

## Capítulo 8: Arreglo bidimensional o Matriz

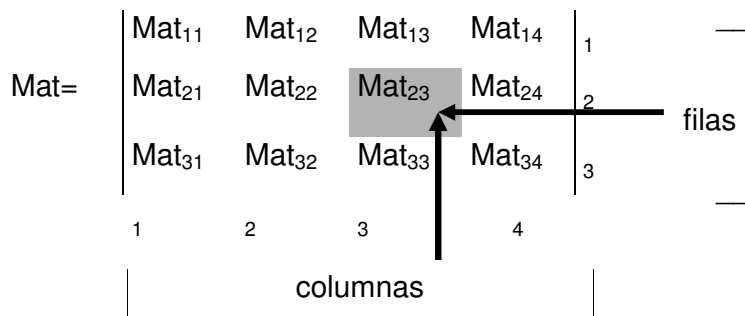
Para acceder a un determinado elemento de un arreglo unidimensional o vector se utiliza un índice.



El tipo arreglo permite utilizar más de un índice para determinar la posición de un elemento, si son dos se denomina arreglo bidimensional o matriz.

Cada elemento de la matriz se determina con la posición de fila y columna.

El elemento sombreado es **Mat[2, 3]**, segunda fila y tercera columna.



El tipo se describe estableciendo la cantidad de filas y de columnas (1er y 2do índice respectivamente)

Type

TipoMatriz = array [1..3, 1..4] of integer;

Identificador de tipo      tipos de índices      tipo base

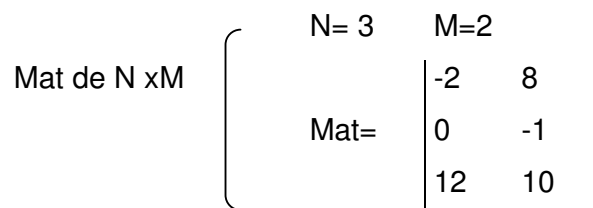
Siguiendo las mismas reglas que los índices de vectores (ambos son arreglos)

Var

Mat : TipoMatriz;

El espacio de memoria para Mat queda reservado, independientemente de que el programa requiera parte o la totalidad de las componentes de la variable Mat.

Si no se utiliza la totalidad del almacenamiento se utilizan dos variables para almacenar cantidad de columnas y de filas.



Procedure **LeeMatriz**(Var Mat: TipoMatriz; Var N, M :byte);

Var

i, j: byte;

Begin

Write ('Ingrese cantidad de filas'); Readln(N);

Write ('Ingrese cantidad de columnas'); Readln(M);

For i:= 1 to N do

For J:= 1 to M do

Begin

Write('fila, columna ', i:3, j:3); Readln(Mat[i, j]);

end

end;

Indices [i , j]	valor
1 , 1	-2
1 , 2	8
2 , 1	0
2 , 2	-1
3 , 1	12
3 , 2	10

los elementos deben ingresar en el orden en que se generan los índices (1ra, 2da y 3ra fila)

Si se ingresan por columnas (1ra y 2da columna) se cambia el orden de los ciclos.

Procedure **LeeMatrizxCol**(Var Mat: TipoMatriz; Var N, M :byte);

Var

i, j: byte;

Begin

Write ('Ingrese cantidad de filas'); Readln(N);

Write ('Ingrese cantidad de columnas'); Readln(M);

For j:= 1 to M do

For i:= 1 to N do

Begin

Write('fila, columna ', I:3, j:3); Readln(Mat[i, j]);

end

end;

Indices [i , j]	valor
1 , 1	-2
2 , 1	0
3 , 1	12
1 , 2	8
2 , 2	-1
3 , 2	10

Procedure **EscribeMatriz** (Mat: TipoMatriz; N,M : Byte); *{en forma matricial}*

Var

i, j: byte;

Begin

For i:= 1 to N do

Begin

For j:= 1 to M do

Write(Mat[i, j]: 4);

Writeln;

end

end;

*Escribe la fila i*

*Al terminar la fila i, salta a la línea siguiente*

Otra posibilidad es la lectura parcial o total de los elementos de la matriz, desordenados.

Para cada uno se ingresa fila, columna y valor. Como no todos ingresan se debe iniciar la matriz en cero.

Procedure **IniciaMatriz** (Var Mat:TipoMatriz; N,M:byte);

Var

i, j: byte;

begin

For j:= 1 to M do

For i:= 1 to N do

Mat[i, j]:=0;

End;

*Nota: una matriz iniciada en cero, puede utilizarse como un conjunto de contadores o acumuladores (índice con significado)*

Procedure **LeeMatrizDesordenados** (Var Mat: TipoMatriz);

Var

i, j: byte;

begin

write('Ingrese fila,0= fin de ingreso'); Readln(I);

While i <> 0 do

Begin

Readln(j);

Readln(Mat[i,j]);

write('Ingrese fila,0= fin de ingreso'); Readln(i);

End;

End;

### Calcular el elemento mínimo de una matriz

Function **Mínimo**(Mat: TipoMatriz; N,M: byte): integer;

Var

i,j: byte; Min: integer;

Begin

Min:= Mat[1,1];

For i := 1 to N do

For j:= 1 to M do

If Min > Mat[i,j] Then

Min:= Mat[i,j];

Mínimo:= Min;

End;

-2	8
0	-1
12	10



Resolver el ejercicio 1 y 2 propuestos al final de Capitulo

### Calcular el mínimo elemento de una fila

Function MinFila(Mat:TipoMatriz; i,M:Byte):integer; {recorre la fila i con j entre 1 y M}

Var

j:Byte;

Min: integer;

Begin

Min:= Mat[i,1];

For j:= 2 to M do

If Min > Mat[i,j];

Min:=Mat[i,j];

MinFila:= Min

End;

Utilizando la función generar un arreglo VMin de N elementos, que contenga el mínimo de cada fila.

Procedure **VectorMinimos** (Mat: TipoMatriz; N,M: byte; Var Vmin: TV);

Var

i:byte;

begin

For i := 1 to N do

VMin[i] := MinFila(Mat, i, M);

End;

-2	8	VMin=	-2
0	-1		-1
12	10		10

A partir de la matriz y de un vector de M elementos reales, calcular cuantas filas coinciden con el vector.

Function **CuantosCoinciden** (Mat: TipoMatriz; N,M: byte; Vec: TV);

Var

Cont, i, j :byte;

begin

Cont:=0;

For i := 1 to N do

Begin

j:= 1 ;

while (j<M) and (Vec[j] = Mat[ i , j]) do

j:=j + 1;

If Vec[j] = Mat[ i , j] then

Cont:= Cont + 1;

End,

CuantosCoinciden:= Cont;

End;

Vec=	12	10
	1	2

Mat=	-2	8
	0	-1
	12	10

una fila coincide



Resolver los ejercicios 3, 4 y 5 propuestos al final de Capitulo

**Matrices cuadradas de NxN**, coincide la cantidad de filas y de columnas. Para este tipo de matrices se define:

- ✓ **diagonal principal** formada por los elementos  $A[i, i]$  con  $i = 1..N$
- ✓ **diagonal secundaria** formada por los elementos  $A[i, N - i + 1]$  con  $i = 1..N$
- ✓ **triangular inferior** los elementos  $A[i, j] = 0$ , con  $i = 2..N$  y  $j < i$

**diagonal principal**

a<sub>11</sub> a<sub>12</sub> a<sub>13</sub> a<sub>14</sub>  
a<sub>21</sub> **a<sub>22</sub>** a<sub>23</sub> a<sub>24</sub>  
a<sub>31</sub> a<sub>32</sub> **a<sub>33</sub>** a<sub>34</sub>  
a<sub>41</sub> a<sub>42</sub> a<sub>43</sub> **a<sub>44</sub>**

**diagonal secundaria**

a<sub>11</sub> a<sub>12</sub> a<sub>13</sub> **a<sub>14</sub>**  
a<sub>21</sub> a<sub>22</sub> **a<sub>23</sub>** a<sub>24</sub>  
a<sub>31</sub> **a<sub>32</sub>** a<sub>33</sub> a<sub>34</sub>  
**a<sub>41</sub>** a<sub>42</sub> a<sub>43</sub> a<sub>44</sub>

**triangular inferior**

a<sub>11</sub> a<sub>12</sub> a<sub>13</sub> a<sub>14</sub>  
**0** a<sub>22</sub> a<sub>23</sub> a<sub>24</sub>  
**0** **0** a<sub>33</sub> a<sub>34</sub>  
**0** **0** **0** a<sub>44</sub>

Type

TM= array[1..5,1..5] of real;

Calcular la suma de la diagonal principal

Function SumaDiagonal (A: TM; N: byte):real;

Var

i :byte;

Sum:real;

Begin

Sum:= 0;

For i := 1 to N do

Sum:=Sum + A[i, i];

SumaDiagonal:= Sum;

End;

Contar la cantidad de elementos negativos debajo de la diagonal

Function CuentaNegativos (A: TM; N: byte):real;

Var

Cont, i :byte;

Begin

Cont:= 0;

For i := 2 to N do

For j := 1 to i - 1 do

If A[i, j] &lt; 0 then

Cont:=Cont + 1;

CuentaNegativos:= Cont;

End;

*Genera solo los índices que están bajo la diagonal*

Resolver los ejercicios 6 y 7 propuestos al final de Capitulo

**Ejercitación***Desarrollar programas Pascal que resuelva los problemas propuestos utilizando funciones y procedimientos. Proponer juegos de datos y verificar su funcionamiento.*

1.- Desarrollar un programa que implemente procedimientos y/o funciones que permitan:

- a) Leer una matriz entera A de NxM
- b) calcular el máximo de los elementos de la columna C ( $1 \leq C \leq M$ ).
- c) calcular el promedio de los elementos de la fila F ( $1 \leq F \leq N$ ).
- d) Generar un vector de M elementos con los promedios de cada columna de la matriz A
- e) Generar un vector V con los elementos de A que sean positivos.

C y F se ingresan por teclado. Mostrar los resultados obtenidos en cada inciso.

2.- En un archivo de texto se ha almacenado una matriz real de PxQ. En la primera línea los valores P y Q, en las P restantes los valores de cada fila (Q reales en cada una). Se pide:

- a) Leer la dimensión y la matriz.
- b) Ingresar desde teclado valores enteros C1 y C2, verificar que ambos pertenezcan al intervalo [1, Q] y permutar las columnas C1 con C2.

- c) Ingresar desde teclado valores enteros F1, F2 y F3, verificar que sean diferentes entre sí y que pertenezcan al intervalo [1, P], luego reemplazar los elementos de la fila F3 por la suma de los elementos correspondientes de las filas F1 y F2.

3.- Calcular el promedio de los elementos de una matriz A de NxN (cuadrada) que cumplan  $A[i, j] < A[j, i]$  (elementos simétricos diferentes)

4.- Se han registrado los mm. de lluvia diaria (puede ser cero) caída en el mes de junio durante N años. Dicha información debe leerse y almacenar en una matriz de Nx30, se pide calcular e informar:

- Cuántos años registraron lluvias todos los días de junio.
- Cuál fue el año (orden 1..N) con menos días de lluvia
- Cual fue el día (1..31) con mayor promedio diario (cantidad de mm.).

5.- En un juego se utiliza un tablero de N x N casillas. En cada una de ellas hay fichas Verdes, Azules o Rojas (en distintas cantidades por casilla y de un solo color). Los datos se almacenan en dos matrices, una contiene la distribución de los colores y otra las cantidades. Se desea saber si el color que tiene más fichas es el más extendido en todo el tablero.

Ejemplo N = 4

R	A	A	V
V	A	A	A
R	R	R	V
A	V	V	A

7	1	8	3
0	7	4	1
9	6	7	5
3	3	1	2

La respuesta es NO porque el Rojo tiene más fichas (4 casillas y 29 fichas) pero el más extendido es el Azul (7 casillas y 26 fichas).

6.-Dada una matriz A de NxM elementos enteros, se desea generar un arreglo lineal B con los elementos de la matriz A que cumplan:

$$\begin{array}{ll} A[i, j] \text{ es par} & \text{y} \quad A[i, j] > A[i, 1] \\ A[i, j] \text{ es impar} & \text{y} \quad A[i, j] < A[i, 1] \end{array} \quad \text{ó}$$

Ejemplo: N=3 y M=4

A =

1	8	-6	-7
4	5	2	6
0	5	2	13

→ B =

8	-7	6	2
---	----	---	---

7.-Leer una matriz A de NxN números enteros. Armar un vector V con los elementos de la matriz que pertenecen a filas de orden par y cumplen con la condición de ser mayores que el promedio de los elementos de la columna a la que pertenecen. Mostrar V.

Ejemplo:

Si  $A = \begin{bmatrix} 2 & 3 & -2 & 5 \\ 1 & 8 & 3 & 5 \\ 0 & -9 & 4 & 2 \\ 6 & 2 & 12 & 0 \end{bmatrix}$  entonces  $V = [8 \ 5 \ 6 \ 2 \ 12]$