir la solución ( $x$  e  $y$ ).