

EL CICLO DO.

INTRODUCCION.

Una de las más importantes características de la computadora es la habilidad para efectuar cálculos iterativos, esto es, repetir el conjunto de una serie de operaciones, varias veces, tantos como el programador lo desee. En esta unidad se enfatiza el uso del ciclo DO que es un procedimiento suministrado por FORTRAN para dicho propósito.

OBJETIVOS.

Al terminar esta unidad debes de ser capaz de :

1. Explicar con tus propias palabras en que consiste la proposición DO.
2. Conocer y manejar fluidamente los elementos que integran la instrucción DO.
3. Conocer y manejar fluidamente las reglas de la proposición DO.
4. Conocer y manejar fluidamente la instrucción CONTINUE.
5. Dado un diagrama de flujo con símbolo de iteración, hacer con cierta facilidad el diagrama de flujo correspondiente.
6. Dado un listado de un programa que contenga la proposición DO, hacer con cierta facilidad el diagrama de flujo correspondiente.
7. Resolver problemas empleando la proposición DO.

PROCEDIMIENTO DE APRENDIZAJE.

1. Estudiar el anexo de la unidad.
2. Si tienes dudas, acude a asesoría o consulta cualquier libro de FORTRAN que contenga el tema.
3. Consulta el manual FORTRAN/3000 II, P.

Requisito :

Antes de presentar el examen de esta unidad, deberás entregar un programa corrido por computadora en el que se emplee la proposición DØ.

Se sugieren algunos de los vistos en esta unidad o de la unidad V.

EXAMEN DE AUTOEVALUACION.

Podrás solicitar el examen de esta unidad cuando seas capaz de resolver las siguientes preguntas :

1. Señalar si es F ó V.

- a. En cualquier evento se puede usar una variable conocida como índice del ciclo de iteración con el propósito de contar cuántas veces se han efectuado las actividades usadas en el ciclo.

F

V

- b. Cuando se usa el símbolo de iteración, un número encerrado en un círculo nos indica el límite de la iteración.

F

V

- c. Las actividades centradas de un ciclo de iteración pueden tener otro ciclo de iteración como componente aunque éste no termine dentro del primer ciclo.

F

V

2. Explica con tus propias palabras las siguientes proposiciones :

$$DØ 20 I = 1, 11, 3$$

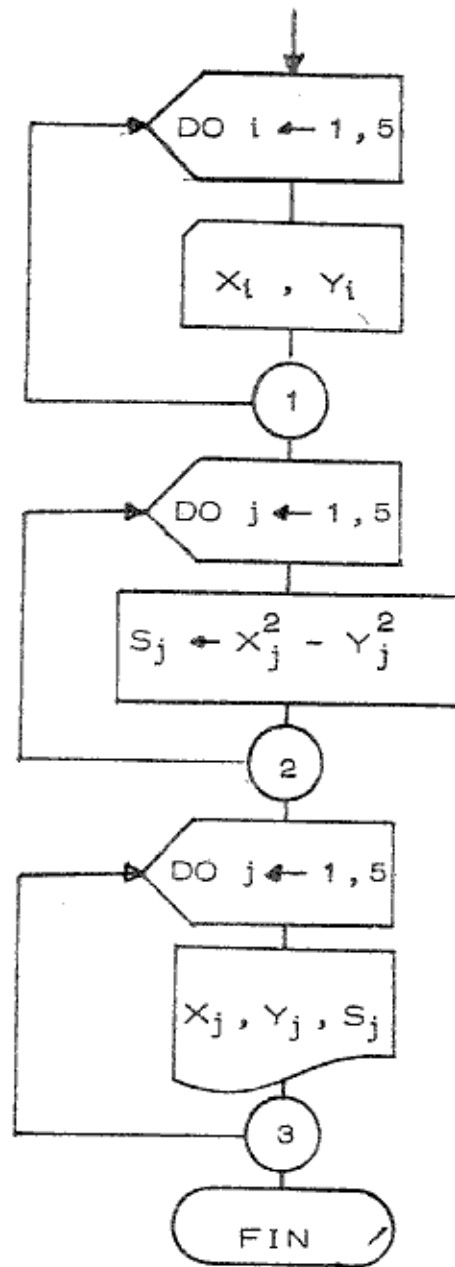
$$DØ 10 J1 = 1, 20$$

$$DØ 30 I1 = N1, N2, N3$$

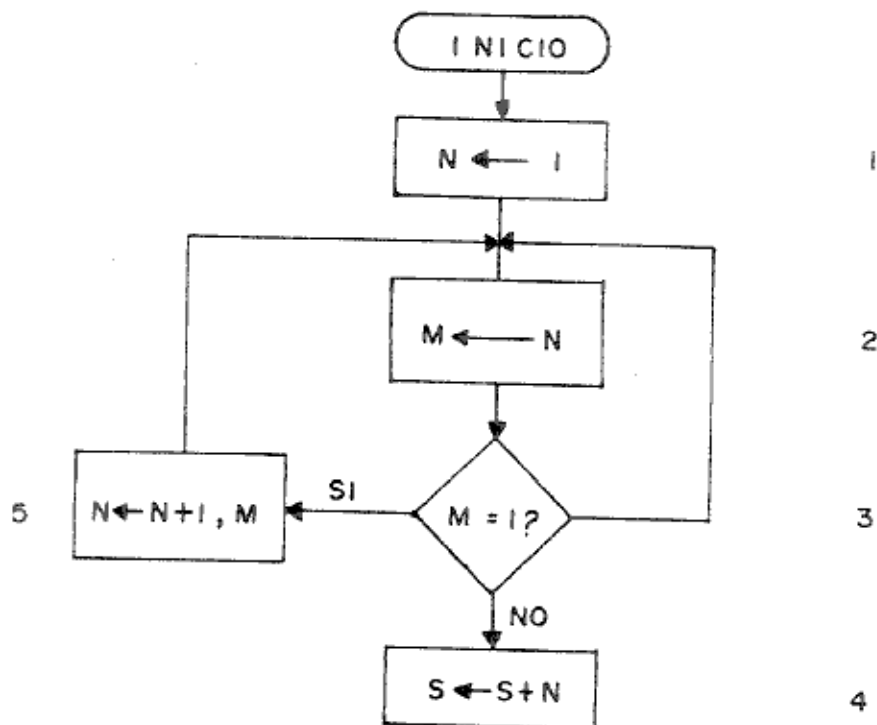
3. Explica claramente para que se usa la proposición CONTINUE.

4. Menciona cada una de las reglas de la proposición DØ.

5. En los siguientes diagramas de flujo, hacer la codificación correspondiente.



6. Se desea encontrar la suma de los números nones que existen entre 1 y 200. ¿El siguiente diagrama de flujo nos servirá para este propósito?. Si no es así, menciona sus fallas.



7. Desarrolla el diagrama de flujo y programa que resuelva c/u de los incisos:
- Dados N números positivos y negativos, contar cuántos positivos hay y cuántos negativos, comprobando que cumpla con las 5 características de un buen algoritmo.
 - Encontrar un número menor de un conjunto de "n" números reales y la posición que ocupa este número en el conjunto.
 - Escribe un programa FORTRAN que calcule e imprima la suma de m enteros consecutivos empezando con un número n entero positivo cualquiera.
 - Escribe un programa que produzca una tabla de conversión de temperaturas para convertir valores enteros de grados Fahrenheit a sus valores equivalentes en grados centígrados, en grados Kelvin y en grados Rankine. Las ecuaciones de conversión son:

$$C = \frac{5(F - 32)}{9}$$

$$R = F + 459.67$$

$$K = C + 273.15$$

donde

F = temperatura en grados Fahrenheit

C = temperatura en grados Centígrados

R = temperatura en grados Rankine

K = temperatura en grados Kelvin

Hacer que el programa escriba una línea de encabezamiento con nombres para cada una de las columnas de la tabla.

- e. Escribe un programa FORTRAN para calcular e imprimir una tabla de valores $\tan \theta$ para valores enteros positivos de θ en grados, comenzando en un valor θ_1 , con incremento θ_3 y terminando con un valor θ_2 .

MATERIAL ANEXO.

PROCESOS ITERATIVOS :

Un algoritmo es un procedimiento para resolver un problema. Para ser efectivo un algoritmo debe ser verdadero, debe tener en cuenta todas las alternativas que se pueden presentar y debe producir una respuesta en un número finito de pasos.

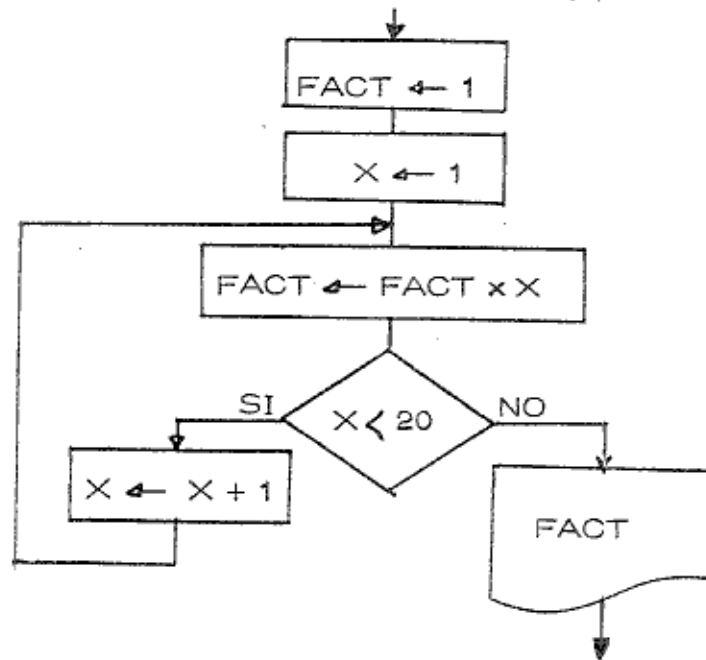
En muchos algoritmos es necesario llevar a cabo el mismo cálculo varias veces, usando cada vez un valor diferente para una variable. La ejecución de este tipo de procedimiento iterativo o recursivo en un computador se llama repetición del ciclo.

El siguiente segmento de programa se puede usar, por ejemplo, para calcular $20!$.

```

FACT ← 1
X ← 1
1  FACT = FACT * X
   IF(20-X)2,2,3
3  X ← X + 1
   GO TO 1
2  WRITE (6, *)FACT

```



- Paso 1 inicia los valores de FACT y X
- Paso 2 calcula (principio del ciclo)
- Paso 3 compara el valor actual de X (es decir, el valor en este ciclo). Observa que si X es igual a 20. Entonces se imprimir el resultado.

- Paso 4 aumenta el valor de X en 1
- Paso 5 devuelve control al principio del ciclo para permitir la continuación del cálculo
- Paso 6 es una proposición de salida que se usa una vez que se haya completado el cálculo (WRITE...)

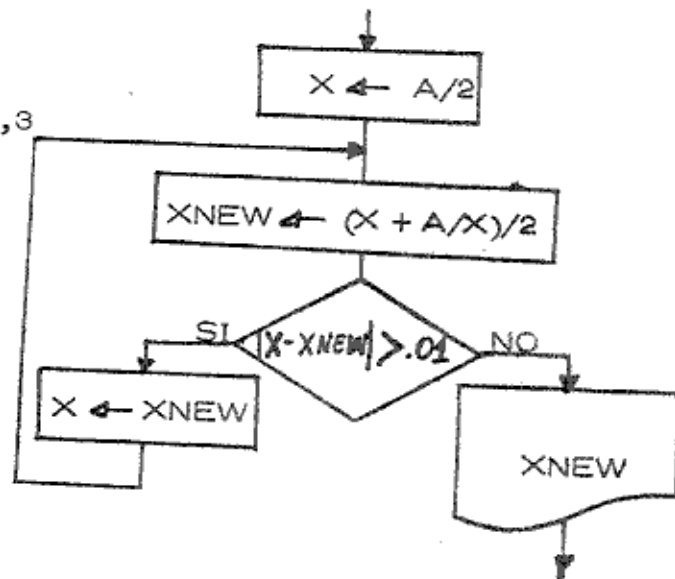
Otro ejemplo:

Consideremos el siguiente algoritmo, usado para calcular la raíz cuadrada de un número positivo A.

$$X_{i+1} = \frac{X_i + A/X_i}{2}$$

Si suponemos que el valor de A se encuentra almacenado, entonces podemos programar este procedimiento en la siguiente forma:

- X = A/2.
- 1 XNEW=(X+A/X)/2.
 - IF (ABS(X-XNEW)-.01) 2,2,3
 - 3 X=XNEW
 - GØ TØ 1
 - 2 WRITE(6, *)XNEW



- Paso 1 inicia el primer estimativo para el valor de la raíz cuadrada
- Paso 2 calcula la siguiente aproximación de la raíz usando el procedimiento descrito en el algoritmo
- Paso 3 compara el último estimativo para el valor de la raíz con el estimativo obtenido previamente. Si dos estimativos consecutivos se diferencian por un valor menor o igual a ± 0.01, se imprime el valor actual

2. reemplaza el valor actual de X con el valor último estimativo. (Este paso es necesario ya que el algoritmo calcula estimativos sucesivos en base del último estimativo)
3. devuelve el control al comienzo del ciclo
4. es una proposición de salida que imprime el resultado una vez que la comparación hecha en el paso 3 revela una diferencia satisfactoria.

Debemos observar que los dos ejemplos dados utilizan el procedimiento de repetición del ciclo. En el cálculo del factorial el ciclo se repite exactamente 20 veces. En el cálculo de la raíz cuadrada el ciclo se repite un número de veces indefinido, hasta que la comparación de prueba se satisface.

El ciclo DO es una manera de codificación FORTRAN que permite producir los pasos descritos en los ejemplos automáticamente. La proposición DO está íntimamente relacionada con el símbolo de iteración.

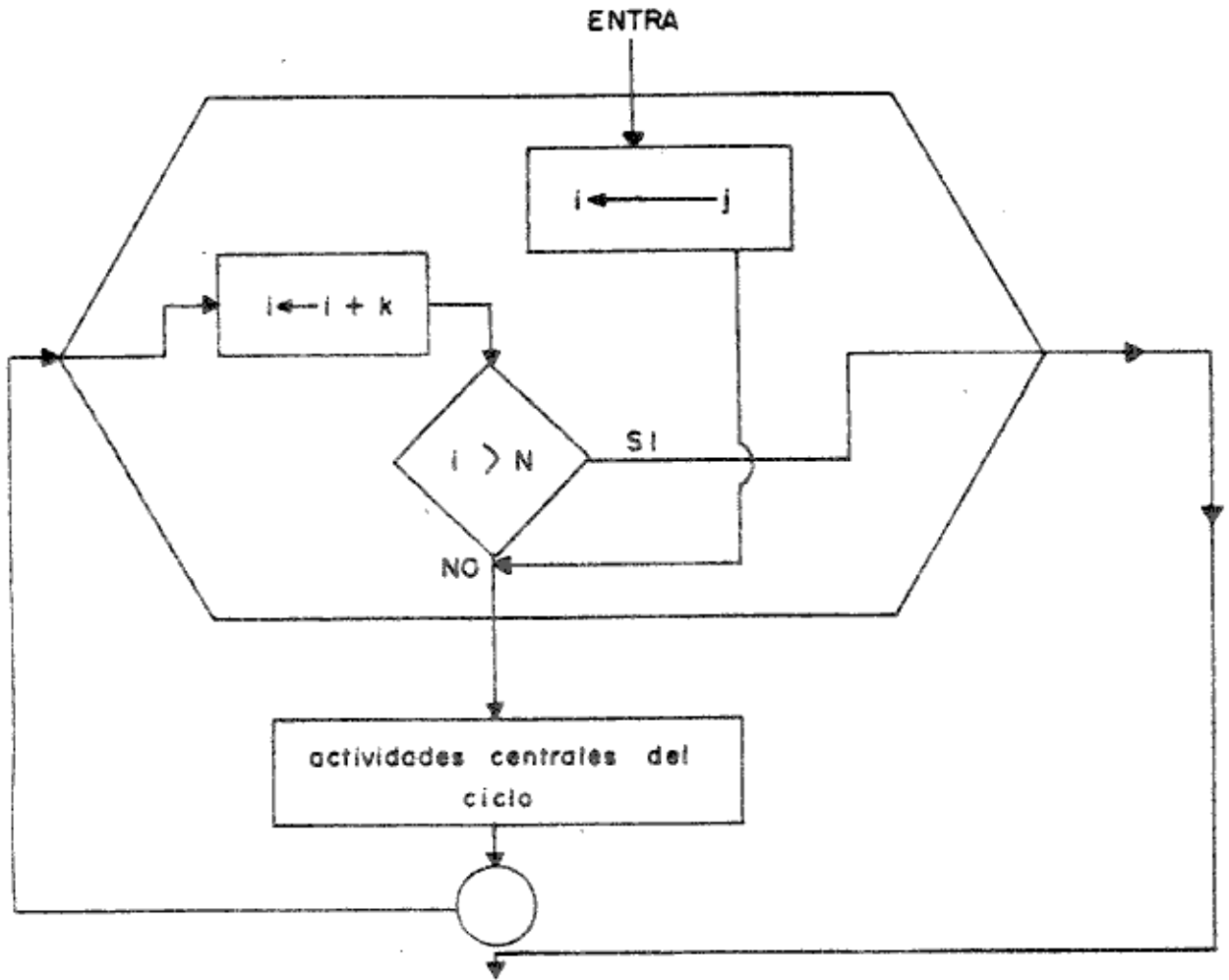
REPRESENTACION SIMPLIFICADA DEL CICLO DE ITERACION: Una iteración puede usarse tan a menudo que simplificar su representación se justifique.

Considerando en forma general el ciclo de iteración tenemos las siguientes características.

1. Iniciación del índice del ciclo de iteración (contador).
2. Realización de actividades centrales del ciclo iteración.
3. Incremento del contador.
4. Comprobación del contador con un valor límite n.

La finalidad más importante del ciclo de iteración es hacer que estas actividades centrales se repitan n veces.

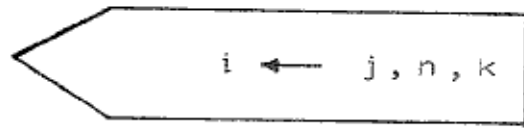
Ahora nuestro propósito es introducir un símbolo que nos represente completamente el punto de unión de la iteración con las demás actividades a desarrollar en el diagrama de flujo. Las actividades que se sustituyen las representaremos dentro del símbolo.



Descripción:

1. A \underline{i} se da un valor entero inicial \underline{j} .
2. Después ejecutamos las actividades centrales.
3. Se incrementa el valor de \underline{i} .
4. Se compara el valor de \underline{i} con N . Si se excede, se ejecuta la proposición - que sigue del círculo. Si \underline{i} no excede el límite de N , se ejecutan de nuevo las actividades centrales (2).
5. Cuando se termina la iteración, \underline{i} deja de tener un valor definido (se queda con el último valor $n + 1$) y no puede usarse de nuevo sin volver a definir - su valor (ejemplo $i \leftarrow 1$).

Como una convención arbitraria, usaremos el siguiente símbolo adelante de las actividades centrales de iteración.



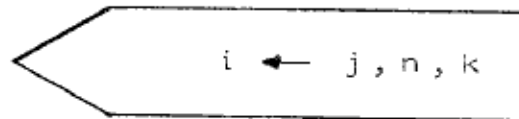
asígnale a i el valor de j, incrementalala en un valor k hasta un valor n

vamos a estar de acuerdo en que el valor inicial j, el valor de comprobación n y el incremento k están restringidos a ser sólo datos de referencia y no expresiones*. Vamos a convenir también arbitrariamente en que los datos de referencia deberán de ser positivos y enteros**. Se acordará aquí en suministrar un número encerrado en un círculo entre la última actividad central y la siguiente actividad del diagrama, este número servirá como punto final del ciclo y que siga ejecutando las siguientes actividades del diagrama de flujo.

NOTA: En la H.P. FORTRAN/3000;

- * j, n, k pueden ser expresiones aritméticas de algún tipo excepto complejos. Los valores son truncados a enteros.
- ** j y n pueden ser valores positivos, negativos o ceros. k puede ser un valor positivo o negativo, pero no cero.

PROPOSICION DO:



asígnale a i el valor de j, incrementalala en un valor k hasta un valor n

La proposición ejecutable DO, puede escribirse de dos maneras:

DO mi = j, n, k

o

DO mi = j, n

donde

m es el número encerrado en el círculo que se encuentra entre la última actividad central y la siguiente actividad del programa. Sirve como punto final del ciclo. La proposición que lleva este número se escribe

m CONTINUE

sólo se usa como punto de referencia para transferir el control al DO. Puede omitirse la proposición CONTINUE poniendo en la última proposición que encierra el DO, el número m

i es una variable entera, sin índice cuyo valor se aumenta en repeticiones sucesivas del ciclo. Se llama índice del ciclo

j, n, k son constantes enteras sin índice o variables enteras sin índice, que llevan el nombre de parámetros del índice

j es el valor de iniciación de i

n el valor máximo que i puede asumir y lleva el nombre de valor de prueba

k es el valor del incremento que se da a i en cada repetición del ciclo. Se llama el incremento. Cuando k no aparece en el DO, como en la 2a. representación, se le asigna un valor igual a uno.

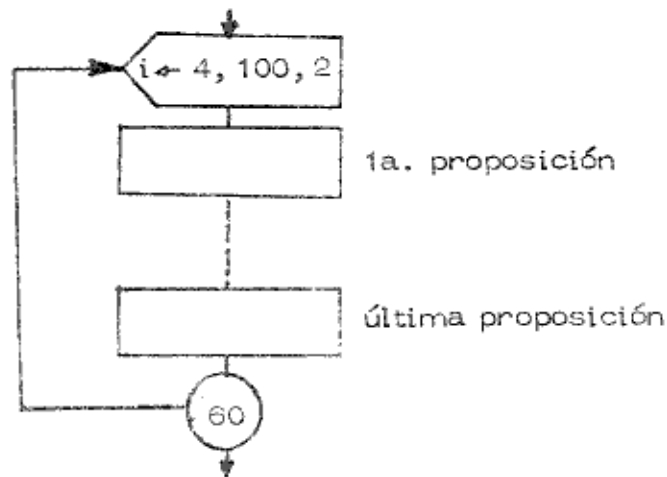
Ejemplo del uso de la proposición DO

```

7
DO 60 I = 4, 100, 2

```

La instrucción comienza en la columna 7 y se lee de la siguiente manera: Ejecuta desde la siguiente instrucción hasta la proposición número 60 inclusive, desde que I vale cuatro, hasta que I vale 100 incrementando el valor de I de dos en dos.



RESTRICCIONES DE LA PROPOSICION DO

La proposición que se usa en un programa FORTRAN debe seguir las siguientes reglas :

1. El intervalo del DO no debe incluir proposición alguna que redefina el índice (en FORTRAN 3000, sí se puede, consultar el manual página 44).

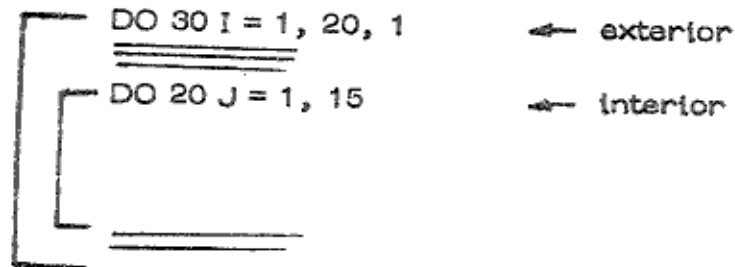
```

DO 90 I = 1, 15
.
.
.
I = .....      NO ES VALIDO
.
.
.
90

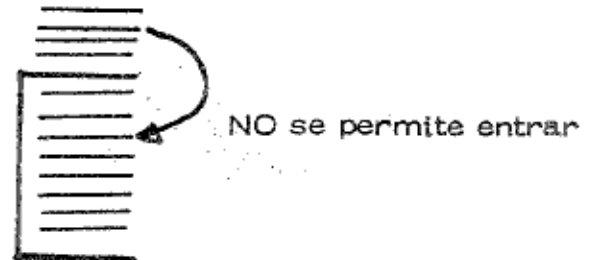
```

2. Si los parámetros del índice se escriben como variables enteras, entonces el intervalo del DO no puede incluir proposición alguna, que redefina alguno de estos parámetros. Es decir, a las variables j , n , k se les debe "asignar" un valor antes de que el programa llegue al comienzo del DO.
3. La primera proposición en el intervalo del DO debe ser una proposición ejecutable.
4. La última proposición en el intervalo del DO no debe ser una proposición de transferencia. Si el programa así lo exige, entonces se usa como proposición última un CONTINUE.

5. Es posible anidar dos o más ciclos DO siempre y cuando el DO interior se encuentre enteramente dentro del DO exterior. No hay límite para el número de ciclos que se pueden anidar. Los nombres que se usan para los índices de un DO interior deben ser diferentes a los usados en el DO exterior; más aún, estos nombres deben ser diferentes entre ellos mismos.



6. Es permitido transferir el control fuera del intervalo del DO.
7. No se permite transferir el control desde una proposición del intervalo a una proposición en el intervalo del DO. La única excepción a esta regla ocurre cuando el intervalo incluye una proposición que traslada control a un subprograma (se verá en otras unidades), en cuyo caso la transferencia al ciclo es en realidad el regreso del control desde el subprograma.



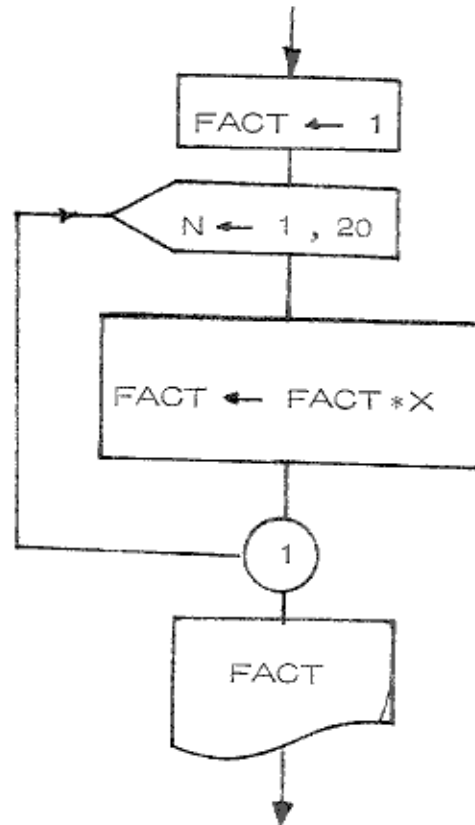
8. Es permitido transferir control dentro del intervalo del DO. En un nido de ciclos DO es posible transferir control desde un ciclo interior a un ciclo exterior ya que la transferencia ocurre hacia afuera del ciclo interior y dentro del intervalo del ciclo exterior. La transferencia en sentido opuesto no se permite.

Como ejemplo del uso de la proposición DO, vamos a calcular 20!

```

FACT=1
DO 1 N=1, 20
1 FACT=FACT * X
WRITE(6, *) FACT

```



- Paso a Inicia FACT
 Paso b Inicia el ciclo DO
 Paso c Calcula
 Paso d Imprime los resultados

Cuando la proposición DO se cumple (es decir, cuando $N=21$ en este caso) el control pasa a la proposición WRITE (es decir, la siguiente proposición). Esta es una salida normal del ciclo DO.

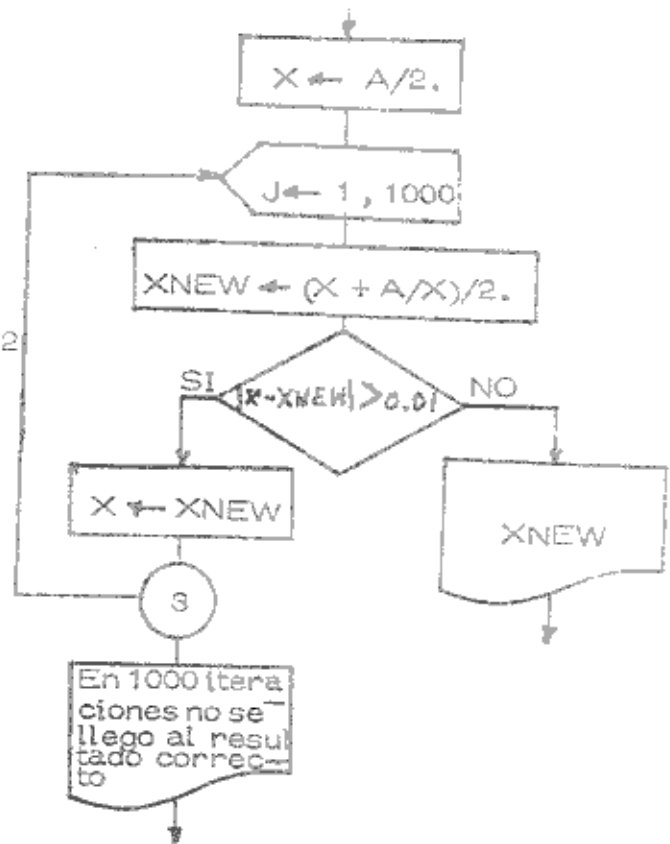
Ejemplo: 4

Examinemos ahora el uso del ciclo DO para calcular la raíz cuadrada de A. Antes de escribir el programa, debemos decidir qué número máximo de iteraciones es permisible de acuerdo con el resultado que se quiere obtener. Escogamos a 1000 como un cálculo muy conservativo. Supondremos que la localización A ya contiene el número cuya raíz cuadrada se requiere. El programa podría ser entonces:

```

X=A/2.
DO 3 J=1,1000
XNEW=(X+A/X)/2.
IF(ABS(X-XNEW).LE.0.01)GOTO 2
3 X=XNEW
2 WRITE(6,*)X NEW
.....

```



- Paso a inicia el valor de X para el primer cálculo de la raíz cuadrada
- Paso b inicial el ciclo DO
- Paso c calcula el siguiente estimativo de acuerdo con el algoritmo
- Paso d compara el último valor estimado. Si la comparación de prueba es cierta, entonces el control se transfiere fuera del ciclo DO (una salida anormal) y se imprime el resultado en el paso f. Si la comparación de prueba no es cierta, el control se transfiere a la proposición número 3, la cual se encuentra dentro del ciclo DO.
- Paso e el valor actual de X se reemplaza con el último estimativo de la raíz. Esta es la última proposición del intervalo del DO; en este punto se aumenta el valor del índice J, si el último valor de J es menor o igual a 1000, entonces el control se transfiere a la primera proposición del intervalo. Si J es mayor que 1000, entonces se lleva a cabo una salida normal con un WRITE que diga "EN 1000 ITERACIONES NO SE LLEGO AL RESULTADO CORRECTO"

Ejemplo 5: Leer y escribir una lista de 10 valores de L.

```

DO 3 I=1, 10, 1
READ(5, *) L
WRITE(6, *) 'L=', L
3 CONTINUE
STOP
END

```

Ejemplo 6 :

Hacer un diagrama de flujo y el programa para que de un conjunto de valores $x_1, x_2, x_3, \dots, x_n$ calcule cuántos de éstos negativos.

Codificación del ejemplo 6.

```

READ(5, *) N
K=0
DO 5 I=1, N, 1
IF (X) 4, 5, 5 6 IF (X.GE.0) GOTO 5
4 K=K+1
5 CONTINUE
WRITE(6, *) K
STOP
END

```

Ejemplo 7 :

El siguiente ejemplo incluye un ciclo DO anidado. El índice del ciclo principal es I. El valor inicial de I es 1, su incremento es 1 y su valor máximo es 2. El índice del ciclo anidado es J. El valor inicial de J es 1, su incremento es 1 y su valor máximo es 4.

```

C NIDO
C PROGRAMA PARA ILUSTRAR EL USO DE UN CICLO DO ANIDADO
IOTA=0
DO 4 I=1, 2
DO 4 J=1, 4
K=I*J
IOTA=I+A+J
4 WRITE(6, *) 'I=', I, 'J=', J, 'K=', K, 'IOTA=', IOTA
STOP
END

```

Desarrolla el diagrama de flujo correspondiente. Corre el programa por computadora.

Ejemplo 8:

Existen circunstancias para las cuales es inconveniente, o imposible, especificar el número deseado por adelantado de iteraciones. El proceso de iteración termina cuando se cumple una condición descrita por una proposición de prueba. La prueba se puede llevar a cabo por medio de la proposición IF aritmética, o la proposición IF lógica.

Como ilustración trataremos un programa que determina el volumen de un gas, de acuerdo a la ecuación de Van der Waal, cuando se conocen la temperatura y la presión. La ecuación de Van der Waal, para un mol de gas, es

$$P + \frac{A}{V^2} (V - B) = RT$$

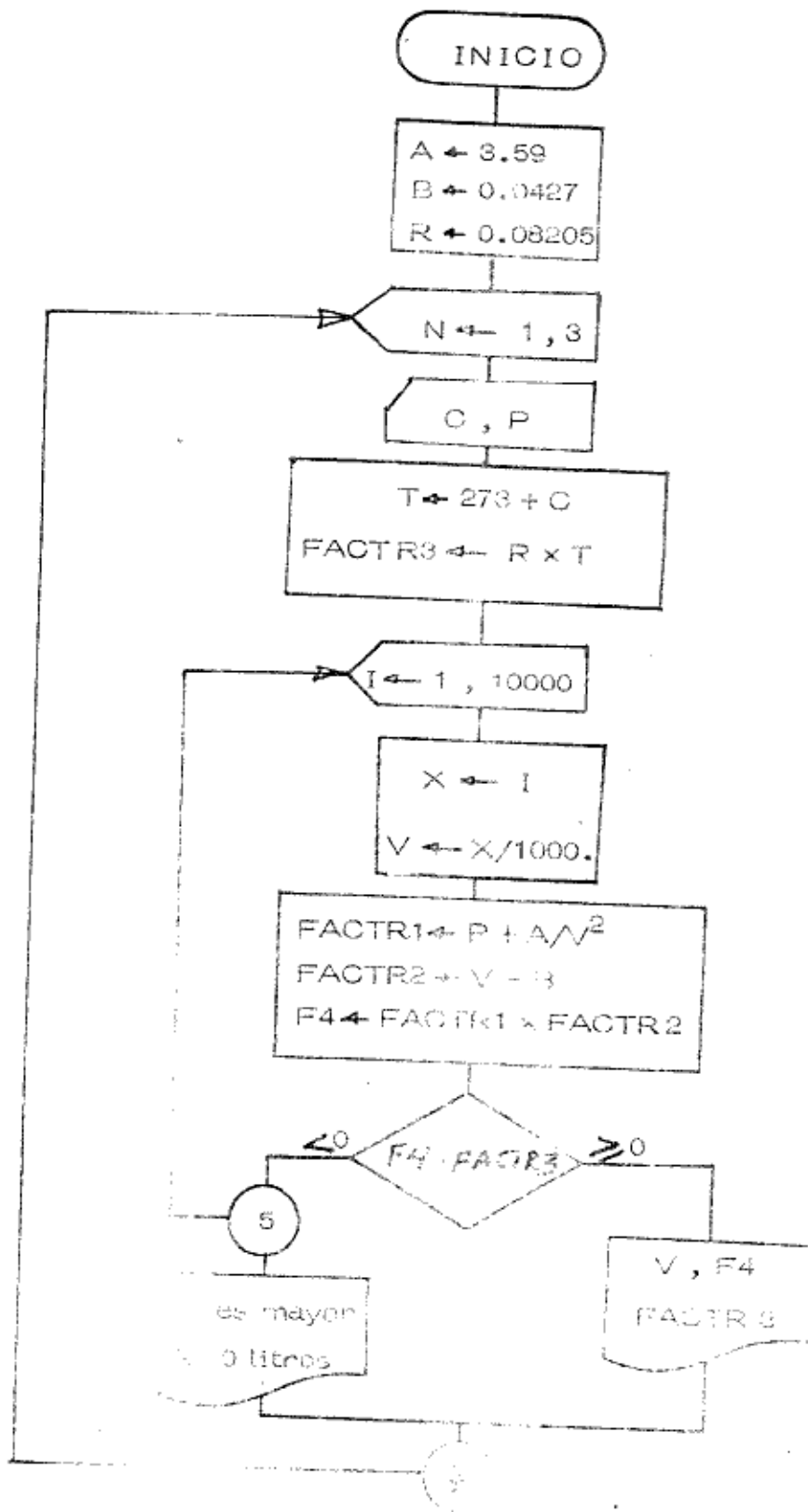
donde

- P = presión, atm
 T = temperatura absoluta, °K
 R = constante de un gas ideal = 0.08205 atm/°K mol
 V = volumen, litros
 A, B = constante para un gas dado

Deseamos encontrar el volumen de una molécula de CO₂ a una temperatura específica en grados centígrados y a una presión específica en atmósferas. Los valores de los parámetros A y B, para el CO₂ son

$$A = 3.59 \text{ atm l}^2 \text{ mol}^{-2}$$

$$B = 0.04271 \text{ mol}^{-1}$$



(F. I.)

Los nombres usados en el programa son los mismos usados como símbolos en la ecuación

Símbolos en la fórmula	Nombre en el programa
$P + \frac{A}{\sqrt{2}}$	FACTR 1
$V - B$	FACTR 2
RT	FACTR 3

El primer DO 7 N = 1, 3 (línea 4) nos indica que vamos a dar 3 pares de valores de C y P, por lo tanto los cálculos los efectuará tantas veces como pares de valores le demos a leer. El valor específico de la temperatura esta en grados centígrados y la presión en atmósferas (son datos de entrada).

Consideremos ahora el 2º DO que abarca desde la línea 8 hasta la línea 15. En la primera pasada del ciclo I = 1, X = 1 y V = 0.001. FACTR 1 y FACTR 2 se evalúan con V = 0.001. El producto de FACTR 1 y FACTR 2 se le asigna a F4. La proposición IF compara entonces F4 con F3. Puesto que F4 es negativo, entonces es menor que FACTR 3 y el programa regresa al comienzo del ciclo DO. En cada parada del ciclo DO el valor de V aumentá (en incrementos de 0.001) hasta que F4 iguala o sobrepasa el valor de FACTR 3. En este momento el programa transfiere control a la proposición número 6 (línea 16) en donde la proposición WRITE, hace que el computador "imprima" el valor actual de V, como datos de interés también se imprime F4 que es el producto de FACTR 1 * FACTR 3 y el valor FACTR 3.

Ejercicio. Escribe el mismo programa para que imprima los datos para cada cálculo.

9-JOB VAN1

C ESTE PROGRAMA CALCULA EL VOLUMEN DE UN GAS POR MEDIO DE LA
C ECUACION DE VAN DER WAAL. LAS CONSTANTES SON DEL GAS CO2, °C,
C ESTA EN GRADOS CENTIGRAOS Y P EN ATMOSFERAS.

```

1 A=3.59
2 B=0.0427
3 R=0.08205
4 DO 7 N=1,3
5 P=ANS,11C,P
6 T=273.15
7 FAC=1000*P
8 U=1-1/1000
9 R=1
10 V=V/1000
11 FACR1=P+A/V**2
12 FACR2=V-B
13 FAC=FACTR1*FACR2
14 U*(P4-FACR3)5.6.6
15 5 CONTINUE
16 WRITE(6,2)
17 GO TO 7
18 WRITE(6,3)V,F4,FACR3
19 7 CONTINUE
20 1 FORMAT(2F5.2)
21 2 FORMAT(20X,1V MAYOR A 10 LITROS)
22 3 FORMAT(20X,V=,F10.4,FACTR1*FACR2=,F10.4,FACTR3=,F10.4)
23 STOP
24 END

```

SENTRY

V MAYOR A 10 LITROS C=32 P=0.75

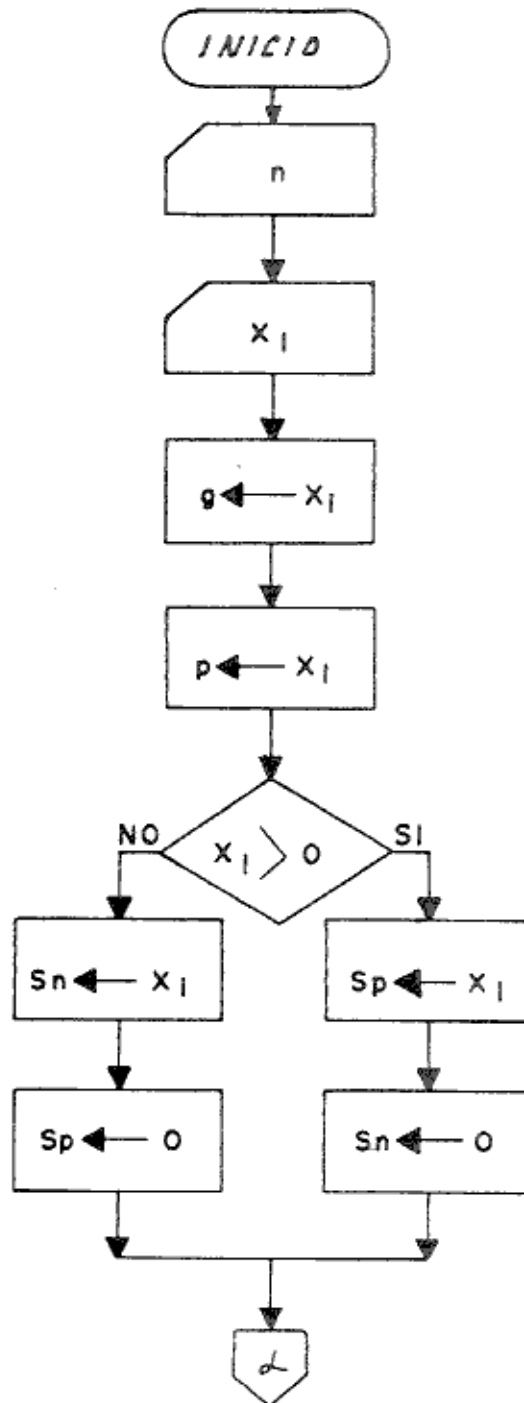
V= 3.2340FACTR1*FACR2= 25.0302FACTR3= 25.0253 C = 32. P = 7.5
V= 0.1120FACTR1*FACR2= 25.0306FACTR3= 25.0253 C = 32 P = 7.5

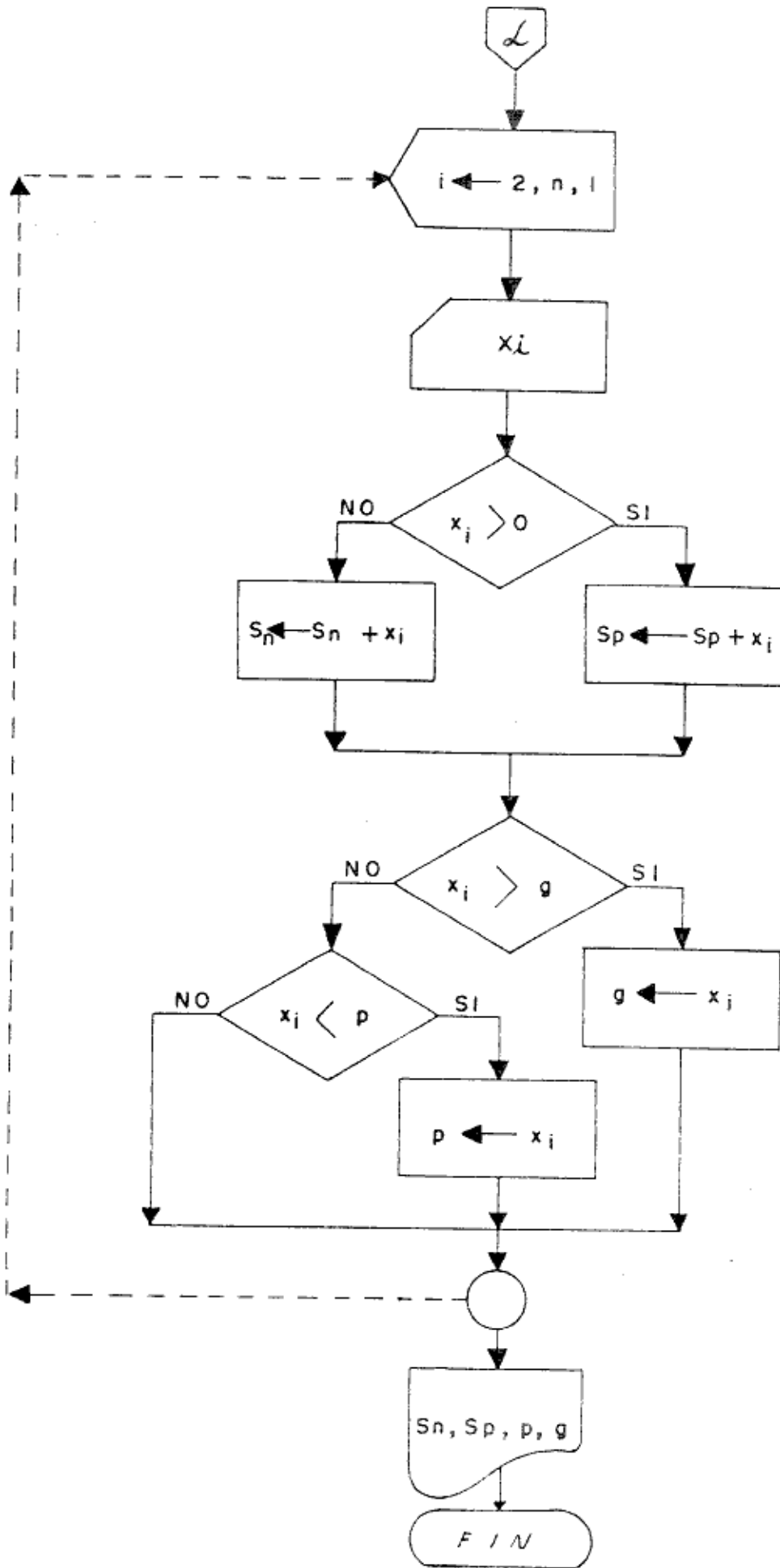
CURS USAGE OBJECT CODE= 952 BYTES,ARRAY AREA= 0 BYTES,TOTAL AREA AVAILABLE= 87800 BYTES
BYT. D. BYTES NUMBER OF ERRORS= 0, NUMBER OF WARNINGS= 0, NUMBER OF EXTENSIONS= 0
SERVILE TIME= 0.78 SEC,EXECUTION TIME= 12.56 SEC, MATFIV - JUL 1973 VIL4 9.51.45 MONDAY 9 JAN 78

Ejercicio:

Codificar el siguiente problema.

Dado un conjunto $S = x_1, x_2, \dots, x_n$. Desarrollar el diagrama de flujo para calcular, el menor, la suma de los positivos, y la suma de los negativos de los elementos S .





EJERCICIOS

1. A y B son dos vectores de n componentes, calcule

$$S = \sum A_i B_i \text{ para } i \text{ impar}$$

2. $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ son n parejas del plano x, y. Sean:

$$S_1 = \sum_{i=1}^n x_i y_i, \quad S_2 = \sum_{i=1}^n x_i \cdot \sum_{i=1}^n y_i$$

$$S_3 = \sum_{i=1}^n x_i^2, \quad S_4 = \left(\sum_{i=1}^n x_i \right)^2$$

Lea n, $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ y calcule S_1, S_2, S_3 y S_4 .

3. Se sabe de cálculo que:

$$\text{Sen } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Calcule $\text{sen } x$ para 5 valores de x usando los primeros 18 términos de la serie.

4. Se conocen los extremos a y b de un intervalo del eje real y la expresión matemática que describe a una función f. Se desea un diagrama de flujo y programa que calcula

$$\int_a^b f(x) dx \text{ usando la fórmula de Simpson}$$

$$\int_a^b f(x) dx = \frac{4h}{3} (y_1 + y_3 + \dots + y_{m-1}) + \frac{2h}{3} (y_2 + y_4 + \dots + y_{m-2}) + \frac{h}{3} (y_0 + y_m)$$

donde :

m es un número por igual al número de divisiones iguales del intervalo $[a, b]$.

$$h = \frac{b - a}{m}$$

$$y_i = f(a + ih) \quad ; \quad i = 0, 1, 2, \dots, m$$