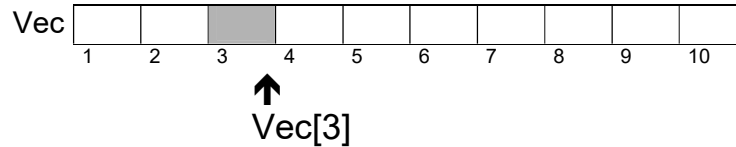


Capítulo 6

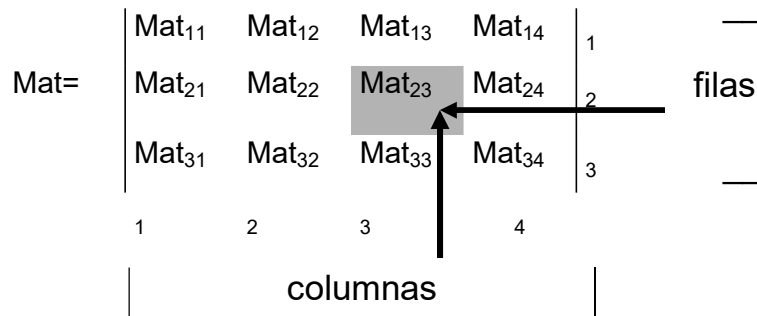
1. Arreglos bidimensionales o Matrices. Declaración del tipo.
2. Lectura, escritura operaciones sobre matrices. Ejemplos
3. Matrices cuadradas. Diagonal principal y secundaria. Matriz triangular, simétrica.

1. Arreglos bidimensionales o Matrices

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. Cada elemento de la matriz se accede con dos índices : fila y columna. El elemento sombreado es **Mat[2, 3]**, segunda fila y tercera columna.

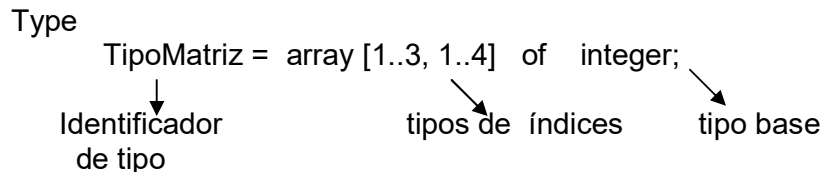


Se generaliza un arreglo n-dimensional para n índices. Aunque no es común utilizar más de tres índices.

1.1. Declaración del tipo

Cuando se describe el tipo se establece:

- a. El identificador o nombre del tipo
- b. Para cada índice, la cantidad máxima de elementos y el rango sus posiciones.
- c. El tipo de los valores que almacena



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.

2. Lectura, escritura operaciones sobre matrices. Ejemplos

2.1. Lectura por filas, por columnas, desordenada

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

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)

i	j	Mat[i,j]
3	1	-1
1	2	8
2	1	7

Mat de N xM

$$\begin{matrix} N=3 & M=2 \\ \text{Mat}= \begin{vmatrix} 0 & 8 \\ 7 & 0 \\ -1 & 0 \end{vmatrix} \end{matrix}$$

Procedure

LeeMatrizDesordenados (Var

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;

Mat: TipoMatriz);

2.2. Escritura en formato matricial

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

2.3. Calcular el elemento mínimo de una matriz

Function **Minimo**(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];

Minimo:= Min;

End;

-2	8
0	-1
12	10

2.4. 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] then
      Min:=Mat[i,j];
  MinFila:= Min
End;

```

2.4.1. 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;

```

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

2.5. 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

3. 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

```
a11 a12 a13 a14
a21 a22 a23 a24
a31 a32 a33 a34
a41 a42 a43 a44
```

diagonal secundaria

```
a11 a12 a13 a14
a21 a22 a23 a24
a31 a32 a33 a34
a41 a42 a43 a44
```

triangular inferior

```
a11 a12 a13 a14
0 a22 a23 a24
0 0 a33 a34
0 0 0 a44
```

Type

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

3.1.Calcular la suma de la diagonal principal (traza)

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;

3.2.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] < 0 then

Cont:=Cont + 1;

CuentaNegativos:= Cont;

End;

Genera solo los índices que están bajo la diagonal