UNIDAD VIII

ARREGLOS. PROPOSICION DIMENSION, CICLOS IMPLICITOS.

INTRODUCCION.

En el análisis y diseño de ingeniería existen situaciones que requieren el empleo de matrices o vectores. En FORTRAN podemos agrupar conjuntos de variables a los que se le llaman arreglos con los que podemos manejar lo que en matemáticas se llama matrices.

OBJETIVOS.

Al terminar esta unidad deberás ser capaz de:

1. Explicar que es una variable con subíndice, la proposición DIMENSION, un arreglo.
2. Manejar fluidamente el DO explícito y el DO implícito para dar datos de arreglos y pedir valores guardados en arreglos.
3. Dado un problema expresado en palabras, hacer con cierta habilidad y fluidez el diagrama de flujo así como la codificación.

PROCEDIMIENTO DE APRENDIZAJE.

Estudiar cuidadosamente el anexo.

Requisito: Antes de presentar el examen de esta unidad, deberás entregar dos programas corridos que contengan el tema. Pueden ser de los ejercicios o de la autoevaluación.

EXAMEN DE AUTOEVALUACION.

1. Dos arreglos de una dimensión llamados A y B contienen cada uno 30 elementos. Escribe el segmento de un programa usando la proposición DO para calcular

$$D = \sum_{i=1}^{30} (A_i - B_i)^2)^{1/2}$$

2. Un arreglo de dos dimensiones denominado (A) tiene 6 filas y 6 columnas. - Los elementos de (A) se perforan en tarjetas de datos uno por tarjeta. Escribe un programa que utilice una nidificación de proposiciones DO (DO dentro de un DO) para ejecutar repetidamente la proposición.

READ (5 , 7) A (I , J)

con el propósito de almacenar el arreglo A.

3. Repite el problema 2, excepto que los elementos del arreglo A) aparecen en las tarjetas de datos en orden de columna.

4. Repite el problema 2, excepto que no se conoce el tamaño del arreglo A una tarjeta inicial de datos listados números enteros, que son el tamaño de la fila y la columna.

5. Para el arreglo A del problema 1, determinar la suma de los elementos que aparecen en la diagonal principal.

6. Para el arreglo A del problema 1, determinar la suma de los elementos interiores de este arreglo.

7. Dados los arreglos A y B

A =

7.5	8.4	6.8	7.5	9.6
12.7	19.6	5.5	10.5	6.6
6.5	11.2	4.2	8.6	14.1
8.3	6.4	14.6	7.5	12.3
7.5	6.4	9.8	6.5	6.7

B =

8.8	9.4	13.5	5.4	9.3
7.8	15.3	18.1	9.8	10.7
6.2	8.6	14.6	10.1	2.6
15.8	9.6	7.6	8.6	9.8
15.8	7.4	9.8	6.4	6.8

Hacer el diagrama de flujo y programa para la computadora para :

a. Leer A y B

b. Escribir A y B

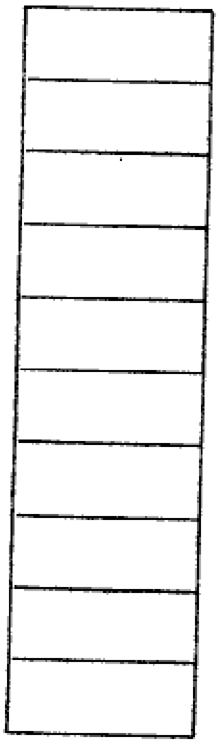
c. Determinar

$$C_{ij} = \left[A_{ij}^{2} - B_{ij}^{2} \right]$$

$i = 1 , 5$

$j = 1 , 5$

X



Observaciones

El arreglo se llama X.

Está formado por 10 variables.

Todas las variables se llaman X.

De las afirmaciones anteriores surgen varias preguntas importantes, la primera sería:

¿En un arreglo, cómo distinguimos una variable de otra?

La respuesta es como sigue:

Después del nombre del arreglo en este caso X se pone entre paréntesis un número, una variable entera, o una expresión que nos dé como resultado un número entero que sea mayor que cero y nos indica el número de la variable a la que nos referimos.

En seguida se presenta una lista de formas válidas para referirse a alguna de las localidades de memoria.

X(5)	La localidad número 5 del arreglo X.
X(L)	La localidad número <u>L</u> del arreglo X. (En este caso <u>L</u> debe ser previamente definida).
X(L + N)	La localidad número L + N del arreglo X. (En este caso L y N deben ser previamente definidos).

En nuestro ejemplo, cuando usamos constante entera (número) debe ser mayor que cero y menor o igual que 10. Y cuando usamos variables enteras o expresiones, el valor de la variable o de la expresión debe quedar comprendido entre 1, 10 .

En nuestro ejemplo anterior se dijo que el arreglo se llama X, la siguiente pregunta es :

¿Qué reglas se deben seguir para asignar nombre a un arreglo?

Como un arreglo es en realidad un conjunto de variables, se van a tener arreglos donde se puedan guardar números reales y arreglos donde se puedan guardar números enteros y las reglas empleadas para asignar nombre a los arreglos son las mismas que aquéllas para asignar nombre a las variables, es decir :

1º Se emplean de 1 a 6 caracteres alfa numéricos. (En la H.P. se permite hasta 15 caracteres).

2º No se permiten blancos en el nombre.

3º El primer caracter es una letra.

4º Si la primera letra es I, J, K, L, M, N, el arreglo sirve para guardar valores enteros.

5º Si el primer caracter es una letra diferente de I, J, K, L, M, N, el arreglo sirve para guardar valores reales.

Del ejemplo anterior observamos que el arreglo X es de una dimensión (en matrices se llama vector), la pregunta siguiente es :

¿Podemos tener arreglos de más de una dimensión?

La respuesta es que sí se pueden tener arreglos de más de una dimensión.

<u>Ejemplo :</u>

Considere un arreglo de 2 dimensiones (4 filas y 5 columnas) que se llame A.

A =

<u>Observaciones:</u>

1º El arreglo A está formado por 20 variables reales.

2º Para referirnos a alguna de las variables se requiere de 2 números, uno que nos indique la fila y [illegible] la columna.

Para indicar dichos números, en forma semejante a los arreglos de una dimensión, después del nombre se abre un paréntesis y se ponen 2 números, el primero de los cuales nos indica la fila y el segundo la columna.

Ejemplo:

A(4, 3) Es el elemento que pertenece al arreglo A que se encuentra en la fila número cuatro y en la columna número 3.

Fila — Columna

EL PRIMER NUMERO NOS INDICA LA FILA Y EL SEGUNDO LA COLUMNA

Otra forma de referirnos a alguna variable de un arreglo, es empleando variables enteras y expresiones cuyo resultado sea un número entero.

Por ejemplo:

A(N1, N2) Es la variable que se pertenece al arreglo A; que se encuentra en la fila N1 y en la columna N2 (donde N1 y N2 debieron ser definidos y son enteros mayores que cero y $N1 \leq 4$, $N2 \leq 5$) es decir $0 < N1 \leq 4$ y $0 < N2 \leq 5$)

A(N+1, N−1) Es la variable [illegible], que se encuentra en la fila N+1 y en la columna N−1 (donde N1 debió ser definido y se debe cumplir que: $0 < N+1 \leq 4$ y $0 < N-1 \leq 5$)

Ejemplos de formas aceptables para referirse a alguna localidad de memoria de un arreglo de 2 dimensiones:

A(I, N)

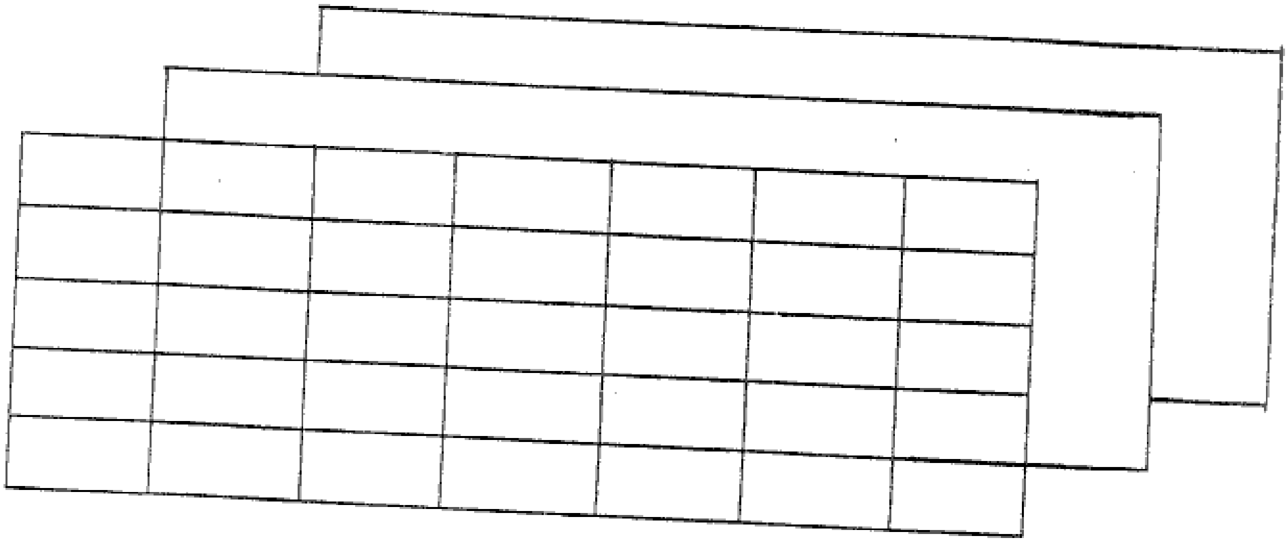
A(I, 2)

A(N, N)

[illegible]N, [illegible]-M)

También se pueden tener arreglos de 3 dimensiones y una aplicación gráfica sería como sigue:

SIP =



El arreglo SIP.

El tamaño del arreglo es 5 filas, 7 columnas y 3 planos, es decir 5x7x3. Es decir son 105 variables reales, para referirnos a alguna de ellas, digamos la que se encuentra en la cuarta fila, 3 columna del segundo plano sería :

SIP(4 , 3 , 2)

Fila

Columna

Plano

y como en los casos anteriores, también se puede emplear para indicar direcciones, variables enteras previamente definidas o expresiones cuyo resultado sea un número entero mayor que cero y que no exceda los límites del tamaño del arreglo.

En todos los ejemplos vistos, se ha hablado del nombre de un arreglo, de la dimensión, y del tamaño, y se dijo que las reglas para asignar nombre a un arreglo son las mismas que para asignarlo a alguna variable, de lo anterior se observa la necesidad de declararle a la computadora cuales van a ser los arreglos que se emplean en un programa, así como también su dimensión y tamaño. Para lo anterior se emplea la proposición DIMENSIØN que es una proposición de especificación, es la primera instrucción del programa cuya forma general es :

```
6
  DIMENSION        A(10) , B(5, 10) , N(15, 2)
```

Proposición

Lista de arreglos separados por comas

En cada arreglo se indica:

a. Si es real o entero por el nombre

b. El tamaño por números enteros mayores que cero

c. La dimensión

Un número	——	1 dimensión
2 números	——	2 dimensiones
3 números	——	3 dimensiones

En caso de 2 dimensiones el primer número se refiere a la fila y el segundo a la columna.

Ejemplo:

```
6
 DIMENSIØN A(10), X1(5, 8), NP(6)
```

Significa que se van a emplear 3 arreglos. Un arreglo real A de 1 dimensión y con 10 lugares.
Un arreglo real X1 de dos dimensiones con 5 filas y 8 columnas y:
Un arreglo entero NP de una dimensión de 6 lugares.

Ejemplo 1.

Supóngase que tenemos una lista con N números y deseamos ordenarlos en forma ascendente.

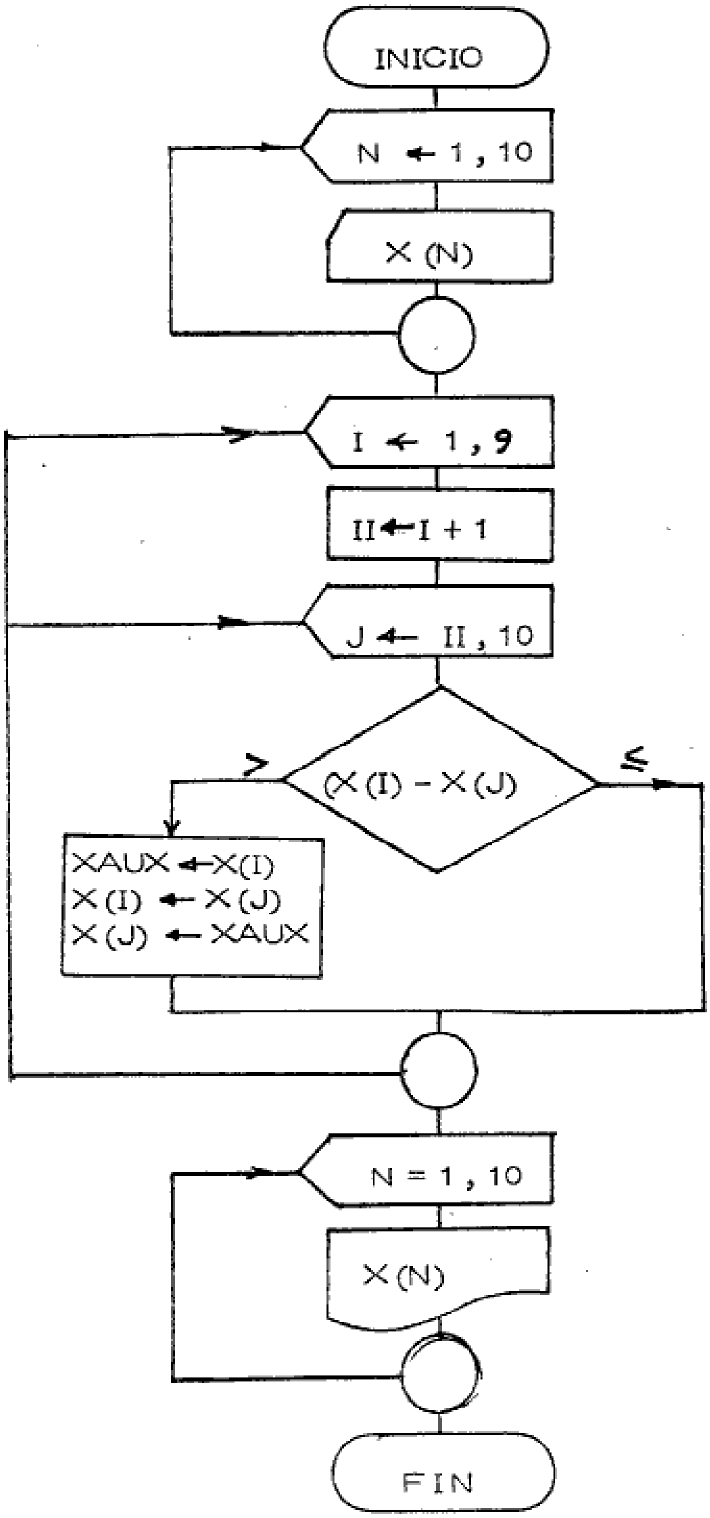
Es decir si introducimos como datos el siguiente arreglo X.

X
6.5
7.9
3.1
6.8
75.4
16.5
17.9
5.4
13.5
1.2

Hacer el diagrama de flujo y programa para la computadora para tener como salida:

X
1.2
3.1
5.4
6.5
6.8
7.9
13.5
16.5
17.9
75.4

Una buena solución sería comparar el primer elemento con el segundo, si el primero, es menor que el segundo, entonces comparamos el primero con el tercero, continuamos hasta encontrar una disminución (si no existe ninguna disminución, entonces los valores ya están ordenados en forma creciente). Al encontrar esta disminución intercambiamos los elementos y regresamos a iniciar el proceso nuevamente como si se tratara de una lista nueva, el proceso lo repetimos hasta que no se encuentra ninguna disminución.



CICLOS IMPLICITOS :

Los elementos de un arreglo se pueden usar independientemente, en las proposi ciones de entrada/salida (E/S), en la misma forma como se usan las demás va riables; sin embargo, pueden también usarse por medio de ciclos DO implíci- tos.

Hay varios puntos sumamente interesantes, relativos al modo en que se maneja cierta lista de variables como las matrices. Por ejemplo, supongamos que tene mos como matriz llamada Q,que tiene 30 términos, y que deseamos asignar va lores numéricos a los primeros 15 de esos términos utilizando la proposición READ. Una de las posibilidades sería escribir el postulado como sigue :

READ (5 , 23) Q (1) , Q (2) , Q (3) , ... para los 15 términos

Esto resultaría bastante tedioso y no sería apropiado en FORTRAN. ¿Se les ocurre alguna otra forma de resolver el problema ?

Si se utiliza la proposición DO, tendremos :

```
      DO 19 K=1 , 15
      READ (5 , 23)Q (K)
   19 CONTINUE
```

En términos estrictos, la proposición CONTINUE no es necesaria en este caso; pero no se causa ninguna dificultad al incluírlo. El número 23 es el FORMAT, correspondiente a la instrucción READ.

Este proceso puede mejorarse :

READ (5 , 23) (Q(K) , K=1 , 15)

Se trata de la instrucción READ con un lazo "DO implícito". La lógica es la misma que en la proposición DO que vimos hace un momento. La computadora examinará la lista de izquierda a derecha, de la manera habitual, y tratará el contenido entre paréntesis como si se encontrara presente el ciclo DO. Pueden incluirse todas las características del proceso iterativo estándar. Por ejemplo, veamos el siguiente postulado READ :

READ (5 , 45) (AD(I) , I=1, M, 2), T , B

En este caso, el contenido entre paréntesis se procesará primero, puesto que la secuencia entre paréntesis se encuentra físicamente antes en la lista. Solamente después de que se hayan leído todos los términos que se encuentra entre paréntesis, se pasará a las otras dos variables de la lista, T y B. En esta proposición, la última variable que se leerá será la B, puesto que, como dijimos, la lista se examina de izquierda a derecha.

El primer término que se lee será AD(1), el siguiente no será AD(2), sino que, de hecho, será AD(3), puesto que I debe incrementarse en etapas de 2.

El proceso se detendrá cuando I sobrepase el valor final, M.

Recordarán que dijimos que es posible "anidar" ciclos DO, y que en esos casos, el DO interno se ejecuta con mayor frecuencia que el externo. En el caso de una lista de variables, es posible un proceso similar. Supongamos que tenemos una matriz llamada ALPHA con ocho hileras y ocho columnas, y que deseamos leer la mitad superior de la matriz, o sea, las primeras cuatro hileras (un total de 32 números). Leeremos los números hilera por hilera. Esto implica que el primer número leído será ALPHA (1 , 1) - primera hilera, primera columna). El siguiente será ALPHA (1 , 2) - primera hilera, segunda columna, y así sucesivamente, hasta ALPHA (1 , 8). El número siguiente que se tomará en consideración será ALPHA (2 , 1) y el término final que se lea deberá ser ALPHA (4 , 8).

Vamos a tratar de escribir un conjunto apropiado de postulados para realizar esta operación, utilizando DO anidados. Tenemos :

```
    DO 1 I=1 , 4
    DO 1 J=1 , 8
1   READ (5 , 2) ALPHA (I , J)
```

En este caso, I tendrá un valor inicial de 1, y mientras conserve ese valor, el DO interior se ejecutará ocho veces para los valores de J que van de 1 a 8, inclusive. A continuación, se incrementará I en 1 y volverá a repetirse el DO interior, etc.

En lugar de expresar los DO anidados, podemos utilizar una modificación de nuestra nueva notación, para obtener :

```
READ (5 , 2) ((ALPHA (I, J), J=1, 8), I=1 , 4)
```

En este caso, el contenido del conjunto interno de paréntesis se procesa primeramente, puesto que corresponde al DO interno. Observa muy cuidadosamente las posiciones de todas las comas deben incluirse siempre, ya que, de otro modo, la proposición carecerá de significado para la computadora.

¿Qué efecto creen ustedes que producirá la proposición siguiente ?

```
READ (5 , 2) ((ALPHA (I , J) , I=1, 4), J=1 , 8)
```

Cuando deseamos leer matrices enteras, como por ejemplo la matriz DATA con 20 elementos. Podemos hacerlo de numerosas formas distintas :

READ (5 , 5) (DATA(I), I=1 , 20)

El FORTRAN nos proporciona un modo muy limpio para resolver esta situación. Lo único que necesitamos decir es :

READ (5 , 5) DATA

O sea, que nos limitamos a proporcionar el nombre de la matriz y esto hará que se lean todos sus términos, en este caso, de acuerdo con el formato 5.

Observen que esta técnica solamente puede utilizarse cuando se vaya a procesar toda la matriz.

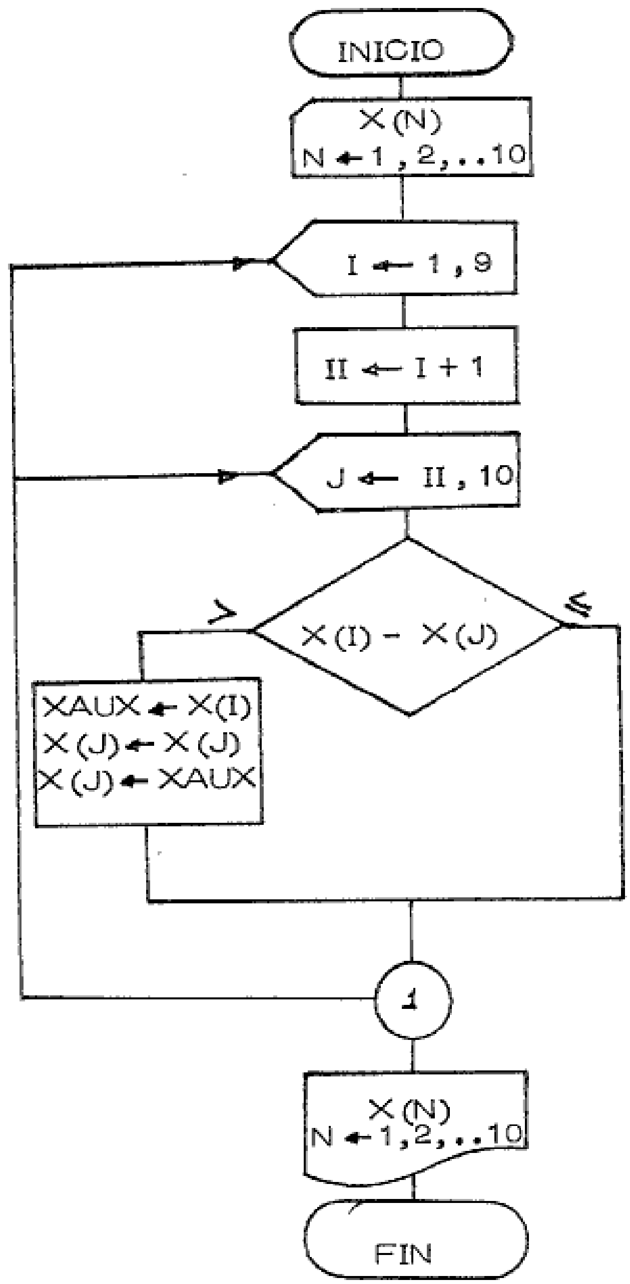
Naturalmente, habrán comprendido ya que, antes de que pueda mencionarse el nombre de una matriz en la proposición READ, tendrá que definírsele a la computadora utilizando la proposición DIMENSION. Sólo las variables con subíndices se dimensionan.

Gran parte de lo que hemos dicho respecto de la proposición READ se explica exactamente en la misma forma a la instrucción WRITE; por ejemplo :

WRITE (6 , 4) A, B(8), (C(I), I=1 , 7)

Ejemplo 2 :

Modificación del ejemplo 1.




```
      DIMENSION X(10)
      READ,(X(N),N=1,10)
      DØ 1 I=1,9
      II=I+1
      DØ 1 J=II,10
      IF (X(I)-X(J))1,1,2
2     XAUX=X(I)
      X(I)=X(J)
      X(J)=XAUX
1     CØNTINUE
      WRITE(6,4)(X(N),N=1,10)
4     FORMAT(25X,'X',/,20X,F10.4
      STØP
      END
```

Ejemplo 3 :

ADICION DE MATRICES.

Se dice que dos matrices A y B son compatibles para la adición cuando son del - mismo orden. Supongamos que la matriz C sea la suma de las matrices A y B - para calcular cualquier elemento de la matriz C, C_{ij}, usamos la fórmula

$$c_{ij} = a_{ij} + b_{ij}$$

Si $A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix}$ $\quad B = \begin{bmatrix} 2 & 4 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

$$C = A + B = \begin{bmatrix} 3 & 6 & 4 \\ -1 & 1 & 1 \\ 6 & 2 & 0 \end{bmatrix}$$

El programa lleva a cabo la operación de adición de dos matrices. Los valores - de los elementos de las matrices A y B se leen por medio de proposiciones de en trada sin formato. El cálculo de la matriz C, igual a la suma de A y B, se lleva a cabo por medio del ciclos DO anidados. Se puede observar que los resultados de la corrida de prueba coinciden con el ejemplo dado anteriormente.

Ejemplo 4 :

Dado un arreglo Y de 5 filas y 6 columnas con los datos que se muestran, hacer el diagrama de flujo y el programa para la computadora para

a. Leer datos.

b. Escribir datos.

c. Determinar el cuadrado de la suma de los elementos de cada fila y el resultado guardarlo en el elemento correspondiente de un arreglo X.

d. Imprimir los resultados.

PAGE 0001

HP32102B.01.01 FORTRAN/3000 (C) HEWLETT-PACKARD CO. 1978 FRI, NOV 9, 1979,

```
$CONTROL USLINIT
C***********************************************************************
C         EDGAR  SOTO  SOLIS      E43J72FT          SPO. C-04
C***********************************************************************
C
C     SUMAR DOS MATRICES (DE LAS MISMAS DIMENSIONES).
C
      DIMENSION JAM(10,10),MAT(10,10),MATSUM(10,10)
      READ(5,*)M,N,((JAM(I,J),J=1,N),I=1,M),((MAT(I,J),J=1,N),I=1,M)
      DO 1 I=1,M
      DO 1 J=1,N
    1 MATSUM(I,J)=JAM(I,J)+MAT(I,J)
      WRITE(6,2)((JAM(I,J),J=1,N),(MAT(I,J),J=1,N),I=1,M),((MATSUM(I,J),
     3J=1,N),I=1,M)
    2 FORMAT(56X,'SUMAR LAS DOS MATRICES :',// ,8(46X,5I3,10X,5I3,/),///
     *,56X,'SUMA DE LAS MATRICES',//,8(55X,5I3,/))
      STOP
      END
```

PROGRAM UNIT
MAIN' COMPILED

SUMAR LAS DOS MATRICES :

2	12	3	5	17
61	7	9	11	15
25	2	4	8	25
17	17	19	21	3
4	2	54	24	8
36	1	9	5	7
0	9	5	15	7
6	7	1	1	1

12	2	17	25	9
8	6	8	5	19
41	43	7	13	15
11	2	3	7	9
20	7	8	1	2
1	9	8	19	8
5	4	12	6	33
43	25	6	7	8

SUMA DE LAS MATRICES

14	14	20	30	26
69	13	17	16	34
66	45	11	21	40
28	19	22	28	12
24	9	62	25	10
37	10	17	24	15
5	13	17	21	40
49	32	7	8	9

Y =

6.3	- 1.3	2.8	4.5	10.6	7.5
16.7	5.4	16.8	7.5	14.1	1.0
7.5	2.8	4.6	7.5	- 8.1	6.5
1.5	6.8	9.4	15.6	7.6	8.4
6.3	2.5	15.4	3.8	9.6	7.8

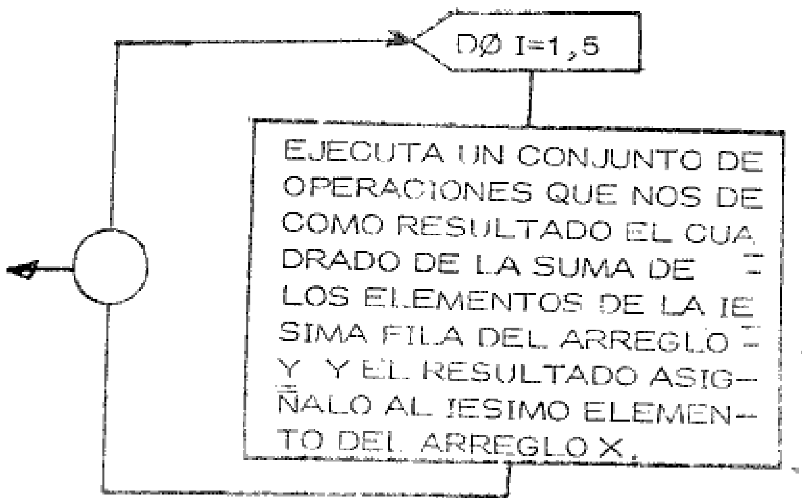
X =

El problema consiste en sumar los elementos de cada fila, el resultado elevarlo al cuadrado y guardarlo en el lugar correspondiente del arreglo X.

<u>Solución</u>:

La solución del problema se puede plantear como que se debe considerar cada una de las filas del arreglo Y y se debe ejecutar un conjunto de operaciones que nos debe dar como resultado el cuadrado de la suma.

Para el diagrama de flujo se tiene:



Nuestro problema ahora consiste en resolver lo expresado en el cuadro anterior, y en este caso es efectuar una sumatoria y el resultado elevarlo al cuadrado y asignarlo al elemento correspondiente de X.

Para realizar la sumatoria se debe iniciar el valor de la variable donde se va a guardar el resultado con cero, después por medio de un postulado DO efectuar la suma de 6 elementos. Y finalmente elevar al cuadrado el resultado de la suma, este conjunto de actividades corresponde a las que se deben ejecutar para obtener lo que se indica en el cuadro, es decir es el conjunto de actividades que se van a ejecutar 5 veces. La variable donde se va a guardar el resultado es el elemento correspondiente del arreglo X, así si nos encontramos en la fila 1 del arreglo Y, la variable será X(1), si nos encontramos en la iésima fila del arreglo Y, la variable será X(I).

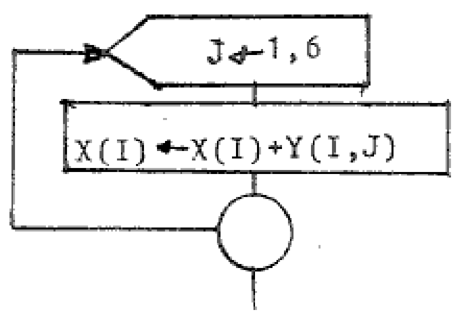
La instrucción es :

X(I) = 0.0

Para efectuar la suma, ya se mencionó el uso del postulado DO para el cual se requiere un contador diferente de I (I nos controla la fila en que nos encontramos).

Se puede usar J (nos controla S el elemento que se está sumando y varía de 1 a 6 ya que hay 6 columnas).

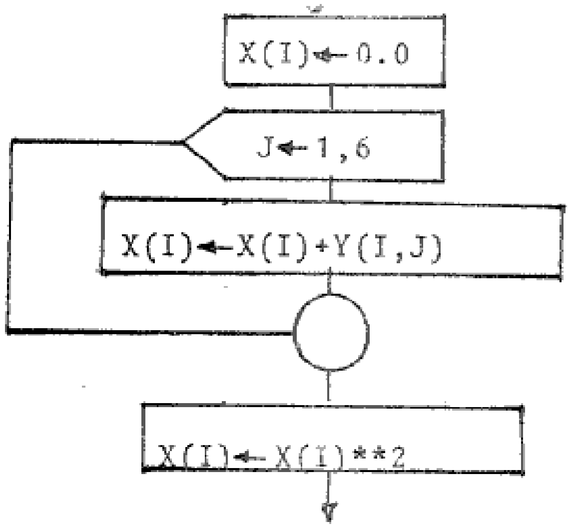
Y la instrucción será :



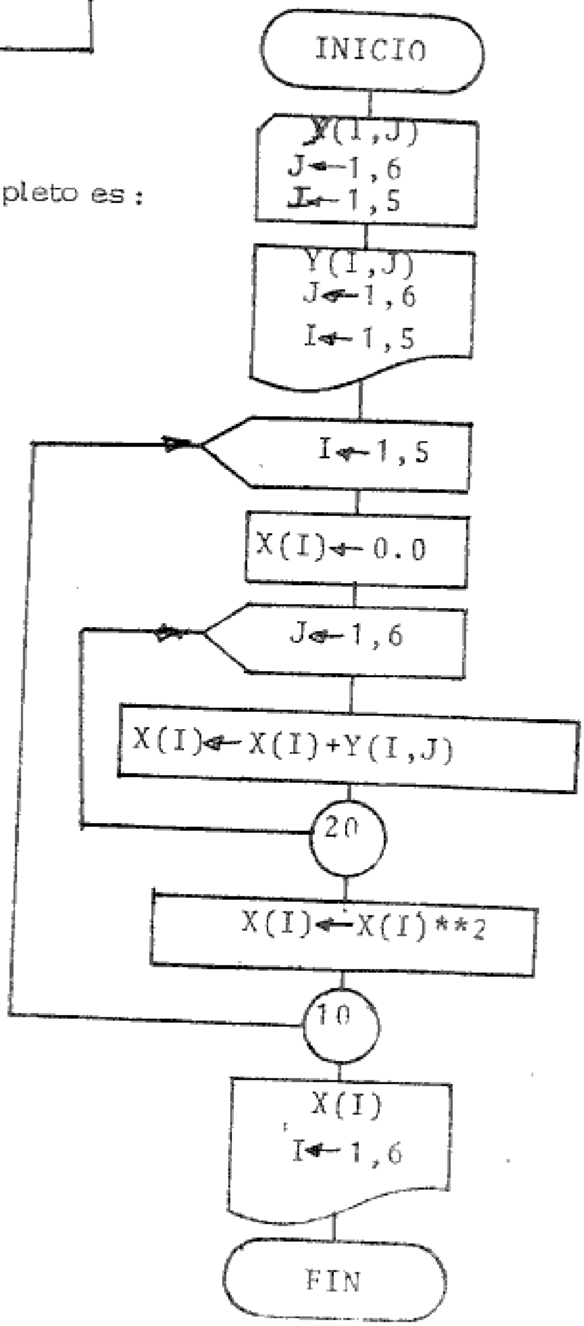
Al terminar el DO anterior, se debe elevar al cuadrado el resultado de sumar y guardarlo en X(I), lo anterior se obtiene con la siguiente instrucción :

X(I) ← X(I)**2

Así el conjunto de instrucciones indicadas en el cuadro son :



Y el diagrama de flujo completo es :



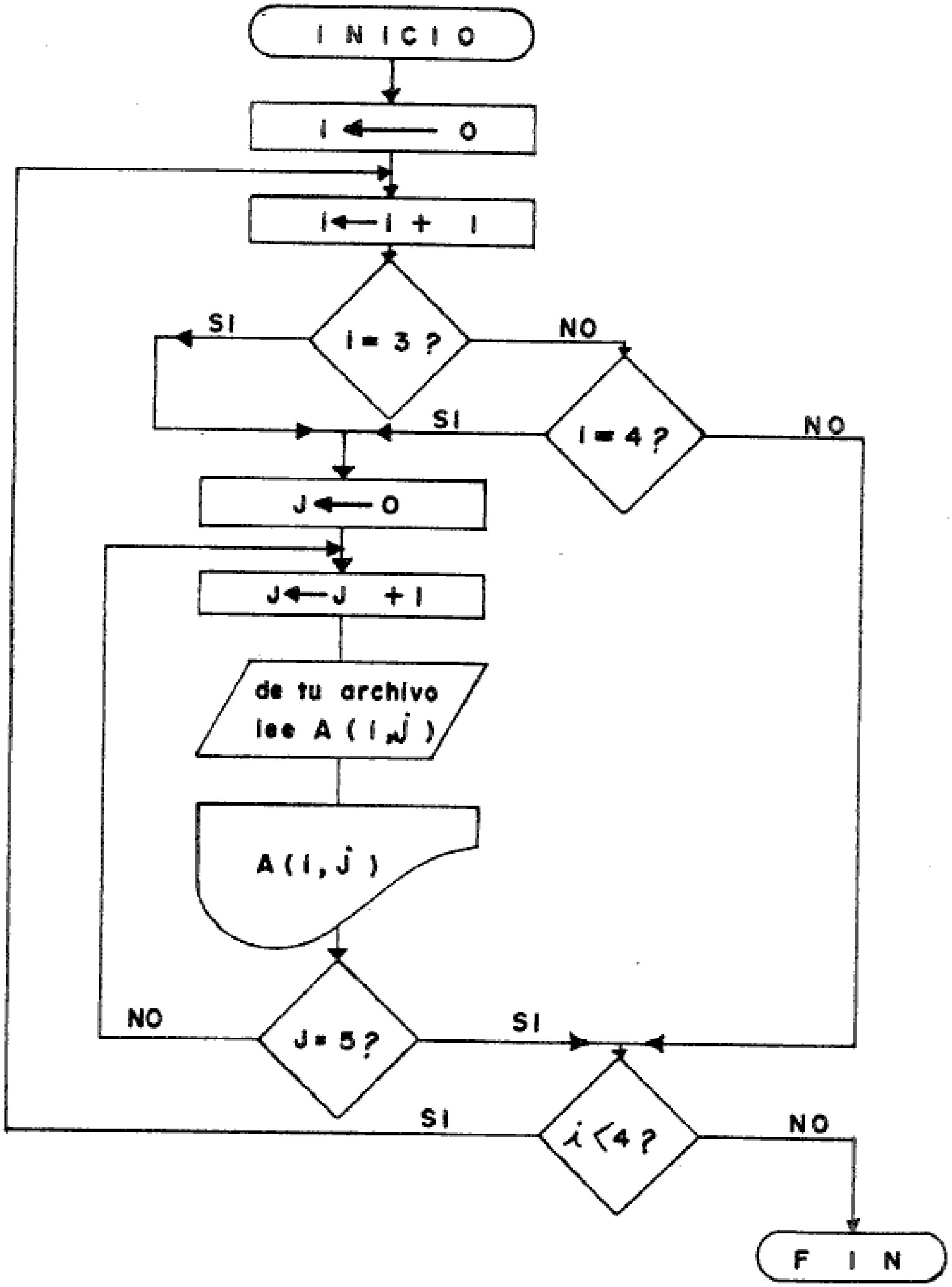
La codificación será :

```
      TARJETAS DE CØNTRØL
      DIMENSIØN Y(5,6),X(5)
C INSTRUCCIØN PARA LEER LØS DATØS PØR FILAS
      READ(5,8)((Y(I,J),J=1,6),I=1,5)
    8 FØRMAT(10F8.3)
C INSTRUCCIØN PARA ESCRIBIR LØS DATØS PØR FILAS ESCRIBIENDØ 6
C VALØRES EN CADA LINEA
      WRITE(6,9)((Y(I,J),J=1,6),I=1,5)
    9 FØRMAT(6(3X,F8.3))
C INSTRUCCIØNES PARA DETERMINAR LØS VALØRES DEL ARREGLØ X
      DØ 10 I=1,5
      X(I)=0.0
      DØ 20 J=1,6
      X(I)=X(I)+Y(I,J)
   20 CØNTINUE
      X(I)=X(I)**2
   10 CØNTINUE
C INSTRUCCIØN PARA ESCRIBIR LØS VALØRES DEL ARREGLØ X
      WRITE(6,12)(X(I),I=1,6)
   12 FØRMAT(5X,F15.6)
      STOP
      END
      TARJETAS DE CØNTRØL
 6.3 ,  -1.3 ,  2.8 ,  4.5 ,  10.6 ,  7.5 ,  16.7 ,  5.4 ,  16.8 ,  7.5
14.1     1.0    7.5    2.8     4.6    7.5    -8.1    6.5     1.5    6.8
 9.4    15.6    7.6    8.4     6.3    2.5    15.4    3.8     9.6    7.8
```

Supongamos que tenemos que analizar el nivel de inventario, existencias de seguridad, etc. de los diferentes artículos en una fábrica.

Artículo	Nivel de Inv.	Inv. de Seguri-dad	$ Precio	días tiempo de entrega	Demanda Semanal
1	100	20	5.20	3	83
2	250	50	2.50	5	250
3	36	5	40	1	40
4	587	50	12.0	4	600
5	187	15	38	2	300

Supongamos que la computadora tiene en su archivo todos los valores de la tabla anterior y en este momento necesito saber todos los datos que me pueda proporcionar de los artículos 3 y 4. ¿Cómo estaría hecho el diagrama de flujo?



Análisis del diagrama de flujo: Aquí encontramos dos iteraciones y están anidadas.

La iteración A está dado por:

1. i ← 0 Asigna a i un valor de cero

2. i ← i + 1 Aumenta a i en uno

3. i < 4? Es i menor que cuatro? sí: ve a incrementar a i en uno no: termina.

La iteración B está dada por:

1. J ← 0 Asigna a J un valor cero

2. J ← J + 1 Aumenta a J en uno

3. J = 5? Es J igual con cinco? si pregunta si i < 4, no incrementa en uno el valor de J.

El objetivo del diagrama es obtener las filas 3 y 4 de la tabla 1, probémoslo

PASOS	ITER A		ITER B		PROCESOS COMPLEMENTARIOS		
	i	i<4?	J	J=5?	i = 3?	i = 4?	A(i, J)
1	Ø, 1	[illegible]			NO	NO	
2	2				NO	NO	
3	3		Ø, 1	NO	SI		36
4			2	NO			5
5			3	NO			40
6			4	NO			1
7			5	SI			40
8	4		Ø, 1		NO	SI	587
9			2	NO			50

PASOS	ITER A		ITER B		PROCESOS COMPLEMENTARIOS		
	i	i<4?	J	J=5?	i = 3?	i = 4?	A(i , J)
10			3	NO			12.0
11			4	NO			4
12		SI	5	SI			600
FIN							

Ejercicio:

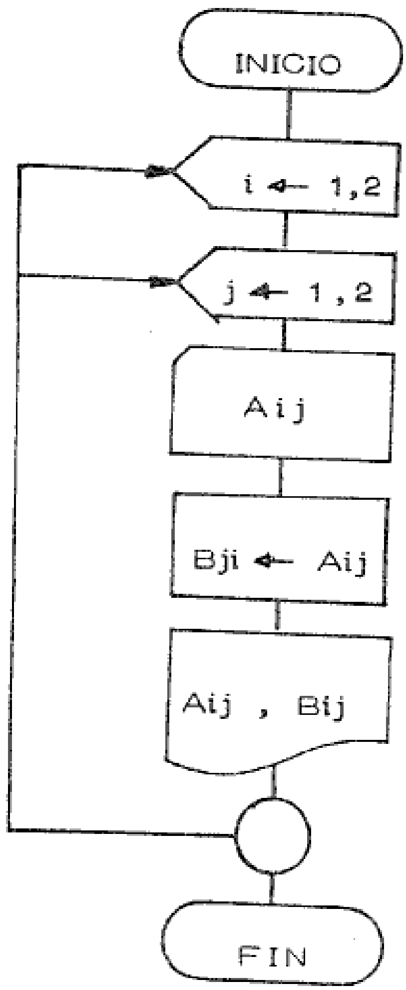
Desarrolla un diagrama de flujo que lea y escriba toda la tabla 1. Pruebalo para ver si funciona. Desarrolla el programa y pruebalo en la computadora.

EJERCICIOS.

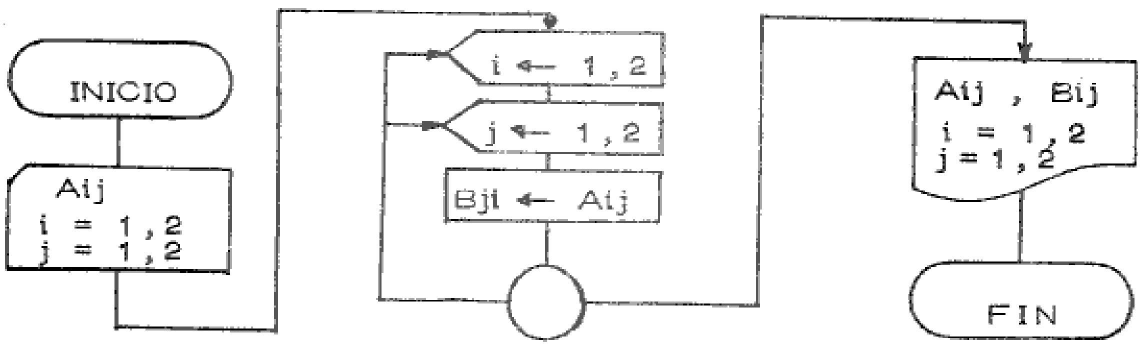
1. Codificar los siguientes diagramas de flujo

a. Dada una matriz A obtener la matriz transpuesta B.

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$



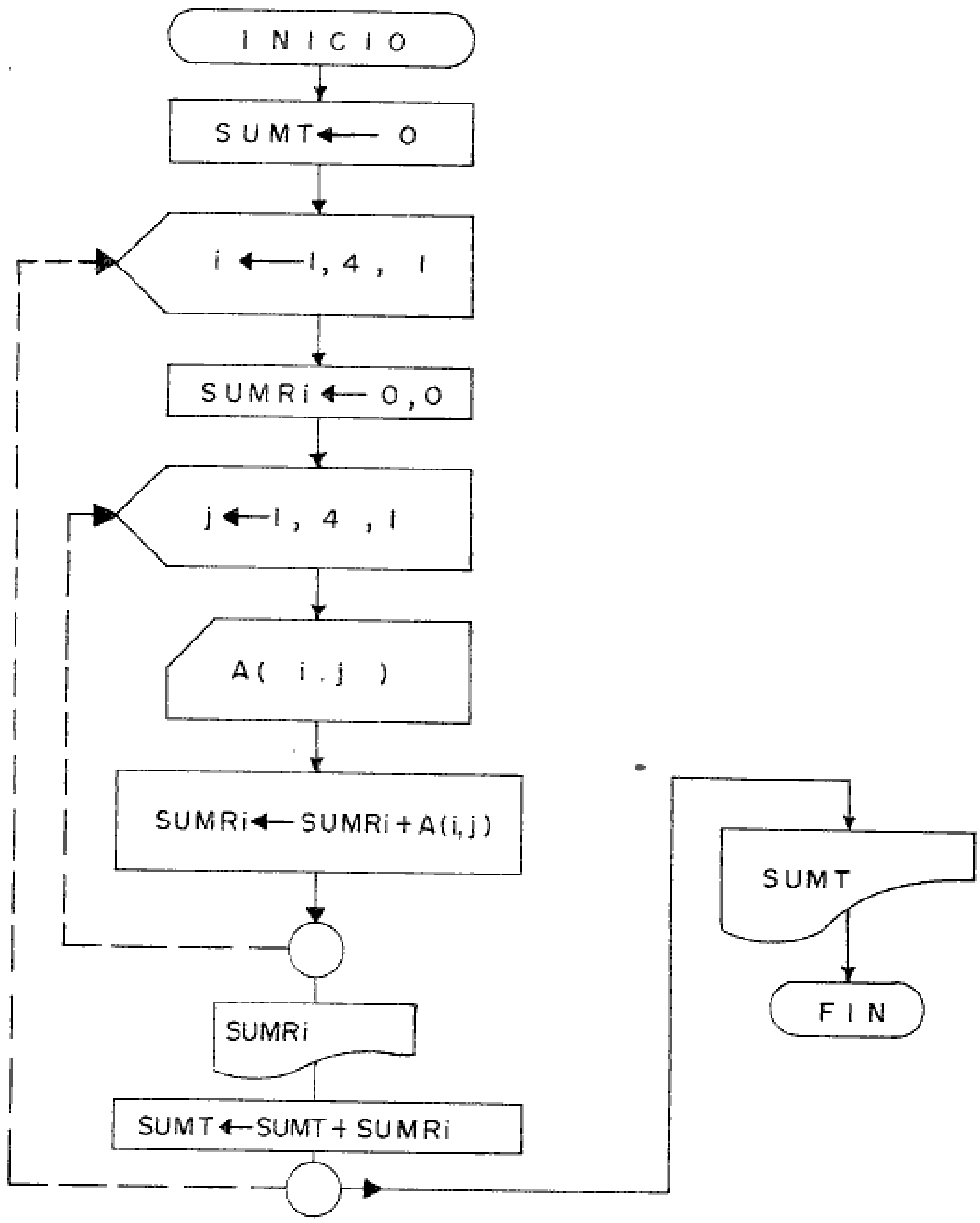
b. Si se le dan la lectura de la matriz antes, el diagrama de flujo se puede escribir como sigue:



c. Diagrama de flujo para obtener la suma de los elementos comprendidos en el marco y la suma de cada renglón dentro de él, de acuerdo al siguiente arreglo.

4.0	3.8	7.5	10.3	7.2
3.4	8.7	3.6	3.4	3.7
5.6	7.8	13.0	4.6	4.6
3.7	8.7	9.7	3.6	5.7
6.7	2.8	7.6	3.1	8.1

d. Mejora este diagrama y codificalo



2. Escribir el diagrama de flujo y programa para cada uno de los ejercicios :

a. Sumar los renglones de una matriz A y encontrar el total de las sumatorias

$$A = \begin{bmatrix} 1 & 5 & 6 \\ 7 & 8 & -6 \\ 4 & 9 & 8 \end{bmatrix} \qquad S = \begin{bmatrix} 12 \\ 9 \\ 21 \end{bmatrix} \qquad S_T = 42$$

b. De la misma matriz calcula la suma de cada columna y obtener el total de las sumatorias.

c. Un tablero de damas en una matriz de 8 renglones y 8 columnas, cuyos elementos son Kij. Un 1 en Kij representa la presencia de una ficha, mientras que un 0 en Kij representa la ausencia de una ficha. Examina el tablero entero y guarda en N el número de fichas del tablero. Suponer que los elementos de la matriz K sólo contienen <u>1</u> y <u>0</u>.

d. Elaborar una tabla de valores de

$$f(x) = e^x + x^2 - 12$$

para

x = 0.0, 0.2, 0.4, ..., hasta que f(x) sea positiva.

3. El cálculo del valor eficaz, valor medio cuadrático (RMS), de una corriente formada por varias corrientes sinusoidales de frecuencias diferentes es necesario en varias ocasiones. El valor eficaz (RMS) se da por medio de la siguiente fórmula

$$I_e = \sqrt{I_{e1}^2 + I_{e2}^2 + \ldots\ldots + I_{en}^2}$$

Escriba un programa que calcule el valor eficaz (RMS) de una corriente que pueda contener hasta 10 componentes. Asigna la variable con subíndice X(I) como la representación de las componentes.

Datos para usar como ejemplo

Grupo	X(1)	X(2)	X(3)	X(4) ....
1	2.0	0.9	0.5	
2	3.7	2.8	1.4	0.6

$$I_e = \sqrt{2^2 + 0.9^2 + 0.5^2} = \sqrt{5.06} = 2.25 \text{ A, amp, RMS}$$

4. Escriba un programa que encuentre el elemento de mayor valor en una matriz de NHIL hileras y MCOL columnas sujeta a la restricción de NHIL $\leqslant$ 10 y MCOL $\leqslant$ 10. El programa debe leer la matriz, escribirla y luego imprimir el elemento de mayor valor.

5. Escriba un programa que calcule el promedio, la varianza y la desviación normal de un conjunto de datos denotados por X(I). Las fórmulas requeridas son:

$$\text{Promedio} = \frac{\sum_{1}^{N} X(I)}{N}$$

$$\text{Varianza} = \frac{\sum_{1}^{N} (X(I) - \text{Promedio})^2}{N}$$

$$\text{Desviación normal} = (\text{varianza})^{1/2}$$