

Capítulo 10

Matrices

Definición

Una matriz es un conjunto de datos organizados en forma de filas y columnas en donde para referenciar cada dato necesitaremos establecer claramente en qué fila y en qué columna se encuentra. Tomemos la siguiente matriz de datos enteros como ejemplo

Nro. de Fila	↓	(1)	(2)	(3)	(4)	(5)	(6)	← Nro. de Columna
(1)		10	32	-61	24	19	66	
(2)		25	36	2	12	100	1	
(3)		15	20	26	30	35	18	
(4)		10	21	23	26	85	17	

La anterior es, pues, una matriz de números enteros cuyos datos están organizados en 4 filas y 6 columnas. De esta manera para ubicar exactamente un dato solo tendremos que referenciar su posición en cuanto a filas y columnas y sabremos exactamente a qué dato nos estamos refiriendo. Note usted que se necesita tanto de la ubicación exacta de la fila como de la ubicación de la columna para que no exista ninguna duda en cuanto al dato al cual nos estemos refiriendo. Por ejemplo, cuál es el dato que queda en la tercera fila...? Usted seguramente dirá que en la tercera fila están los datos 15, 20, 26, 30, 35 y 18. De manera que no podría preguntarse de manera individual por **el dato** de la tercera fila sino por **los datos** de la tercera fila.

De acuerdo a esto la posición de un dato en una matriz siempre estará determinado por el número de su fila y el número de su columna. Así, por ejemplo, cuál es el dato que se encuentra en la Fila 3 Columna 2..? Pues el número 20.

Nro. de Fila →

	(1)	(2)	(3)	(4)	(5)	(6)	← Nro. de Columna
(1)	10	32	-61	24	19	66	
(2)	25	36	2	12	100	1	
(3)	15	20	26	30	35	18	
(4)	10	21	23	26	85	17	

Cuál es el dato que se encuentra en la Fila 2 Columna 4..? Respuesta: el número 12.

Nro. de Fila →

	(1)	(2)	(3)	(4)	(5)	(6)	← Nro. de Columna
(1)	10	32	-61	24	19	66	
(2)	25	36	2	12	100	1	
(3)	15	20	26	30	35	18	
(4)	10	21	23	26	85	17	

Cuál es el dato que se encuentra en la Fila 1 Columna 6..? Respuesta : el número 66.

Nro. de Fila →

	(1)	(2)	(3)	(4)	(5)	(6)	← Nro. de Columna
(1)	10	32	-61	24	19	66	
(2)	25	36	2	12	100	1	
(3)	15	20	26	30	35	18	
(4)	10	21	23	26	85	17	

Cuál es el dato que se encuentra en la Fila 5 Columna 3..? Respuesta: No existe Fila 5. Para cada una de las preguntas usted puede notar que existe una única respuesta pues se ha dado exactamente la posición en Fila y Columna de cada uno de los datos solicitados.

Si quisiéramos mostrar en pantalla el contenido de la primera casillita de la primera fila de esta matriz y asumiendo que toda la matriz se llama M entonces simplemente diríamos:

Escriba M (1 , 1)

Esto significaría que escriba en pantalla el contenido de la matriz M en la fila 1 columna 1. Siempre que se use esta notación el primer número representará el número de la fila y el segundo representará el número de la columna. Ejecutada esta orden saldrá en pantalla el número 10.

Asimismo si se da la orden

Escriba M (3, 4)

Estaremos mostrando en pantalla el número que se encuentra en la matriz M en la fila 3 columna 4 o sea que nos referimos al número 30. De esta manera podremos referenciar todos y cada uno de los datos almacenados en la matriz.

Si quisiéramos escribir todos los datos almacenados en la segunda fila entonces diríamos

Escriba M (2, 1), M (2, 2), M (2, 3), M (2, 4), M (2, 5), M (2, 6)

Con lo cual saldrían en pantalla los siguientes datos

25 36 2 12 100 1

Note usted que en la orden

Escriba M (2, 1), M (2, 2), M (2, 3), M (2, 4), M (2, 5), M (2, 6)

El valor que referencia la fila se mantiene constante mientras que el valor que referencia la columna varía desde la primera hasta la última (en este caso la primera columna es 1 y la última es 6). De tal manera que si quisiéramos incorporar variables adicionales para simplificar la escritura de los datos de la segunda fila podríamos decir

```
.
.
Para Col = 1 hasta 6
    Escriba M ( 2, Col )
Fin_Para
.
```

O escrito con un ciclo Mientras sería

```

.
.
.
Col = 1
Mientras Col <= 6
    Escriba M ( 2, Col )
    Col = Col + 1
Fin_Mientras
.
.

```

He colocado unos puntos suspensivos para indicar que estos conjuntos de instrucciones son parte de un algoritmo. Note usted que utilizando apropiadamente un ciclo, su variable índice nos va a servir para referenciar progresivamente las columnas de la matriz y permitir que en pantalla aparezcan todos los números de la segunda fila.

Si quisiéramos mostrar en pantalla todos los números de todas las filas tendríamos dos formas: la primera e ineficiente que sería

```

Escriba M ( 1, 1 ), M ( 1, 2 ), M ( 1, 3 ), M ( 1, 4 ), M ( 1, 5 ), M ( 1, 6 )
Escriba M ( 2, 1 ), M ( 2, 2 ), M ( 2, 3 ), M ( 2, 4 ), M ( 2, 5 ), M ( 2, 6 )
Escriba M ( 3, 1 ), M ( 3, 2 ), M ( 3, 3 ), M ( 3, 4 ), M ( 3, 5 ), M ( 3, 6 )
Escriba M ( 4, 1 ), M ( 4, 2 ), M ( 4, 3 ), M ( 4, 4 ), M ( 4, 5 ), M ( 4, 6 )

```

Puede usted notar que al mostrar los datos por fila mientras se mantiene constante el valor de referencia de una fila se avanza en las columnas desde 1 hasta 6. Por lo tanto, en una segunda versión ineficiente de este conjunto de instrucciones podríamos escribir lo mismo de la siguiente forma

```

.
.
.
Para Col = 1 hasta 6
    Escriba M ( 1, Col )
Fin_Para

Para Col = 1 hasta 6
    Escriba M ( 2, Col )
Fin_Para

Para Col = 1 hasta 6
    Escriba M ( 3, Col )
Fin_Para

Para Col = 1 hasta 6
    Escriba M ( 4, Col )
Fin_Para
.
.

```

O si quisiéramos escribir los datos, columna a columna, entonces se podría utilizar

```

.
.
Para Col = 1 hasta 6
    Escriba M ( 1, Col ), M ( 2, Col), M ( 3, Col ), M ( 4, Col )
Fin_Para
.
.

```

Ya puede usted notar que en cualquiera de los dos casos se repiten instrucciones o conjuntos de instrucciones lo que hace suponer que con dos ciclos anidados (o sea adicionando un ciclo interno a la estructura planteada) podremos recorrer todos los datos de la matriz mostrándolos en pantalla.

De acuerdo a esto podremos recorrer la matriz en mención utilizando los dos ciclos siguientes

```

.
.
Para Fil = 1 hasta 4
    Para Col = 1 hasta 6
        Escriba M ( Fil , Col )
    Fin_Para
Fin_Para
.
.

```

o también se hubiera podido plantear de la siguiente forma (utilizando ciclos Mientras anidados):

```

.
.
Fil = 1
Mientras Fil <= 4
    Col = 1
    Mientras Col <= 6
        Escriba M ( Fil , Col )
        Col = Col + 1
    Fin_Mientras
    Fil = Fil + 1
Fin_Mientras
.
.

```

Deberá el lector desarrollar una prueba de escritorio de cada uno de estos conjuntos de instrucciones con el ánimo de que pueda conceptualizar de una manera mas clara el funcionamiento estos ciclos anidados en unión con la idea de una matriz.

En estos ejemplos hemos asumido que tenemos una matriz de 4 filas y 6 columnas con datos. Qué sucede si en vez de querer mostrar los datos en pantalla quisiéramos leer datos y almacenarlos en la matriz...? Entonces lo que tendríamos que hacer es cambiar la orden

Escriba M (Fil , Col)

Por

Lea M (Fil , Col)

De esta manera cuando, por ejemplo, la variable *Fil* almacene el valor 2 y la variable *Col* almacene el valor 3, la orden *Escriba M (Fil , Col)* se entenderá como *Lea un entero y guárdelo en la matriz M en la Fila Fil (que vale 2) Columna Col (que vale 3)* o sea en la posición (2 , 3) sabiendo que el primer entero siempre representa la fila y el segundo entero siempre representa la columna.

De esta forma si utilizamos el siguiente conjunto de instrucciones

```
.
.
Fil = 1
Mientras Fil <= 4
    Col = 1
    Mientras Col <= 6
        Lea M ( Fil , Col )
        Col = Col + 1
    Fin_Mientras
    Fil = Fil + 1
Fin_Mientras
.
.
```

Estaremos leyendo 24 enteros y cada uno de ellos irá quedando almacenado en una “casillita” independiente dentro de la matriz en el orden de las filas, es decir, primero llena todas las “casillas” de una fila, luego continua con la siguiente fila hasta llegar a la cuarta fila que es la última (considerando que seguimos hablando de la misma matriz 4x6 con la que hemos construido todos los ejemplos hasta el momento).

Características de una Matriz

Siempre que vayamos a utilizar una matriz es importante que tengamos en cuenta que los datos en ella están organizados en forma de filas y columnas. Esto es importante porque ello precisamente es lo que justifica la utilización del concepto de Matriz dentro de un algoritmo. Una matriz tendrá las siguientes características:

Nombre

Toda matriz por ser **un** conjunto de variables deberá tener un identificador que permita referirse a ella en cualquier momento dentro del algoritmo. Dicho nombre se ajusta a todas las normas y

reglas que existen para cualquier variable. Dentro de las particularidades del nombre de una matriz es importante que este sea altamente mnemónico o sea que fácilmente haga referencia a su contenido. Esto será muy útil al utilizar la matriz en un determinado algoritmo.

Tipo

Este representa cuál es el tipo de dato que se va a almacenar en cada una de las casillitas de la matriz. Es importante anotar que tanto una matriz como un vector son conjuntos de datos homogéneos esto quiere decir que todos los datos almacenados en ellos deben ser del mismo tipo. De tal forma que usted no podrá pensar en que tiene una matriz de 8 filas por 5 columnas y que en las primeras cuatro filas va a almacenar datos de tipo entero y en las otras cuatro va a almacenar datos de tipo real.

Dimensión

Se refiere específicamente a la cantidad de filas y columnas que va a tener la matriz. Es muy útil que tenga en cuenta que “dimensionar” una matriz no es mas que determinar claramente cuántas filas y cuántas columnas va a tener antes de comenzar cualquier orden. También es bueno que recuerde que el hecho de que se dimensione una matriz no implica la obligatoriedad de utilizar todas las filas o todas las columnas de ella. Por ejemplo, si usted dimensiona una matriz de 1000 filas por 1000 columnas puede utilizar dentro de un algoritmo las primeras 15 filas y 12 columnas. Esta es precisamente una de las grandes desventajas de los arreglos dado que no siempre vamos a saber con certeza cuántas filas y cuántas columnas vamos a necesitar y mas de una vez vamos a pecar por exceso debido a que es posible que dimensionemos mas columnas o mas filas de las que realmente necesitábamos ó por defecto pues también nos pueden hacer falta y haber dimensionado menos de las que eran.

Tamaño

Se refiera al resultado de multiplicar la cantidad total de filas por la cantidad total de columnas de una matriz. De esta manera una matriz de 4 filas por 6 columnas tendrá un tamaño de 24 posiciones. Se denomina tamaño relativo la cantidad total de posiciones que se usan de una matriz. Por ejemplo si se tiene una matriz 6 x 9 (recuerde que el primer número representa la cantidad de filas y el segundo la cantidad de columnas) y de ella usted solo utiliza las 4 primeras filas y las 7 primeras columnas entonces su tamaño absoluto es 6x9 o sea 54 posiciones y su tamaño relativo es 4x7 o sea 28 posiciones. Es obvio pensar que el tamaño relativo es menor o igual que el tamaño absoluto, nunca es mayor.

Destinación

Es muy importante que cuando utilice una matriz dentro de un algoritmo sepa porque la necesita y tenga sentido incorporar en él un conjunto de campos de memoria con formato de filas y columnas.

Ejemplo Con Matrices No.1

Enunciado

Leer una matriz 3 x 4 y determinar en qué posición está el mayor número par.

Clarificación del Objetivo

Fundamentalmente se trata de leer 12 datos enteros, almacenarlos en una matriz que tenga 3 filas y 4 columnas y buscar en qué posición está el mayor número par. Recordemos entonces que un número para es aquel que es divisible exactamente entre 2 y siendo N cualquier número $\text{Si } N / 2 * 2 = N$ entonces N es un número par asumiendo obviamente que N es además un entero para que al realizar las correspondientes operaciones se aplique aritmética entera y no se generen decimales.

Cómo vamos a determinar cuál es el mayor número par..? Recorremos una a una las posiciones dentro de la matriz y vamos preguntando en cada una de ellas si su contenido es par o sea si se cumple con la decisión $\text{Si } N / 2 * 2 = N$ tomando como N cualquier número almacenado en la matriz (o sea que N es un $M(i, j)$ para cualquier valor de i comprendido entre 1 y el tope máximo de las filas y para cualquier valor de j comprendido entre 1 y el tope máximo de las columnas).

En caso de que algún contenido de la matriz sea par entonces lo comparamos con una variable que va a almacenar el mayor número par y si dicho contenido es mayor que el que previamente se había almacenado en esa variable entonces se cambia el dato almacenado en ella por el nuevo valor mayor que se acaba de encontrar. Al tiempo que seleccionamos el mayor para que encontremos vamos almacenando su posición en dos variables debido a que tenemos que almacenar tanto la columna como la fila el número que encontremos.

Finalizado esto mostramos en pantalla lo que nos solicitan que es la posición del mayor número par y de paso podemos decir cuál era ese número encontrado. La variable que va a almacenar el mayor número par ha de ser inicializada en un número muy pequeño para garantizar la efectividad del algoritmo. Pudiera ser inicializada en cero si tenemos la garantía de que todos los datos leídos van a ser positivos. Para efectos de garantizar la efectividad de este algoritmo vamos a inicializar dicha variable con el número -30000, un número verdaderamente pequeño.

Algoritmo

Algoritmo Ejem_Matriz_1

Variables

Entero :M (3 , 4),
Fil,

// Almacenará los 12 datos leídos

// Se utilizará como índice para las filas


```

Col, // Se utilizará como índice para las columnas
Mayor_Par, // Almacenará el mayor número para hallado
Fil_Mayor_Par, // Almacenará la fila en donde se encuentra el
// mayor número par hallado
Col_Mayor_Par, // Almacenará la columna en donde se
// encuentra el mayor número par hallado

Inicio
Escriba " Digite 12 números enteros " // Solicita los 12 números

Para Fil = 1 hasta 3 // y los almacena en una matriz de 3 filas por
    Para Col = 1 hasta 4 // 4 columnas, dejando cada número en cada
        Lea M ( Fil , Col ) // una de las posiciones de la matriz
    Fin_Para
Fin_Para

Mayor_Par = -30000 // Se inicializa la variable que va a almacenar
// el mayor número par en un número muy muy
// pequeño

Para Fil = 1 hasta 3 // Con estos dos ciclos se recorre la matriz (por
// filas) preguntando en cada posición si su
// contenido es par y además si es mayor que
// el último mayor número par encontrado
    Para Col = 1 hasta 4
        Si M ( Fil , Col ) / 2 * 2 = M ( Fil , Col ) Y M ( Fil , Col ) > Mayor_Par
            Mayor_Par = M ( Fil , Col ) // en caso de que la
            Fil_Mayor_Par = Fil // sea Verdadera se
            Col_Mayor_Par = Col // almacena tanto el
// número encontrado
// como la fila y la
// columna en donde se //
// encontró
        Fin_Si
    Fin_Para
Fin_Para

Si Mayor_Par = -30000 // Si esta variable aún almacena -30000 ello
// querrá decir que la matriz no tenía números
// pares
Escriba " No existen números pares en la matriz "
Sino // Si esta variable tiene un valor diferente se
// mostrará en pantalla no solo el número sino
// la posición en donde se encontró
Escriba " El mayor par es ", Mayor_Par, " y está en la fila ", Fil, " columna ", Col
Fin_Si
Fin

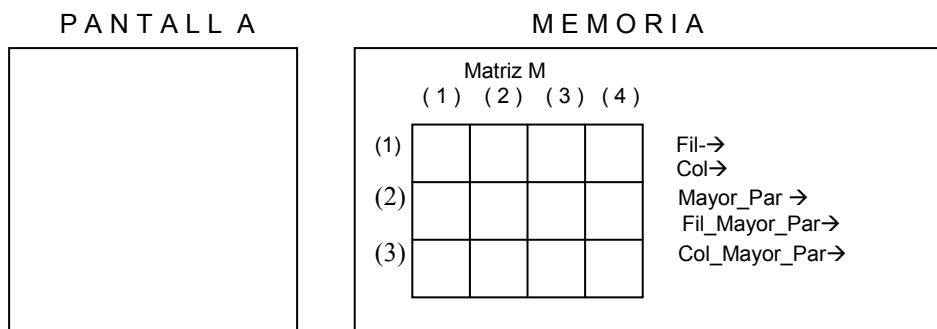
```

Prueba de Escritorio

Lo primero que haremos será declarar en memoria las variables que van a ser necesarias para el desarrollo de este algoritmo.

*Algoritmo Matrices_1**Variables*

Entero : $M(3, 4)$,
Fil,
Col,
Mayor_Par,
Fil_Mayor_Par,
Col_Mayor_Par,



A continuación solicitamos los datos del ejercicio y procedemos a leerlos almacenándolos uno a uno en la matriz destinada para ello. Este almacenamiento se hace en el siguiente orden: Se inicia un ciclo con la variable *Fil* que tomará valores entre 1 y 3 para hacer referencia a cada una de las filas del vector, dentro de este ciclo se tiene otro ciclo en donde la variable *Col* tomará valores entre 1 y 4 para hacer referencia a cada una de las posiciones que existen en cada fila de la matriz o sea, como quien dice, para hacer referencia a cada una de las columnas de la matriz. Dentro de estos dos ciclos se ejecutará la orden *Lea M (Fil , Col)* que significará *Lea un entero y almacénelo en la matriz M en la fila Fil columna Col*.

Inicio

Escriba “ Digite 12 números enteros “

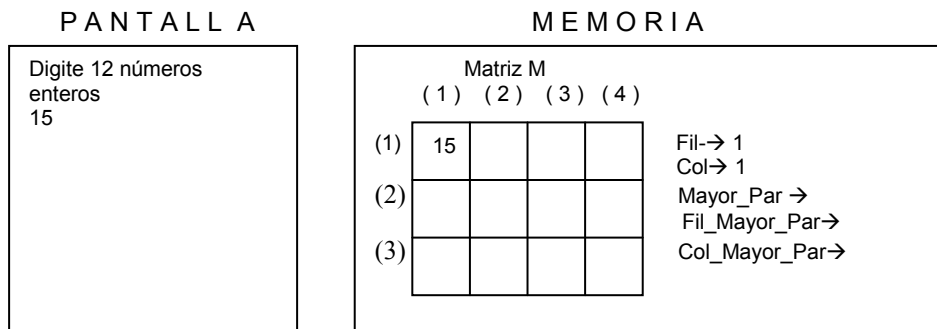
Para Fil = 1 hasta 3

Para Col = 1 hasta 4

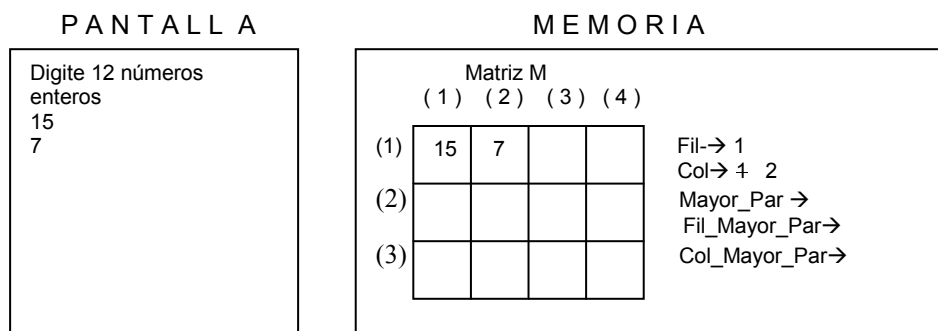
Lea M (Fil , Col)

Fin_Para

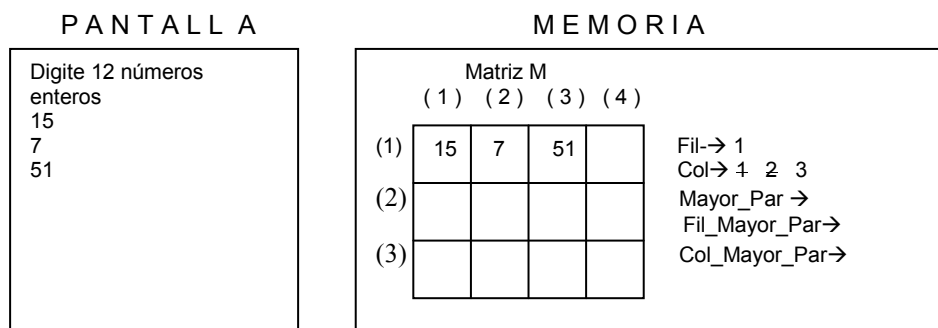
Fin_Para



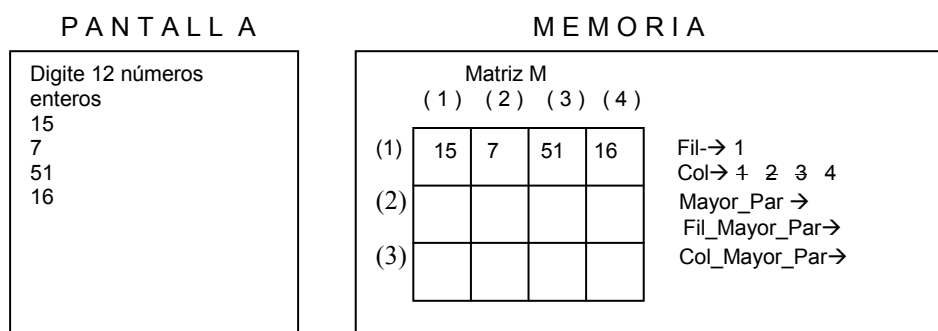
Se inicia la variable Fil con el valor 1 para comenzar el ciclo externo. Se inicia igualmente el ciclo interno entonces cuando Col valga 1 se leerá un entero (supongamos que es el número 15) y se almacenará en la matriz M en la fila 1 columna 1.



Cuando Col valga 2 se leerá otro entero (supongamos que es el número 7) y se almacenará en la matriz M en la fila 1 columna 2. Recuerde que estamos ejecutando el ciclo interno pues solo volvemos al ciclo externo cuando el interno haya terminado.



Cuando Col valga 3 se leerá otro entero (supongamos que es el número 51) y se almacenará en la matriz M en la fila 1 columna 3.



Cuando Col valga 4 se leerá otro entero (supongamos que es el número 16) y se almacenará en la matriz M en la fila 1 columna 4. Con esto finalizaríamos el almacenamiento de datos en una fila (por eso es que se dice que estamos recorriendo la matriz por filas). Como ya llegamos al tope final del ciclo interno pues este iba a generar números entre 1 y 4 utilizando la variable Col como índice entonces nos salimos del ciclo interno y vamos al ciclo externo e incrementamos en 1 el contenido de la variable Fil (o sea que esta queda con el valor 2) con lo cual comenzaríamos a llenar la segunda fila de la matriz.

PANTALLA

Digite 12 números enteros

15

7

51

16

11

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)	15	7	51	16
(2)	11			
(3)				

Fil→ 1 2

Col→ 1

Mayor_Par →

Fil_Mayor_Par→

Col_Mayor_Par→

Se incrementa en 1 el contenido de la variable Fil y se reinicia el ciclo interno o sea que la variable Col comienza de nuevo con el valor 1. De esta manera al ejecutar la orden de lectura y asumiendo que el valor leído fuera el número 11, éste se almacenaría en la matriz M en la fila 2 columna 1. Luego de esto incrementamos en 1 el contenido de la variable Col.

PANTALLA

Digite 12 números enteros

15

7

51

16

11

23

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)	15	7	51	16
(2)	11	23		
(3)				

Fil→ 1 2

Col→ 1 2

Mayor_Par →

Fil_Mayor_Par→

Col_Mayor_Par→

Cuando la variable Col sea igual a 2 entonces se leerá un dato entero (supongamos que es el número 23) y se almacenará en la matriz M en la fila 2 columna 2.

PANTALLA

Digite 12 números enteros

15

7

51

16

11

23

25

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

(3)

Fil→ 1 2

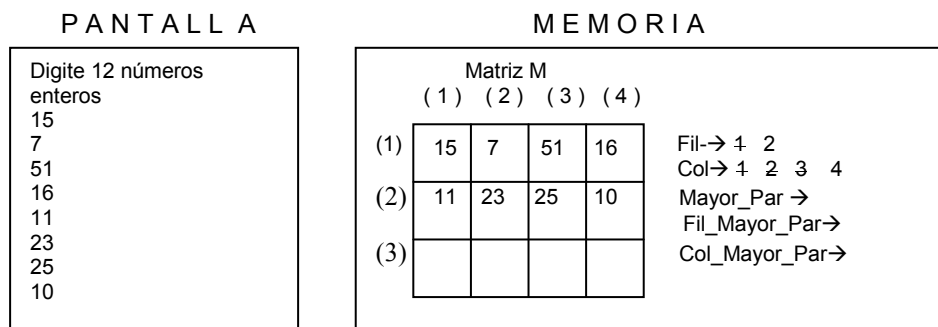
Col→ 1 2 3

Mayor_Par →

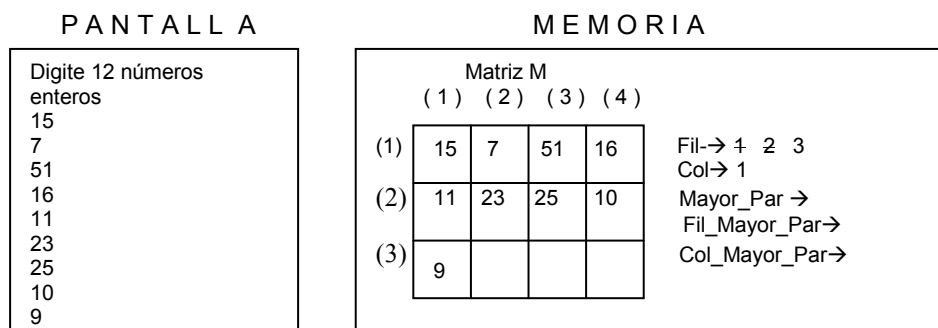
Fil_Mayor_Par→

Col_Mayor_Par→

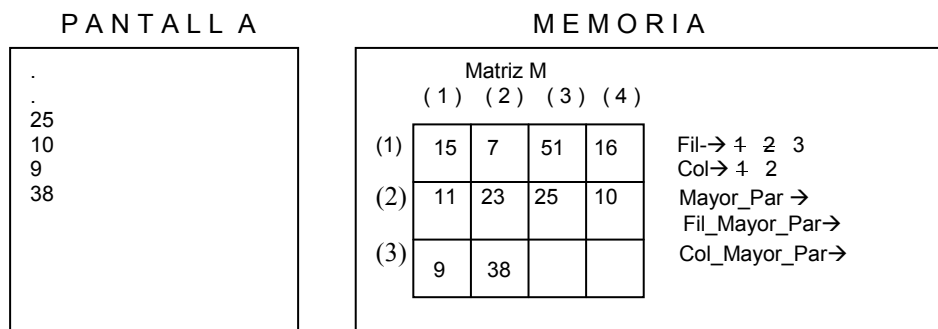
Cuando la variable Col sea igual a 3 entonces se leerá un dato entero (supongamos que es el número 25) y se almacenará en la matriz M en la fila 2 columna 3.



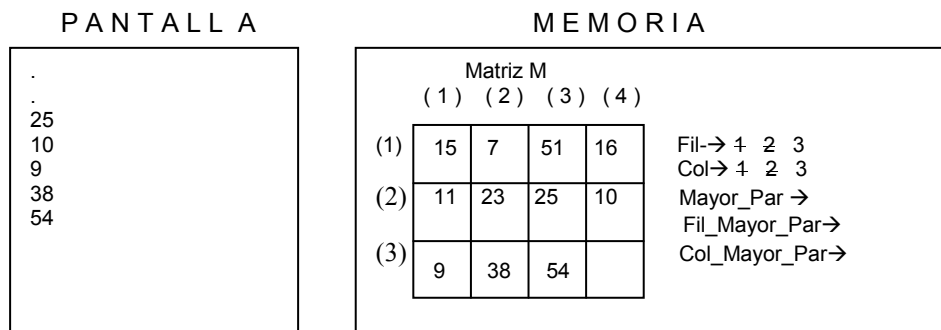
Cuando la variable Col sea igual a 4 entonces se leerá un dato entero (supongamos que es el número 10) y se almacenará en la matriz M en la fila 2 columna 4. En este momento se habrá llegado al tope del ciclo interno y se vuelve al ciclo externo para incrementar en 1 el contenido de la variable Fil. Volvemos entonces a iniciar el ciclo interno o sea que la variable Col volverá a tomar valores desde 1 hasta 4.



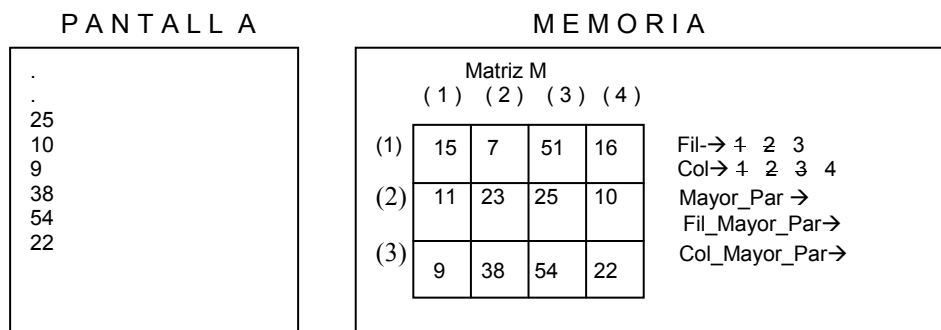
Cuando Col valga 1 entonces se leerá un dato entero (supongamos que es 9) y se almacenará en la matriz M en la fila 3 columna 1. Incrementamos en 1 el contenido de la variable Col y volvemos a leer un dato.



Cuando Col valga 2 entonces se leerá un dato entero (supongamos que es 38) y se almacenará en la matriz M en la fila 3 columna 2. Incrementamos en 1 el contenido de la variable Col y volvemos a leer un dato.



Cuando Col valga 3 entonces se leerá un dato entero (supongamos que es 54) y se almacenará en la matriz M en la fila 3 columna 3. Incrementamos en 1 el contenido de la variable Col y volvemos a leer un dato.



Finalmente cuando Col valga 4 entonces se almacenará el dato leído que asumimos es el número 22. En ese instante terminaremos el ciclo interno pues hemos llegado al tope de este ciclo y también hemos llegado al tope del ciclo externo razón por la cual continuamos con la instrucción que está después del ciclo Para externo.

Inicializamos la variable Mayor_Par con el número -30000 dado que en este número se va a almacenar el mayor número par almacenado en la matriz entonces lo iniciamos con un valor muy pequeño para poder realizar unas comparaciones efectivas. Para ello debemos tener la certeza de que el número -30000 no fue digitado entre los números leídos. Vamos a asumir que en ningún momento y bajo ninguna prueba de escritorio se leerá el número -30000 como parte de los números leídos.

Seguidamente daremos inicio a un ciclo, utilizando la variable Fil como índice, que va a tomar valores entre 1 y 3 (de 1 en 1) y dentro de este ciclo generamos otro ciclo utilizando la variable

Col como índice que va a tomar valores entre 1 y 4. De esta manera estamos establecemos los toques de las variables que van a permitir que se recorra la matriz por filas. En cada posición vamos a preguntar si su contenido es par y si además es mayor que el último número par encontrado. De ser así almacenamos no solo el número encontrado sino la fila y la columna en donde se encuentra dicho número es decir su posición exacta.

$Mayor_Par = -30000$

Para Fil = 1 hasta 3

Para Col = 1 hasta 4

Si $M(Fil, Col) / 2 * 2 = M(Fil, Col)$ Y $M(Fil, Col) > Mayor_Par$

$Mayor_Par = M(Fil, Col)$

$Fil_Mayor_Par = Fil$

$Col_Mayor_Par = Col$

Fin_Si

Fin_Para

Fin_Para

PANTALLA

.
.
38
54
22

MEMORIA

	Matriz M				
	(1)	(2)	(3)	(4)	
(1)	15	7	51	16	Fil → 1
(2)	11	23	25	10	Col → 1
(3)	9	38	54	22	Mayor_Par → -30000
					Fil_Mayor_Par →
					Col_Mayor_Par →

Cuando la variable Fil valga 1 y la variable Col valga 1 la decisión

Si $M(Fil, Col) / 2 * 2 = M(Fil, Col)$ Y $M(Fil, Col) > Mayor_Par$

Se convertirá en

Si $M(1, 1) / 2 * 2 = M(1, 1)$ Y $M(1, 1) > Mayor_Par$

Que a su vez es

Si $15 / 2 * 2 = 15$ Y $15 > -30000$

Vemos que tanto la primera decisión como la segunda son Falsas dado que 15 no es par y además no es mayor que -30000. Por lo tanto incrementamos en 1 el contenido de la variable Col que corresponde al índice del ciclo interno y volvemos a tomar la decisión.

PANTALLA

.
.
38
54
22

MEMORIA

	Matriz M				
	(1)	(2)	(3)	(4)	
(1)	15	7	51	16	Fil → 1
(2)	11	23	25	10	Col → 2
(3)	9	38	54	22	Mayor_Par → -30000
					Fil_Mayor_Par →
					Col_Mayor_Par →

Cuando la variable Fil valga 1 y la variable Col valga 2 la decisión

$$Si\ M(Fil, Col) / 2 * 2 = M(Fil, Col) \ Y\ M(Fil, Col) > Mayor_Par$$

Se convertirá en $Si\ M(1, 2) / 2 * 2 = M(1, 2) \ Y\ M(1, 2) > Mayor_Par$

Que a su vez es $Si\ 7 / 2 * 2 = 7 \ Y\ 7 > -30000$

Vemos que tanto la primera decisión como la segunda son Falsas dado que 7 no es par y además no es mayor que -30000. Por lo tanto incrementamos en 1 el contenido de la variable Col que corresponde al índice del ciclo interno y volvemos a tomar la decisión.

PANTALLA

.

.

38

54

22

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1

Col→ 1 2 3

Mayor_Par → -30000

Fil_Mayor_Par→

Col_Mayor_Par→

Cuando la variable Fil valga 1 y la variable Col valga 3 la decisión

$$Si\ M(Fil, Col) / 2 * 2 = M(Fil, Col) \ Y\ M(Fil, Col) > Mayor_Par$$

Se convertirá en $Si\ M(1, 3) / 2 * 2 = M(1, 3) \ Y\ M(1, 3) > Mayor_Par$

Que a su vez es $Si\ 51 / 2 * 2 = 51 \ Y\ 51 > -30000$

Vemos que tanto la primera decisión como la segunda son Falsas dado que 7 no es par y además no es mayor que -30000. Por lo tanto incrementamos en 1 el contenido de la variable Col que corresponde al índice del ciclo interno y volvemos a tomar la decisión.

PANTALLA

.

.

38

54

22

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1

Col→ 1 2 3 4

Mayor_Par → -30000

Fil_Mayor_Par→

Col_Mayor_Par→

Cuando la variable Fil valga 1 y la variable Col valga 4 la decisión

$$\text{Si } M(\text{Fil}, \text{Col}) / 2 * 2 = M(\text{Fil}, \text{Col}) \text{ Y } M(\text{Fil}, \text{Col}) > \text{Mayor_Par}$$

Se convertirá en $\text{Si } M(1, 4) / 2 * 2 = M(1, 4) \text{ Y } M(1, 4) > \text{Mayor_Par}$

Que a su vez es $\text{Si } 16 / 2 * 2 = 16 \text{ Y } 16 > -30000$

Podemos notar entonces que el número 16 es par y además es mayor que el número –30000 (tomando como base la recta de números enteros) por lo tanto como ambas condiciones son verdaderas entonces toda la decisión es verdadera dado que están unidas a través de un operador booleano Y. Por lo tanto ejecutamos las órdenes

$\text{Mayor_Par} = M(\text{Fil}, \text{Col})$
 $\text{Fil_Mayor_Par} = \text{Fil}$
 $\text{Col_Mayor_Par} = \text{Col}$

Que, tomando los valores correspondientes de Fil y Col se convertirán en

$\text{Mayor_Par} = 16$
 $\text{Fil_Mayor_Par} = 1$
 $\text{Col_Mayor_Par} = 4$

En este instante hemos llegado al tope de Col (que era 4) por lo tanto nos salimos al ciclo externo e incrementamos en 1 el contenido de la variable Fil para volver a reiniciar el ciclo interno desde 1 hasta 4.

PANTALLA

.

38

54

22

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1 2

Col→ 1

Mayor_Par → -30000

Fil_Mayor_Par→ 1

Col_Mayor_Par→ 4

Cuando la variable Fil valga 2 y la variable Col valga 1 la decisión

$$\text{Si } M(\text{Fil}, \text{Col}) / 2 * 2 = M(\text{Fil}, \text{Col}) \text{ Y } M(\text{Fil}, \text{Col}) > \text{Mayor_Par}$$

Se convertirá en $\text{Si } M(2, 1) / 2 * 2 = M(2, 1) \text{ Y } M(2, 1) > \text{Mayor_Par}$

Que a su vez es $\text{Si } 11 / 2 * 2 = 11 \text{ Y } 11 > 16$

Vemos que tanto la primera decisión como la segunda son Falsas dado que 11 no es par y además no es mayor que -30000. Por lo tanto incrementamos en 1 el contenido de la variable Col que corresponde al índice del ciclo interno y volvemos a tomar la decisión.

PANTALLA

.

.

38

54

22

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1 2

Col→ 1 2

Mayor_Par → -30000

Fil_Mayor_Par→ 1

Col_Mayor_Par→ 4

Cuando la variable Fil valga 2 y la variable Col valga 2 la decisión

$$Si M(Fil, Col) / 2 * 2 = M(Fil, Col) \text{ Y } M(Fil, Col) > Mayor_Par$$

Se convertirá en $Si M(2, 2) / 2 * 2 = M(2, 2) \text{ Y } M(2, 2) > Mayor_Par$

Que a su vez es $Si \quad 23 \quad / 2 * 2 = \quad 23 \quad \text{ Y } \quad 23 \quad > \quad 16$

Vemos que tanto la primera decisión como la segunda son Falsas dado que 23 no es par y además no es mayor que -30000. Por lo tanto incrementamos en 1 el contenido de la variable Col que corresponde al índice del ciclo interno y volvemos a tomar la decisión.

PANTALLA

.

.

38

54

22

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1 2

Col→ 1 2 3

Mayor_Par → -30000

Fil_Mayor_Par→ 1

Col_Mayor_Par→ 4

Cuando la variable Fil valga 2 y la variable Col valga 3 la decisión

$$Si M(Fil, Col) / 2 * 2 = M(Fil, Col) \text{ Y } M(Fil, Col) > Mayor_Par$$

Se convertirá en $Si\ M(2, 3) / 2 * 2 = M(2, 3) \ Y\ M(2, 3) > Mayor_Par$

Que a su vez es $Si\ 25 / 2 * 2 = 25 \ Y\ 25 > 16$

Vemos que tanto la primera decisión como la segunda son Falsas dado que 25 no es par y además no es mayor que -30000. Por lo tanto incrementamos en 1 el contenido de la variable Col que corresponde al índice del ciclo interno y volvemos a tomar la decisión.

PANTALLA

.

.

38

54

22

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1 2

Col→ 1 2 3 4

Mayor_Par → -30000

Fil_Mayor_Par→ 1

Col_Mayor_Par→ 4

Cuando la variable Fil valga 2 y la variable Col valga 4 la decisión

$Si\ M(Fil, Col) / 2 * 2 = M(Fil, Col) \ Y\ M(Fil, Col) > Mayor_Par$

Se convertirá en $Si\ M(2, 4) / 2 * 2 = M(2, 4) \ Y\ M(2, 4) > Mayor_Par$

Que a su vez es $Si\ 10 / 2 * 2 = 10 \ Y\ 10 > 16$

Podemos notar que el número 10 es par o sea que la primera decisión es verdadera pero la segunda decisión es Falsa dado que el número 10 no es mayor que el número 16 por lo tanto como están unidas con un operador booleano Y y según este para que toda la condición sea Verdadera tendrían que ser Verdaderas sus partes, entonces toda la condición es Falsa. Por lo tanto nos salimos del ciclo interno debido a que ya llegamos a su tope y volvemos al ciclo externo a incrementar en 1 el contenido de la variable Fil. Con esto volvemos a entrar e ejecutar el ciclo interno iniciando la variable Col en 1 y llevándola de 1 en 1 hasta 4.

PANTALLA

.

.

38

54

22

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1 2 3

Col→ 1

Mayor_Par → -30000

Fil_Mayor_Par→ 1

Col_Mayor_Par→ 4

Cuando la variable Fil valga 3 y la variable Col valga 1 la decisión

$$\text{Si } M(\text{Fil}, \text{Col}) / 2 * 2 = M(\text{Fil}, \text{Col}) \text{ Y } M(\text{Fil}, \text{Col}) > \text{Mayor_Par}$$

Se convertirá en $\text{Si } M(3, 1) / 2 * 2 = M(3, 1) \text{ Y } M(3, 1) > \text{Mayor_Par}$

Que a su vez es $\text{Si } 9 / 2 * 2 = 9 \text{ Y } 9 > 16$

Podemos notar que el número 9 no es par ni tampoco es mayor que 16 por lo tanto toda la condición es Falsa. Incrementamos entonces en 1 el contenido de la variable Col.

PANTALLA

.

.

38

54

22

Matriz M

(1)(2)(3)(4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→123

Col→12

Mayor_Par→-30000

Fil_Mayor_Par→1

Col_Mayor_Par→4

Cuando la variable Fil valga 3 y la variable Col valga 2 la decisión

$$\text{Si } M(\text{Fil}, \text{Col}) / 2 * 2 = M(\text{Fil}, \text{Col}) \text{ Y } M(\text{Fil}, \text{Col}) > \text{Mayor_Par}$$

Se convertirá en $\text{Si } M(3, 2) / 2 * 2 = M(3, 2) \text{ Y } M(3, 2) > \text{Mayor_Par}$

Que a su vez es $\text{Si } 38 / 2 * 2 = 38 \text{ Y } 38 > 16$

Podemos notar que el número 38 es par y a la vez es mayor que el número 16 por lo tanto ejecutamos las órdenes

```

Mayor_Par = M(Fil, Col)
Fil_Mayor_Par = Fil
Col_Mayor_Par = Col

```

Que tomando sus correspondientes valores se convierte en

```

Mayor_Par = 38
Fil_Mayor_Par = 3
Col_Mayor_Par = 2

```

Y volvemos a incrementar en 1 el contenido de la variable Col.

PANTALLA

.

.

38

54

22

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1 2 3

Col→ 1 2 3

Mayor_Par→ 30000 16

38

Fil_Mayor_Par→ 1 3

Col_Mayor_Par→ 4 2

Cuando la variable Fil valga 3 y la variable Col valga 3 la decisión

$$Si M(Fil, Col) / 2 * 2 = M(Fil, Col) \text{ Y } M(Fil, Col) > Mayor_Par$$

Se convertirá en $Si M(3, 3) / 2 * 2 = M(3, 3) \text{ Y } M(3, 3) > Mayor_Par$

Que a su vez es $Si \quad 54 \quad / 2 * 2 = \quad 54 \quad \text{ Y } \quad 54 > 38$

Como el número 54 es par y además es mayor que 38 entonces se ejecutan las órdenes

$Mayor_Par = M(Fil, Col)$
 $Fil_Mayor_Par = Fil$
 $Col_Mayor_Par = Col$

Que tomando sus correspondientes valores se convierte en

$Mayor_Par = 54$
 $Fil_Mayor_Par = 3$
 $Col_Mayor_Par = 3$

Y volvemos a incrementar en 1 el contenido de la variable Col.

PANTALLA

.

.

38

54

22

MEMORIA

Matriz M

(1) (2) (3) (4)

(1)

15

7

51

16

(2)

11

23

25

10

(3)

9

38

54

22

Fil→ 1 2 3

Col→ 1 2 3 4

Mayor_Par→ 30000 16

38 54

Fil_Mayor_Par→ 1 3 3

Col_Mayor_Par→ 4 2 3

Cuando la variable Fil valga 3 y la variable Col valga 4 la decisión

$$\text{Si } M(\text{Fil}, \text{Col}) / 2 * 2 = M(\text{Fil}, \text{Col}) \text{ Y } M(\text{Fil}, \text{Col}) > \text{Mayor_Par}$$

Se convertirá en $\text{Si } M(3, 4) / 2 * 2 = M(3, 4) \text{ Y } M(3, 4) > \text{Mayor_Par}$

Que a su vez es $\text{Si } 22 / 2 * 2 = 22 \text{ Y } 22 > 54$

La primera decisión es verdadera debido a que el número 22 es par pero la segunda condición es falsa pues 22 no es mayor que 54 entonces, debido a la presencia del operador booleano Y, toda la expresión es Falsa. Con esto hemos llegado al tope del ciclo interno y también hemos llegado al tope del ciclo externo entonces a continuación nos salimos de ambos ciclos y ejecutamos la orden que se encuentra después del Fin_Para del ciclo externo.

Preguntamos entonces por el contenido de la variable Mayor_Par debido a que si esta variable todavía almacena el número -30000 (y teniendo nosotros toda la seguridad que este número en ningún momento fue digitado) esto querrá decir que en toda la matriz no habían números pares y por lo tanto así lo avisaremos al usuario del programa. En este caso esta decisión es Falsa debido a que el contenido actual de la variable Mayor_Par es 54 entonces se ejecuta el correspondiente Sino de esta pregunta. Por tanto escribiríamos en pantalla la información solicitada desde el principio tomando los últimos valores de las variables correspondientes.

Si Mayor_Par = -30000

Escriba "No existen números pares en la matriz"

Sino

Escriba "El mayor par es ", Mayor_Par, " y está en la fila ", Fil, " columna ", Col

Fin_Si

PANTALLA

```
.
38
54
22

El mayor par es 54 y está
en la fila 3 columna 3
```

MEMORIA

Matriz M				
	(1)	(2)	(3)	(4)
(1)	15	7	51	16
(2)	11	23	25	10
(3)	9	38	54	22

Fil → 1 2 3
 Col → 1 2 3 4
 Mayor_Par → -30000 16
 38 54
 Fil_Mayor_Par → 1 3 3
 Col_Mayor_Par → 4 2 3

Finalmente encontramos el fin del algoritmo y con ello termina nuestra prueba de escritorio.

Fin

Ahora podemos ver que si los datos leídos hubieran sido los que aparecen en la matriz, el algoritmo efectivamente hubiera detectado el mayor par y la posición en la cuál se encontraba pues el número 54 es el mayor par (de este grupo de números) y realmente se encuentra en la fila 3 columna 3 de esta matriz. Por lo tanto podemos decir que este algoritmo está correcto.

Ejemplo Con Matrices No.2

Enunciado

Leer una matriz 4x4 y determinar a cuánto es igual la suma de los elementos que se encuentran en su diagonal.

Clarificación del Objetivo

Se define como Matriz Cuadrada toda aquella matriz en donde la cantidad de filas es igual a la cantidad de columnas. Se define como Diagonal de una matriz todos los elementos que se encuentran en posiciones en donde la fila es igual a la columna. Por ejemplo si tenemos la siguiente matriz M de 4 x 4 (o sea de 4 filas y 4 columnas)

Columnas.....>		(1)	(2)	(3)	(4)
Filas.....>	(1)	18	16	25	24
	(2)	15	54	65	12
	(3)	54	45	58	21
	(4)	45	41	74	32

Los números 18, 54, 58, 32 corresponden a los que están ubicados en la diagonal de esta matriz. Porqué? Pues sencillamente porque son los números ubicados en las posiciones en donde la fila es igual a la columna. El número 18 está en la Fila 1 Columna 1, el número 54 está en la Fila 2 Columna 2, el número 58 está en la Fila 3 Columna 3 y el número 32 está en la Fila 4 Columna 4. Por esta razón es que se hablaba al principio de las matrices cuadradas dado que son las que realmente tienen definida la diagonal.

Columnas.....>		(1)	(2)	(3)	(4)
Filas.....>	(1)	18	16	25	24
	(2)	15	54	65	12
	(3)	54	45	58	21
	(4)	45	41	74	32

De acuerdo a lo dicho anteriormente nuestro algoritmo busca realmente leer todos los datos de una matriz de datos enteros de 4 filas por 4 columnas y sumar solo los elementos que se encuentran en la diagonal. Luego lo que primero vamos a hacer será leer todos y cada uno de los datos que han de quedar almacenados en la matriz generando dos ciclos anidados: uno externo que permita referenciar las filas y otro interno que permita referenciar las columnas (dentro de cada fila). Cuando ya la matriz esté completamente llena de datos o sea cuando ambos ciclos hayan terminado entonces procederemos a generar un ciclo con una sola variable que vaya desde 1 hasta 4 y con esa misma variable vamos a referenciar filas y columnas al mismo tiempo. De manera que si esa es la variable Ind entonces cada elemento a sumar (en un acumulador por supuesto) será el elemento $M(Ind, Ind)$ entonces con ello estaremos referenciando solamente a los elementos de la diagonal.

Cuando se haya realizado el recorrido entonces todo lo que tenemos que hacer es mostrar en pantalla el valor acumulado en la variable correspondiente y con ello habremos cumplido con nuestro objetivo.

Algoritmo

Programa Suma_Diagonal

Variables

Entero :M (4 , 4),

I,

J,

Diag

// Matriz en donde se almacenarán los

// datos leídos

// Índice de referencia

// Índice de referencia

// Variable que almacenará el resultado

// de sumar todos los elementos de la

// diagonal

Inicio

Escriba "Digite 16 números enteros "

// Solicita los datos a leer


```

I = 1                                     // Inicializa la variable I en el índice de
Mientras I <= 4                           // la primera fila
    J = 1                                 // Mientras no haya llegado a la última
    Mientras J <= 4                       // fila
        Lea M ( I, J )                   // Comienza en la primera posición
        J = J + 1                         // dentro de la fila
    Fin_Mientras                          // Mientras no haya llegado a la última
    I = I + 1                             // columna dentro de la fila
    Fin_Mientras                          // Lea un entero y almacénelo en la
                                         // matriz M Columna I Fila J
                                         // Pase a siguiente columna dentro de
                                         // la fila
                                         // Fin del ciclo interno
                                         // Pase a la siguiente fila
                                         // Fin del ciclo externo

Diag = 0                                 // Inicialice la variable Diag en 0

I = 1                                     // Inicialice la variable I en 1
Mientras I <= 4                           // Mientras no haya llegado a la última
    Diag = Diag + M ( I, I )              // posición
    I = I + 1                             // Acumule en la variable Diag la suma
    Fin_Mientras                         // de los elementos de la diagonal
                                         // Pase a la siguiente posición
                                         // Fin del ciclo de recorrido

                                         // Muestre el resultado solicitado en
                                         // pantalla
Escriba "La suma de los elementos de la diagonal es igual a ", Diag

Fin                                       // Fin del algoritmo

```

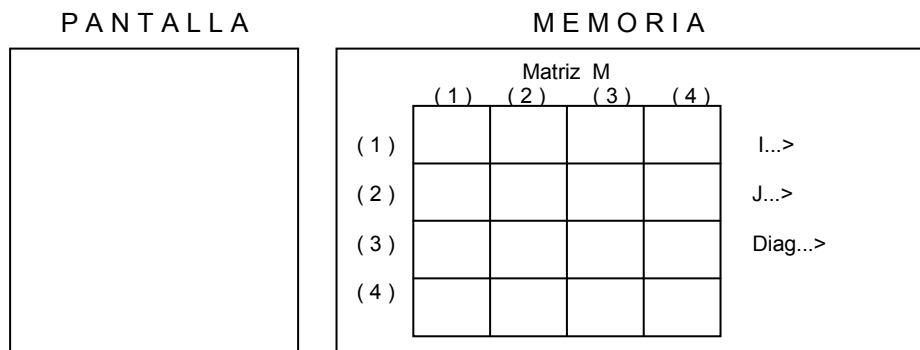
Prueba de Escritorio

Al igual que todos los algoritmos lo primero que hacemos es declarar en memoria todas las variables que vamos a necesitar

Programa Suma_Diagonal

Variables

Entero :M (4, 4), I, J, Diag



Una vez declaradas en memoria las variables que necesitamos, en nuestro algoritmo siguen las instrucciones

Inicio

Escriba "Digite 16 números enteros "

I = 1

Mientras I <= 4

J = 1

Mientras J <= 4

Lea M (I, J)

J = J + 1

Fin_Mientras

I = I + 1

Fin_Mientras

Según las cuales estaríamos leyendo 16 números enteros y los estaríamos almacenando en cada una de las posiciones de la matriz. Iniciamos, pues, la variable I en 1 y mientras el contenido de esta variable sea menor o igual que 4 vamos a desarrollar el conjunto de instrucciones que se encuentran dentro de este ciclo. Este conjunto de instrucciones comienza asignando 1 a la variable J y mientras el contenido de esta variable sea menor o igual que cuatro se deberá leer un dato entero y almacenarse en la matriz M en posición I columna J y luego se deberá incrementar el contenido de la variable J en 1. Cuando se termine este ciclo interno se deberá incrementar el contenido de la variable I y se deberá volver a evaluar la condición del ciclo externo. De esta manera cuando la variable I valga 1 y la variable J valga 1 se leerá un dato entero. Supongamos que es el número 18, luego de lo cual incrementaremos el valor de J en 1.

PANTALLA

Digite 16 números
enteros
18

MEMORIA

Matriz M				
	(1)	(2)	(3)	(4)
(1)	18			
(2)				
(3)				
(4)				

I...> 1
J...> 1
Diag...>

Incrementamos en 1 el contenido de la variable J y volvemos a evaluar la condición mientras J sea menor o igual que 4. Como es Verdadero entonces volvemos a leer otro dato y lo almacenamos en la posición correspondiente. Supongamos que el otro dato leído sea el número 35.

PANTALLA

Digite 16 números
enteros
18
35

MEMORIA

Matriz M				
	(1)	(2)	(3)	(4)
(1)	18	35		
(2)				
(3)				
(4)				

I...> 1
J...> 4 2
Diag...>

No se olvide que cada dato se almacenará en la fila I columna J. Volvemos a incrementar en 1 el valor almacenado en J y volvemos a evaluar la condición de este ciclo interno. Como vemos que aún es Verdadera debido a que el valor almacenado en J sigue siendo menor o igual que 4 entonces volvemos a leer otro dato y lo almacenamos en la Fila 1 Columna 3.

PANTALLA

Digite 16 números enteros

18

35

10

MEMORIA

Matriz M

	(1)	(2)	(3)	(4)
(1)	18	35	10	
(2)				
(3)				
(4)				

I...> 1

J...> 1 2 3

Diag...>

De nuevo, incrementamos por última vez el contenido de la variable J y leemos un dato para ser almacenado en la fila 1 columna 4. Supongamos que el número leído es el 13.

PANTALLA

Digite 16 números enteros

18

35

10

13

MEMORIA

Matriz M

	(1)	(2)	(3)	(4)
(1)	18	35	10	14
(2)				
(3)				
(4)				

I...> 1

J...> 1 2 3 4

Diag...>

En este instante al volver a incrementar en 1 el contenido de la variable J vemos que ya no se cumple la condición de que siga siendo menor o igual que 4 por lo tanto nos salimos de este ciclo interno para ejecutar la instrucción que sigue después de su correspondiente Fin_Mientras. Dicha instrucción representa incrementar en 1 el contenido de la variable I por lo tanto realizamos este incremento y volvemos a evaluar la condición de dicho ciclo externo como vemos que es Verdadera dado que el contenido de la variable I aún es menor o igual que 4 entonces volvemos a inicializar la variable J en 1 para entrar en su correspondiente ciclo. Cuando esta variable valga 1 (de nuevo) entonces se leerá un dato entero y se almacenará en la fila 2 columna 1, debido a que estos son los valores de I y J respectivamente. Supongamos que el nuevo dato leído sea 23.

PANTALLA

Digite 16 números enteros

18

35

10

13

23

MEMORIA

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18	35	10	14
23			

I...> 4 2

J...> 1

Diag...>

Incrementamos el valor almacenado en J en 1 y evaluamos la condición de este ciclo. Como es Verdadera entonces volvemos a leer otro dato (supongamos que es el número 8) y lo almacenamos en la Fila 2 Columna 2.

PANTALLA

Digite 16 números enteros

18

35

10

13

23

8

MEMORIA

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18	35	10	14
23	8		

I...> 4 2

J...> 4 2

Diag...>

De nuevo incrementamos en 1 el contenido de la variable J y volvemos a evaluar la condición. Como dicho contenido sigue siendo menor que 4 entonces volvemos a leer otro dato y lo almacenamos en la posición correspondiente.

PANTALLA

Digite 16 números enteros

18

35

10

13

23

8

11

MEMORIA

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18	35	10	14
23	8	11	

I...> 4 2

J...> 4 2- 3

Diag...>

Luego incrementamos de nuevo el contenido de la variable J en 1 y realizamos una lectura de un dato que quedaría almacenado en la fila 2 columna 4. Supongamos que se lee un 45.

PANTALLA

Digite 16 números enteros

18

35

10

13

23

8

11

45

MEMORIA

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18	35	10	14
23	8	11	45

I...> 4 2

J...> 4 2 3 4

Diag...>

En este momento al volver a incrementar en 1 el contenido de la variable J vemos que ya no cumple la condición de seguir siendo menor o igual que 4 (pues valdría 5) y entonces nos salimos de este ciclo interno. Incrementamos en 1 el contenido de la variable I y volvemos a evaluar la condición del ciclo externo. Como dicho contenido es menor o igual que 4 entonces entramos al ciclo interno, es decir, volvemos a inicializar la variable J en 1 y como dicho valor es menor o igual que 4 entonces leemos un dato entero y lo almacenamos en la fila 3 columna 1.

PANTALLA

.

.

.

45

11

MEMORIA

Matriz M

	(1)	(2)	(3)	(4)
(1)	18	35	10	14
(2)	23	8	11	45
(3)	11			
(4)				

I...> 4 2 3

J...> 1

Diag...>

Incrementamos el contenido de la variable J en 1 y evaluamos la condición del ciclo interno. Como el valor almacenado en J es menor o igual que 4 volvemos a leer un dato (supongamos que es el número 88) y lo almacenamos en la fila 3 columna 3.

PANTALLA

45

11

88

MEMORIA

Matriz M

	(1)	(2)	(3)	(4)
(1)	18	35	10	14
(2)	23	8	11	45
(3)	11	88		
(4)				

I...> 4 2 3

J...> 4 2

Diag...>

De nuevo incrementamos en 1 el contenido de la variable J y al evaluar la condición el ciclo interno vemos que sigue siendo menor que 4 por lo tanto leemos otro dato y lo almacenamos en la posición correspondiente.

PANTALLA

.

.

.

45

11

88

16

MEMORIA

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18	35	10	14
23	8	11	45
11	88	16	

I...> 4 2 3

J...> 4 2 3

Diag...>

Se incrementa el contenido de la variable J y se evalúa la condición del ciclo interno. Como es Verdadera entonces leemos otro dato y lo almacenamos en la matriz M en la fila 3 columna 3.

PANTALLA

.

.

.

45

11

88

16

51

MEMORIA

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18

35

10

14

23

8

11

45

11

88

16

51

I...> 1 2 3

J...> 1 2 3 4

Diag...>

Al volver a incrementar en 1 el contenido de la variable J vemos que la condición deja de ser Verdadera y por lo tanto nos salimos de este ciclo interno e incrementamos en 1 el contenido de la variable I, después de lo cual evaluamos la condición del ciclo externo. Como el valor almacenado en la variable I sigue siendo Verdadero entonces volvemos a inicializar la variable J en 1 y volvemos a leer un dato entero que para este caso vamos a asumir que es el número 13 y lo almacenamos en la fila 4 columna 1.

PANTALLA

.

.

.

45

11

88

16

51

13

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18

35

10

14

23

8

11

45

11

88

16

51

13

I...> 4 2 3 4

J...> 1

Diag...>

Realizamos las mismas operaciones para llenar la cuarta fila de manera que se obtenga llenar la matriz completamente.

PANTALLA

.

.

.

51

13

22

MEMORIA

Matriz M

(1)

(2)

(3)

(4)

(1)

(2)

(3)

(4)

18

35

10

14

23

8

11

45

11

88

16

51

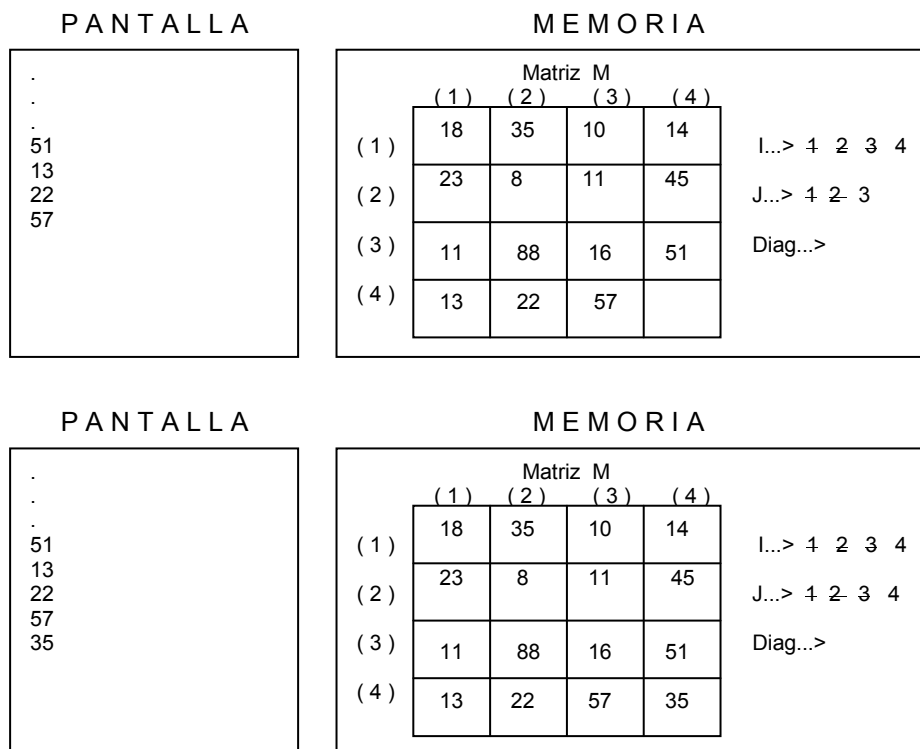
13

22

I...> 1 2 3 4

J...> 1 2

Diag...>



Teniendo ya la matriz llena vemos que al incrementar en 1 el contenido de la variable J, esta ya no es menor o igual que 4 razón por la cual nos salimos al ciclo externo e incrementamos de nuevo el valor almacenado en la variable I con lo cual vemos igualmente que dicho valor no es menor o igual que 4 por lo cual nos salimos también este ciclo externo. El conjunto de instrucciones que sigue nos va a permitir realizar la suma de los elementos que se encuentran ubicados en la diagonal de la matriz. Inicializamos la variable *Diag* en 0 y la variable I en 1. A continuación evaluamos la condición y vemos efectivamente que I es menor o igual que 4 por lo tanto realizamos la operación de suma $Diag = Diag + M(I, I)$ e incrementamos en 1 el contenido de la variable I.

Diag = 0

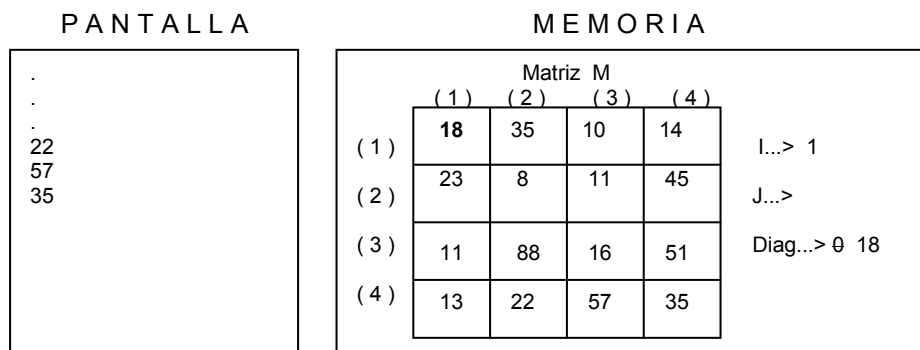
I = 1

Mientras I <= 4

Diag = *Diag* + M (I , I)

I = I + 1

Fin_Mientras



Como la variable I almacena el valor 1 entonces la expresión $Diag = Diag + M(I, I)$
Se convierte en $Diag = Diag + M(1, 1)$

Como el valor almacenado en la fila 1 columna 1 de la matriz M es 18 entonces finalmente la expresión se convierte en $Diag = Diag + 18$

Seguidamente incrementamos en 1 el contenido de la variable I y evaluamos la condición de este ciclo. Vemos que el valor almacenado en I es menor o igual que 4 por lo tanto volvemos a resolver la expresión $Diag = Diag + M(I, I)$.

PANTALLA

Como la variable I almacena el valor 1 entonces la expresión $Diag = Diag + M(I, I)$
Se convierte en $Diag = Diag + M(2, 2)$

Como el valor almacenado en la fila 2 columna 2 de la matriz M es 8 entonces finalmente la expresión se convierte en $Diag = Diag + 18$ quedando almacenado en la variable $Diag$ el número 26.

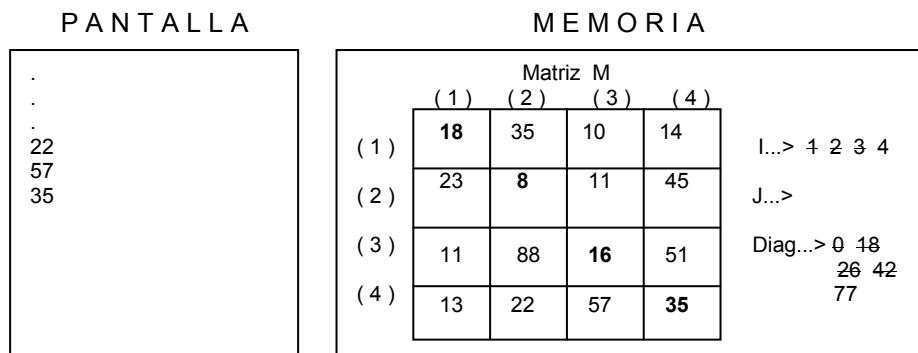
Seguidamente incrementamos en 1 el contenido de la variable I y evaluamos la condición de este ciclo. Vemos que el valor almacenado en I es menor o igual que 4 por lo tanto volvemos a resolver la expresión $Diag = Diag + M(I, I)$.

PANTALLA

Como la variable I almacena el valor 1 entonces la expresión $Diag = Diag + M(I, I)$
Se convierte en $Diag = Diag + M(3, 3)$

Como el valor almacenado en la fila 3 columna 3 de la matriz M es 8 entonces finalmente la expresión se convierte en $Diag = Diag + 16$ quedando almacenado en la variable $Diag$ el número 42.

Seguidamente incrementamos en 1 el contenido de la variable I y evaluamos la condición de este ciclo. Vemos que el valor almacenado en I es menor o igual que 4 por lo tanto volvemos a resolver la expresión $Diag = Diag + M(I, I)$.



Como la variable I almacena el valor 4 entonces la expresión

$$Diag = Diag + M (I , I)$$

Se convierte en

$$Diag = Diag + M (4 , 4)$$

Como el valor almacenado en la fila 4 columna 4 de la matriz M es 8 entonces finalmente la expresión se convierte en

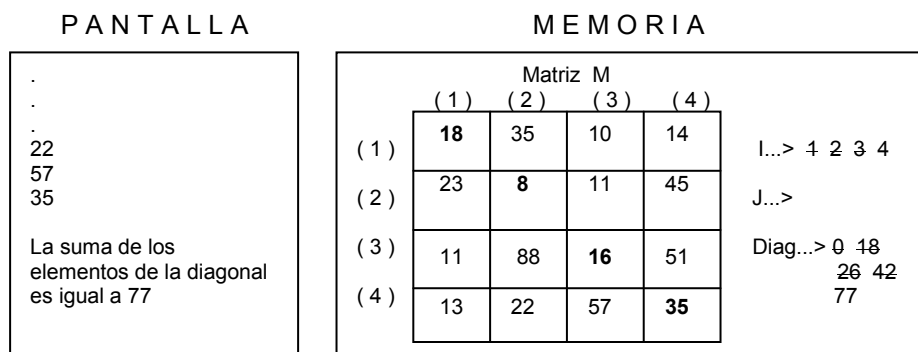
$$Diag = Diag + 35 \text{ quedando}$$

almacenado en la variable *Diag* el número 77.

Volvemos a incrementar el contenido de la variable I y vemos que ya no se cumple la condición de que dicho contenido sea menor o igual que 4 razón por la cual nos salimos de este ciclo y pasamos a la instrucción

Escriba “La suma de los elementos de la diagonal es igual a “, Diag

Con la cual saldrá en pantalla



Después de lo cual encontramos el fin del algoritmo.

Fin

Al revisar los resultados vemos que efectivamente el número mostrado en pantalla al final es igual a la suma de los datos que quedaron ubicados en la diagonal de la matriz por lo tanto podemos dar fé de que este algoritmo cumple con el objetivo planteado que *era Leer una matriz 4x4 y determinar a cuánto es igual la suma de los elementos que se encuentran en su diagonal*

Ejemplo Con Matrices No.3

Enunciado

Leer una matriz 4 x 3 y determinar cuántas veces se repite el mayor de los números almacenados en ella.

Clarificación del Objetivo

Ya sabemos que deben leerse 12 números enteros e irse almacenando uno a uno en las diferentes posiciones de la matriz. Para ello, y tal como lo hemos hecho en los casos anteriores, utilizamos dos ciclos anidados que nos permitan recorrer por filas la matriz al tiempo que vamos leyendo datos enteros y los vamos almacenando en cada una de sus posiciones.

Luego de tener “cargada” la matriz buscamos cuál es el mayor dato almacenado en ella y por lo tanto para ello utilizamos una variable que inicializamos en un valor muy pequeño y contra ella vamos comparando uno a uno los datos almacenados en la matriz. Cada vez que encontremos que un dato es mayor que el número almacenado en la matriz entonces deberemos guardar en dicha variable ese último número mayor encontrado. Este proceso lo haremos al tiempo que recorremos toda la matriz desde la primera fila hasta la última fila posición a posición.

Cuando ya se haya recorrido toda la matriz y tengamos almacenado el número mayor entonces volveremos a recorrer la matriz para poder contar cuántas veces se repite dicho número en ella evaluando si cada dato de la matriz es igual al dato almacenado en la variable que nos va a guardar el mayor número encontrado. Es importante que tenga en cuenta que este proceso solo se podrá realizar cuando se tenga completamente identificado el número mayor. Quiero decir con esto que no podemos contar la cantidad de veces que se encuentra el mayor al tiempo que lo buscamos.

Algoritmo

Programa Cuenta_Mayor

Variables

Entero :M (4, 3),

I,

J,

// Matriz que almacenará los números

// leídos

// Índice de referencia

// Índice de referencia

```

        May_Num,                // Variable que almacenará el mayor de
                                // los números almacenados en la
                                // matriz
        Cont_May                // Variable que almacenará la cantidad
                                // de veces que se encuentra el mayor
                                // valor en la matriz

Inicio
    Escriba " Digite 12 números enteros "    // Solicita los datos que va a leer

    Para I = 1 hasta 4                    // Con este índice se van a referenciar
                                        // las filas
        Para J = 1 hasta 3                // Con este índice se van a referenciar
                                        // las columnas
            Lea M ( I, J )                // Lea un dato entero y almacénelo en
                                        // la matriz M en la fila I columna J
        Fin_Para                        // Fin del ciclo interno
    Fin_Para                            // Fin del ciclo externo

    May_Num = -30000                    // Inicializamos esta variable en un
                                        // valor muy muy pequeño

    Para I = 1 hasta 4                    // Desde la primera hasta la última fila
        Para J = 1 hasta 3                // Desde la primera hasta la última
                                        // posición en cada fila
            Si M ( I, J ) > May_Num // Si algún valor en la matriz es mayor
                                        // que el valor almacenado en la
                                        // variable May_Num
                May_Num = M ( I, J ) // Entonces ese es el nuevo valor
                                        // mayor
            Fin_Si                        // Fin de la decisión
        Fin_Para                        // Fin del ciclo interno
    Fin_Para                            // Fin del ciclo externo

    Cont_May = 0                        // Inicializamos el contador en ceros

    Para I = 1 hasta 4                    // Índice que va a referenciar desde la
                                        // primera hasta la cuarta fila
        Para J = 1 hasta 3                // Índice que va a referenciar desde la
                                        // primera hasta la tercera fila
            Si M ( I, J ) = May_Num // Si algún valor en la matriz es igual al
                                        // número que se encontró como mayor
                Cont_May = Cont_May + 1 // entonces cuéntelo
            Fin_Si                        // Fin de la decisión
        Fin_Para                        // Fin del ciclo interno
    Fin_Para                            // Fin del ciclo externo

                                        // Mostrar la información solicitada
    Escriba "El número mayor es ", May_Num. " y se encuentra ", Cont_May, " veces "

Fin

```

Basado en el enunciado no era obligatorio mostrar cuál era el número mayor que se había encontrado pero teniéndolo a la mano no está de mas mostrarlo. Lo que sí es importante mostrar

en pantalla, dado que es la información que se solicita en el enunciado, es la cantidad de veces que está el número mayor en esta matriz.

Prueba de Escritorio

Como en todos los algoritmos lo primero que hacemos es declarar en memoria las variables que vamos a necesitar.

Programa Cuenta_Mayor

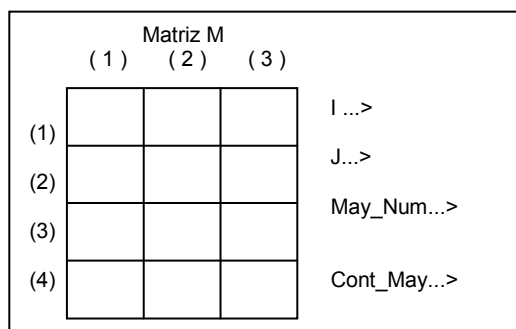
Variables

Entero :M (4, 3),
I,
J,
May_Num,
Cont_May

PANTALLA



MEMORIA



Solicitamos 12 números enteros y los vamos leyendo valiéndonos de dos ciclos anidados que nos permiten recorrer, por filas, la matriz al tiempo que vamos almacenando cada dato en cada una de las posiciones de la matriz. Este ciclo involucra una variable que nos permite referenciar las filas y otra que nos permite referenciar las columnas. Como la variable que referenciará las filas es el índice del ciclo externo y la variable que referenciará las columnas es el índice del ciclo interno entonces nuestro recorrido llenará inicialmente toda la primera fila, luego llenará toda la segunda fila, luego llenará toda la tercera fila y por último llenará toda la cuarta fila con los datos que progresivamente vaya digitando el usuario.

Como en los ejercicios anteriores se ha hecho una prueba de escritorio al conjunto de instrucciones que nos permiten llenar de datos enteros la matriz, vamos a “saltarnos” esta parte del algoritmo debido a que ya tenemos la certeza de que sí funciona y de que sí nos permite llenar la matriz con datos.

Inicio

Escriba “ Digite 12 números enteros “

```

Para I = 1 hasta 4
    Para J = 1 hasta 3
        Lea M ( I, J )
    Fin_Para
Fin_Para

```

PANTALLA

```

Digite 12 números
enteros
5
23
12
14
21
5
23
22
21
23
12

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2 3 4
J...> 1 2 3
May_Num...>
Cont_May...>

Continuando con nuestro algoritmo inicializamos la variable May_Num con un número muy pequeño (para este caso hemos escogido el número -30000). Luego de esto vamos a generar, valiéndonos de un ciclo externo, los números del 1 al 4 que se irán almacenando progresivamente en la variable I y que servirá para referenciar la posición de las filas. Dentro de este ciclo generaremos otro ciclo que utilizará a la variable J como índice, la cual tomará valores entre 1 y 3 y servirá para referenciar las columnas dentro de cada fila. Dentro del ciclo interno preguntaremos si cada uno de los datos almacenados en la matriz M en la fila I columna J es mayor que el valor que esté almacenado en la variable May_Num. Si es así entonces dicho valor deberá quedar almacenado en la variable May_Num. De cualquier forma después se deberá incrementar en 1 el contenido de J para repetir el proceso.

```
May_Num = -30000
```

```

Para I = 1 hasta 4
    Para J = 1 hasta 3
        Si M ( I, J ) > May_Num
            May_Num = M ( I, J )
        Fin_Si
    Fin_Para
Fin_Para

```

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1
J...> 1
May_Num.->-30000
Cont_May...>

Cuando la variable I vale 1 y la variable J vale 1 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(1, 1) > May_Num$ o sea $Si\ 5 > -30000$. Como es Verdadero entonces ejecutamos la orden $May_Num = M(I, J)$ que se convierte en $May_Num = 5$. Incrementamos en 1 el valor almacenado en la variable J y como su nuevo valor sigue siendo menor o igual que 3 entonces volvemos a evaluar la decisión.

PANTALLA

.

.

.

23

12

10

MEMORIA

Matriz M

(1)

(2)

(3)

(1)	5	23	12	I...> 1
(2)	14	21	5	J...> 4 2
(3)	23	22	21	May_Num.->30000 5
(4)	23	12	10	Cont_May...>

Cuando la variable I vale 1 y la variable J vale 2 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(1, 2) > May_Num$ o sea $Si\ 23 > 5$. Como es Verdadero entonces ejecutamos la orden $May_Num = M(I, J)$ que se convierte en $May_Num = 23$. Seguidamente incrementamos en 1 el valor almacenado en la variable J y como su nuevo valor sigue siendo menor o igual que 3 entonces volvemos a evaluar la decisión.

PANTALLA

.

.

.

23

12

10

MEMORIA

Matriz M

(1)

(2)

(3)

(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1

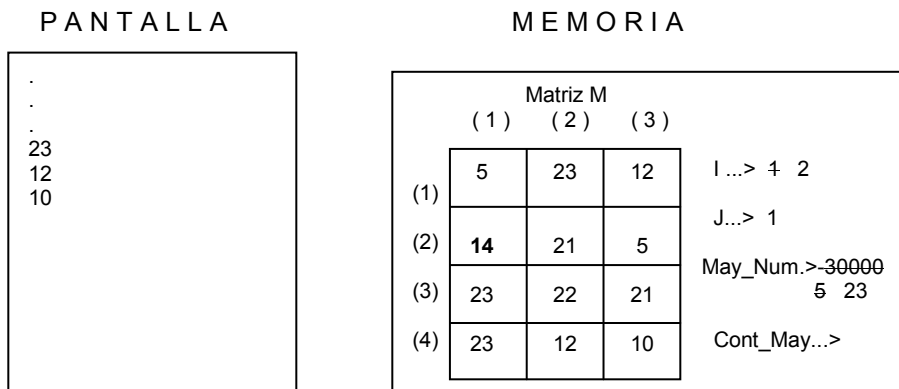
J...> 4 ≠ 3

May_Num.->30000

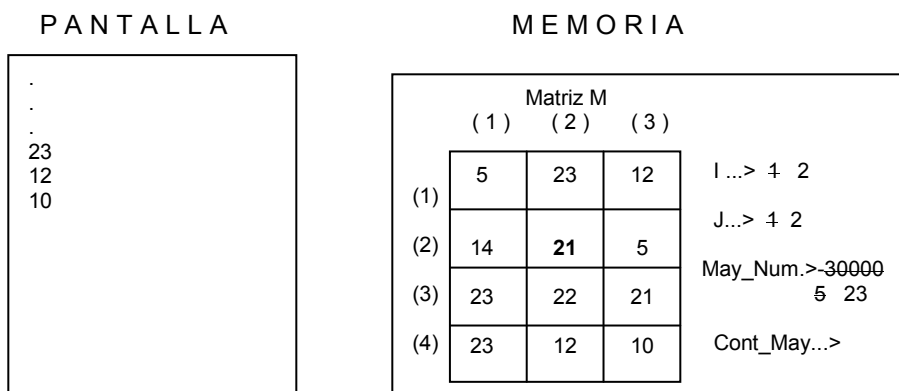
5 23

Cont_May...>

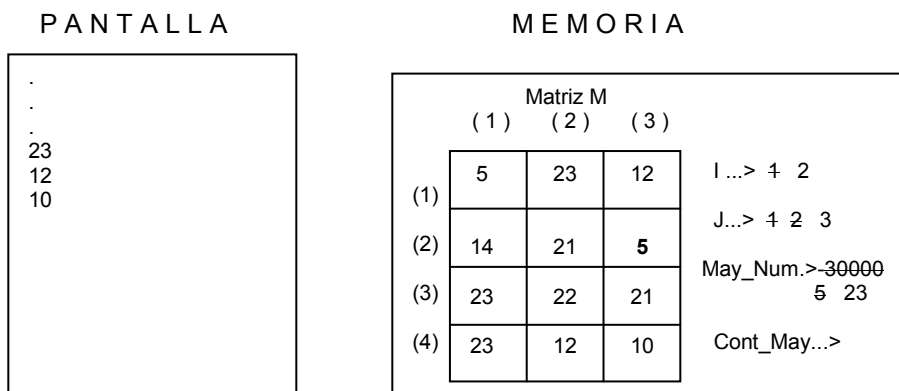
Cuando la variable I vale 1 y la variable J vale 3 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(1, 3) > May_Num$ o sea $Si\ 12 > 23$. Como es Falso entonces nos saltamos la orden de asignación y como ya la variable J tiene el valor 3 (que era su tope) volvemos al ciclo externo e incrementamos en 1 el valor almacenado en la variable I . Como esta variable no ha llegado aún a su tope entonces volvemos a asignarle a la variable J el valor 1 y vamos a generar de nuevo el ciclo interno con valores para J desde 1 hasta 3.



Cuando la variable I vale 2 y la variable J vale 1 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(2, 1) > May_Num$ o sea $Si\ 14 > 23$. Como es Falso entonces incrementamos en 1 el contenido de la variable J.



Cuando la variable I vale 2 y la variable J vale 2 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(2, 2) > May_Num$ o sea $Si\ 21 > 23$. Como es Falso entonces incrementamos en 1 el contenido de la variable J.



Cuando la variable I vale 2 y la variable J vale 3 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(2, 3) > May_Num$ o sea $Si\ 5 > 23$. Como es Falso entonces incrementamos en 1 el contenido de la variable J. En este instante la variable J ha llegado a su tope razón por la

cual nos salimos del ciclo interno y vamos al ciclo externo a incrementar en 1 el contenido de la variable I. Como dicho contenido todavía es menor o igual que 4 entonces volvemos a iniciar el ciclo interno asignándole valores a J desde 1 hasta 3.

PANTALLA

.

.

.

23

12

10

MEMORIA

Matriz M

(1)

(2)

(3)

(1)

(2)

(3)

(4)

5	23	12
14	21	5
23	22	21
23	12	10

I...> 4 2 3

J...> 1

May_Num.>30000

5 23

Cont_May...>

Cuando la variable I vale 3 y la variable J vale 1 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(3, 1) > May_Num$ o sea $Si\ 23 > 23$. Como es Falso entonces incrementamos en 1 el contenido de la variable J.

PANTALLA

.

.

.

23

12

10

MEMORIA

Matriz M

(1)

(2)

(3)

(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 4 2 3

J...> 4 2

May_Num.>-30000

5 23

Cont_May...>

Cuando la variable I vale 3 y la variable J vale 2 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(3, 2) > May_Num$ o sea $Si\ 22 > 23$. Como es Falso entonces incrementamos en 1 el contenido de la variable J.

PANTALLA

.

.

.

23

12

10

MEMORIA

Matriz M

(1)

(2)

(3)

(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 4 2 3

J...> 4 2 3

May_Num.>-30000

5 23

Cont_May...>

Cuando la variable I vale 3 y la variable J vale 3 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(3, 3) > May_Num$ o sea $Si\ 21 > 23$. Como es Falso entonces volvemos al ciclo externo debido a que el índice del ciclo interno ya llegó a su tope. Por tanto incrementamos en 1 el contenido de la variable I y volvemos a generar un ciclo desde 1 hasta 3 para referenciar las posiciones dentro de la cuarta fila.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2 3 4
 J...> 1
 May_Num.>-30000
 5 23
 Cont_May...>

Cuando la variable I vale 4 y la variable J vale 1 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(4, 1) > May_Num$ o sea $Si\ 23 > 23$. Como es Falso entonces volvemos a incrementar en 1 el contenido de la variable J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2 3 4
 J...> 1 2
 May_Num.>-30000
 5 23
 Cont_May...>

Cuando la variable I vale 4 y la variable J vale 2 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(4, 2) > May_Num$ o sea $Si\ 12 > 23$. Como es Falso entonces volvemos a incrementar en 1 el contenido de la variable J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2 3 4
 J...> 1 2 3
 May_Num.>-30000
 5 23
 Cont_May...>

Cuando la variable I vale 4 y la variable J vale 3 entonces la decisión $Si\ M(I, J) > May_Num$ se convierte en $Si\ M(4, 3) > May_Num$ o sea $Si\ 10 > 23$. Como es Falso nos salimos del ciclo interno debido a que la variable J ya llegó a su tope y nos salimos a su ciclo externo debido a que la variable I también ya llegó a su tope.

Continuando con nuestro algoritmo y sabiendo que ya tenemos almacenado en la variable May_Num el mayor valor contenido en la matriz (que es igual al número 23) entonces procedemos a buscar cuántas veces está este valor en la misma matriz. Para ello vamos a inicializar un contador en ceros y de nuevo vamos a recorrer la matriz por filas preguntando en cada uno de los datos contenidos en ella si es igual al mayor valor obtenido.

```

Cont_May = 0
Para I = 1 hasta 4
    Para J = 1 hasta 3
        Si M(I, J) = May_Num
            Cont_May = Cont_May + 1
        Fin_Si
    Fin_Para
Fin_Para

```

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1
 J...> 1
 May_Num.>-30000
 5 23
 Cont_May...> 0

Cuando la variable I valga 1 y J valga 1 la decisión $Si\ M(I, J) = May_Num$ se convertirá en $Si\ M(1, 1) = May_Num$ lo cual significa $Si\ 5 = 23$ lo cual es Falso. Por tanto incrementamos en 1 el valor almacenado en la variable J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1
 J...> 4 2
 May_Num.>-30000
 5 23
 Cont_May...> 0

Cuando la variable I valga 1 y J valga 2 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(1, 2) = \text{May_Num}$ lo cual significa $\text{Si } 23 = 23$ lo cual es Verdadero. Por lo tanto incrementamos en 1 el contenido de la variable Cont_may y volvemos a incrementar en 1 el contenido de la variable J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1
 J...> 1 2 3
 May_Num.>-30000
 5 23
 Cont_May...> 0 1

Cuando la variable I valga 1 y J valga 3 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(1, 3) = \text{May_Num}$ lo cual significa $\text{Si } 12 = 23$ lo cual es Falso. Por tanto nos salimos del ciclo interno debido a que la variable J ya llegó a su tope. Incrementamos de nuevo el contenido de la variable I y volvemos a generar el ciclo interno desde 1 hasta 3 utilizando al variable J como índice.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2
 J...> 1
 May_Num.>-30000
 5 23
 Cont_May...> 0 1

Cuando la variable I valga 2 y J valga 1 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(2, 1) = \text{May_Num}$ lo cual significa $\text{Si } 14 = 23$ lo cual es Falso. Por lo tanto incrementamos en 1 el valor almacenado en la variable J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2
 J...> 1 2
 May_Num.>-30000
 5 23
 Cont_May...> 0 1

Cuando la variable I valga 2 y J valga 2 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(2, 2) = \text{May_Num}$ lo cual significa $\text{Si } 21 = 23$ lo cual es Falso. Por lo tanto incrementamos en 1 el valor almacenado en la variable J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 4 2
 J...> 4 2 3
 May_Num.>-30000
 5 23
 Cont_May...> 0 1

Cuando la variable I valga 2 y J valga 3 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(2, 3) = \text{May_Num}$ lo cual significa $\text{Si } 5 = 23$ lo cual es Falso. Por lo tanto nos salimos del ciclo interno debido a que de nuevo la variable J llegó a su tope. Incrementamos de nuevo el valor almacenado en I y volvemos a entrar al ciclo interno.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 4 2 3
 J...> 1
 May_Num.>-30000
 5 23
 Cont_May...> 0 1

Cuando la variable I valga 3 y J valga 1 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(3, 1) = \text{May_Num}$ lo cual significa $\text{Si } 23 = 23$ lo cual es Verdadero razón por la cual se incrementa en 1 el valor almacenado en Cont_May. Volvemos pues a incrementar en 1 el contenido de J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 4 2 3
 J...> 4 2
 May_Num.>-30000
 5 23
 Cont_May...> 0 4 2

Cuando la variable I valga 3 y J valga 2 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(3, 2) = \text{May_Num}$ lo cual significa $\text{Si } 22 = 23$ lo cual es Falso entonces simplemente incrementamos en 1 el valor almacenado en J.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2 3
 J...> 1 2 3
 May_Num.>-30000
 5 23
 Cont_May...> 0 1 2

Cuando la variable I valga 3 y J valga 3 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(3, 3) = \text{May_Num}$ lo cual significa $\text{Si } 21 = 23$ lo cual es Falso entonces nos salimos del ciclo interno pues J volvió a llegar a su tope. Seguidamente incrementamos en 1 el valor almacenado en I y volvemos a generar el ciclo interno.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2 3 4
 J...> 1
 May_Num.>-30000
 5 23
 Cont_May...> 0 1 2

Cuando la variable I valga 4 y J valga 1 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(4, 1) = \text{May_Num}$ lo cual significa $\text{Si } 23 = 23$ lo cual es Verdadero por lo tanto incrementamos en 1 el valor almacenado en la variable Cont_May.

PANTALLA

```

.
.
.
23
12
10

```

MEMORIA

Matriz M			
	(1)	(2)	(3)
(1)	5	23	12
(2)	14	21	5
(3)	23	22	21
(4)	23	12	10

I...> 1 2 3 4
 J...> 1 2
 May_Num.>-30000
 5 23
 Cont_May.> 0 1 2
 3

Cuando la variable I valga 4 y J valga 2 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(4, 2) = \text{May_Num}$ lo cual significa $\text{Si } 12 = 23$ lo cual es Falso por lo tanto incrementamos en 1 el valor almacenado en J.

PANTALLA

MEMORIA

																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	</
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

Cuando la variable I valga 4 y J valga 3 la decisión $\text{Si } M(I, J) = \text{May_Num}$ se convertirá en $\text{Si } M(4, 3) = \text{May_Num}$ lo cual significa $\text{Si } 10 = 23$ que es Falso por lo tanto nos salimos tanto del ciclo interno como del ciclo externo debido a que ambas variables ya llegaron a sus correspondientes topos.

Finalmente la última orden muestra en pantalla el valor solicitado.

Escriba "El número mayor es ", May_Num. " y se encuentra ", Cont_May, " veces "

PANTALLA

.

.

.

23

12

10

El número mayor es 23 y se encuentra 3 veces

MEMORIA

Matriz M

(1)

(2)

(3)

(1)

(2)

(3)

(4)

5	23	12
14	21	5
23	22	21
23	12	10

I...> 1 2 3 4

J...> 1 2 3

May_Num.>-30000
5 23

Cont_May.> 0 1 2 3

Con lo cual solo quedaría "ejecutar" el fin del algoritmo.

Fin

Vemos pues que realmente de los números leídos, el número 23 es el mayor y está 3 veces en la matriz con lo cual podemos decir que este algoritmo sí cumple con el objetivo planteado inicialmente.

Ejercicios

Algunas posibles soluciones a los siguientes enunciados las puede encontrar en el Libro *Algoritmos* del mismo autor.

Notas Aclaratorias:

- a. En los siguientes enunciados cuando se diga *Leer una matriz $m \times n$ entera* significa leer $m \times n$ datos enteros y almacenarlos en m filas y n columnas para cualquier valor positivo de m y de n .
- b. Cuando el enunciado diga *Posición Exacta* se refiere a la fila y a la columna del dato especificado.

1. Leer una matriz 4×4 entera y determinar en qué fila y en qué columna se encuentra el número mayor.
2. Leer una matriz 4×4 entera y determinar cuántas veces se repita en ella el número mayor.
3. Leer una matriz 3×4 entera y determinar en qué posiciones exactas se encuentran los números pares.
4. Leer una matriz 4×3 entera y determinar en qué posiciones exactas se encuentran los números primos.
5. Leer una matriz 4×3 entera, calcular la suma de los elementos de cada fila y determinar cuál es la fila que tiene la mayor suma.
6. Leer una matriz 4×4 entera y calcular el promedio de los números mayores de cada fila.
7. Leer una matriz 4×4 entera y determinar en qué posiciones están los enteros terminados en 0.
8. Leer una matriz 4×4 entera y determinar cuántos enteros terminados en 0 hay almacenados en ella.
9. Leer una matriz 3×4 entera y determinar cuántos de los números almacenados son primos y terminan en 3.
10. Leer una matriz 5×3 entera y determinar en qué fila está el mayor número primo.

11. Leer una matriz 5x3 entera y determinar en qué columna está el menor número par.
12. Leer una matriz 5x5 entera y determinar en qué fila está el mayor número terminado en 6.
13. Leer una matriz 5x3 entera y determinar en qué columna está el mayor número que comienza con el dígito 4.
14. Leer una matriz 5x5 entera y determinar cuántos números almacenados en ella tienen mas de 3 dígitos.
15. Leer una matriz 5x4 entera y determinar cuántos números almacenados en ella terminan en 34.
16. Leer una matriz 5x4 entera y determinar cuántos números almacenados en ella tienen un solo dígito.
17. Leer una matriz 5x4 entera y determinar cuántos múltiplos de 5 hay almacenados en ella.
18. Leer una matriz 5x5 entera y determinar en qué posición exacta se encuentra el mayor múltiplo de 8.
19. Leer dos matrices 4x5 entera y determinar si sus contenidos son exactamente iguales.
20. Leer dos matrices 4x5 entera, luego leer un entero y determinar si cada uno de los elementos de una de las matrices es igual a cada uno de los elementos de la otra matriz multiplicado por el entero leído.
21. Leer dos matrices 4x5 enteras y determinar cuántos datos tienen en común.
22. Leer dos matrices 4x5 enteras y determinar si el número mayor almacenado en la primera está en la segunda.
23. Leer dos matrices 4x5 enteras y determinar si el número mayor de una de las matrices es igual al número mayor de la otra matriz.
24. Leer dos matrices 4x5 enteras y determinar si el mayor número primo de una de las matrices también se encuentra en la otra matriz.
25. Leer dos matrices 4x5 enteras y determinar si el mayor número primo de una de las matrices es también el mayor número primo de la otra matriz.

26. Leer dos matrices 4x5 enteras y determinar si la cantidad de números pares almacenados en una matriz es igual a la cantidad de números pares almacenados en la otra matriz.
27. Leer dos matrices 4x5 enteras y determinar si la cantidad de números primos almacenados en una matriz es igual a la cantidad de números primos almacenados en la otra matriz.
28. Leer una matriz 4x6 entera y determinar en qué posiciones se encuentran los números cuyo penúltimo dígito sea el 5.
29. Leer una matriz 4x6 entera y determinar si alguno de sus números está repetido al menos 3 veces.
30. Leer una matriz 4x6 entera y determinar cuántas veces está en ella el número menor.
31. Leer una matriz 4x6 entera y determinar en qué posiciones están los menores por fila.
32. Leer una matriz 4x6 entera y determinar en qué posiciones están los menores primos por fila.
33. Leer una matriz 4x6 entera y determinar en qué posiciones están los menores pares por fila.
34. Leer una matriz 4x6 entera y determinar cuántos de los números almacenados en ella pertenecen a los 100 primeros elementos de la serie de Fibonacci.
35. Leer dos matrices 4x6 enteras y determinar cuál es el mayor dato almacenado en ella que pertenezca a la Serie de Fibonacci.
36. Leer dos matrices 4x6 enteras y determinar si el mayor número almacenado en una de ellas que pertenezca a la Serie de Fibonacci es igual al mayor número almacenado en la otra matriz que pertenezca a la Serie de Fibonacci.
37. Leer dos matrices 4x6 enteras y determinar si el número mayor de una matriz se encuentra en la misma posición exacta en la otra matriz.
38. Leer dos matrices 4x6 enteras y determinar si el mayor número primo de una matriz está repetido en la otra matriz.
39. Leer dos matrices 4x6 enteras y determinar si el promedio de las “esquinas” de una matriz es igual al promedio de las “esquinas” de la otra matriz.
40. Leer dos matrices 5x5 enteras y determinar si el promedio entero de los elementos de la diagonal de una matriz es igual al promedio de los elementos de la diagonal de la otra matriz.

41. Leer dos matrices 5x5 enteras y determinar si el promedio entero de todos los elementos que no están en la diagonal de una matriz es igual al promedio entero de todos los elementos que no están en la diagonal de la otra matriz.
42. Leer dos matrices 5x5 enteras y determinar si el promedio entero de los números primos de una matriz se encuentra almacenado en la otra matriz.
43. Leer dos matrices 5x5 enteras y determinar si el promedio entero de los números pares de una matriz es igual al promedio de los números pares de la otra matriz.
44. Leer dos matrices 5x5 enteras y determinar si el promedio entero de los números terminados en 4 de una matriz se encuentra al menos 3 veces en la otra matriz.
45. Leer dos matrices 5x5 enteras y determinar si el promedio entero de los números mayores de cada fila de una matriz es igual al promedio de los números mayores de cada fila de la otra matriz.
46. Leer dos matrices 5x5 enteras y determinar si el promedio entero de los números menores cada fila de una matriz corresponde a alguno de los datos almacenados en las "esquinas" de la otra matriz.
47. Leer dos matrices 5x5 enteras y determinar si el promedio de los mayores números primos por cada fila de una matriz es igual al promedio de los mayores números primos por cada columna de la otra matriz.
48. Leer dos matrices 5x5 entera y determinar si el promedio de los mayores elementos que pertenecen a la serie de Fibonacci de cada fila de una matriz es igual al promedio de los mayores elementos que pertenecen a la serie de Fibonacci de cada fila de la otra matriz.
49. Leer una matriz 3x3 entera y determinar si el promedio de todos los datos almacenados en ella se encuentra también almacenado.
50. Leer una matriz 5x5 y determinar si el promedio de los elementos que se encuentran en su diagonal está almacenado en ella. Mostrar en pantalla en qué posiciones exactas se encuentra dicho dato.