


```

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;
    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;
    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;
    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 **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

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

1. Ingresar una matriz de N x M de enteros resolver mediante procedimientos o funciones los siguientes puntos y mostrar el resultado:

- Dado un número, calcular cuantas veces se repite (puede no estar).
- Indicar cuántos elementos son nulos, positivos y negativos.
- Intercambiar la fila K con la fila H (K y H menor o igual que N)

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

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

2. Ingresar una matriz de MxN de enteros, escribir:

- Los elementos de la fila cuya suma de componentes sea la mayor.
- Los elementos de la columna cuya suma de componentes sea la menor.
- Los elementos del "contorno".

-12	33	4	5
10	0	45	7
62	81	71	8
-7	69	10	9

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

A[fila, columna] <= 0 para fila impar y columna impar ó
 A[fila, columna] > 0 para fila par y columna par

Ejemplo: N=3 y M=4

$$A = \begin{bmatrix} 1 & 8 & -6 & -7 \\ 4 & 6 & 3 & 2 \\ 0 & -5 & 9 & 13 \end{bmatrix} \rightarrow B = \begin{bmatrix} -6 & 6 & 2 & 0 \end{bmatrix}$$

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_{11} a_{12} a_{13} a_{14}
 a_{21} **a_{22}** a_{23} a_{24}
 a_{31} a_{32} **a_{33}** a_{34}
 a_{41} a_{42} a_{43} **a_{44}**

diagonal secundaria

a_{11} a_{12} a_{13} **a_{14}**
 a_{21} **a_{22}** **a_{23}** a_{24}
 a_{31} **a_{32}** a_{33} a_{34}
 a_{41} a_{42} a_{43} a_{44}

triangular inferior

a_{11} a_{12} a_{13} a_{14}
0 a_{22} a_{23} a_{24}
0 **0** a_{33} a_{34}
0 **0** **0** a_{44}

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] < 0 then

Cont:=Cont + 1;

CuentaNegativos:= Cont;

End;

Genera solo los índices que están bajo la diagonal

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

4. Se han registrado las ventas mensuales de cada uno de N empleados en el último semestre. Cada mes se ha establecido un objetivo de importe de ventas a cumplir. Se pide leer la información en una matriz de $6 \times N$ real y un vector de 6 elementos reales, luego calcular e informar:

- Cuantos fueron los meses en los cuales todos los empleados superaron el objetivo
- Dado un número de empleado (1..N) cuantos meses no cumplió el objetivo

5. Una matriz cuadrada A se dice es simétrica si $A[i, j] = A[j, i]$ para todo i, j dentro de los límites de matriz. Escribir una función que determine:

- Cuantos elementos simétricos son diferentes.
- Verificar si es o no simétrica.

6. Dada una matriz $M \times N$ de enteros en el rango de 1 a 50, verificar que ningún número esté repetido, en caso contrario indicar cuántas veces se repite cada uno.

7. Con el fin de lograr un control sobre el movimiento de personas en un edificio de oficinas, se registra cada día del mes de marzo, cuántas personas se dirigen a cada uno de los 25 pisos del edificio.

Los datos están desordenados y consisten en triplas:

Día, Piso Cantidad (pueden repetirse el Día y el Piso, juntos o separados).

Ingresar la información desde un archivo, calcular e informar:

- Para cada piso y cada día, la cantidad de personas que ingresaron.
- Cantidad total de personas ingresadas al edificio durante el mes.
- Promedio diario de personas que ingresaron a cada piso.
- Para un determinado día, el porcentaje de personas que ingresaron dicho día sobre el total del mes.

Pregunta: ¿Cómo convendría construir el programa para fácilmente adaptarlo para cualquier mes del año y edificios de cualquier cantidad de pisos (considerar un máximo de 100 pisos)?

8. Sean T1 y T2 dos matrices de $N \times N$ que representan las fichas (rojas, azules) de un tablero, la primera almacena el carácter 'R' o 'A' y la segunda enteros (cantidad de fichas). Se pide calcular e informar:

- Color predominante (presente en mas casillas)
- Color con más fichas
- Cantidad de columnas de un solo color

9. Un cuadrado mágico de orden N (impar), es una matriz cuadrada de $N \times N$ que contiene números naturales de 1 a N^2 , tal que la suma de cualquier columna, fila o diagonal coincide.

Se construye de la siguiente manera:

- El número 1 se coloca en la casilla central de la primera fila.
- Cada número siguiente se coloca en la casilla correspondiente a la fila anterior y columna posterior
- Si el número es el que sigue a un múltiplo de N, se coloca en la casilla de la fila posterior en la misma columna
- Se considera fila anterior a la primera, la última.
- Se considera columna posterior a la última, la primer.

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9