

Paloma Pérez - Lemaur

DIAGRAMAS DE FLUJO

**ejercicios
y problemas**

PARANINFO S.A.

MADRID

© PALOMA PEREZ-LEMAUR GUEZALA
GRANADA (España)

Reservados los derechos para todos los países.
Ninguna parte de esta publicación, incluido el diseño
de la cubierta puede ser reproducido, almacenado o
transmitido de ninguna forma, ni por ningún medio,
sea éste electrónico, químico, mecánico, electro-óptico,
grabación, fotocopia o cualquier otro, sin la previa auto-
rización escrita por parte de la editorial.

Impreso en España
Printed in Spain

ISBN: 84-283-1555-8

Depósito Legal: M.: 26.670

PARANINFO SA

Magallanes, 25 - 28015 Madrid (0731 0/37/01)

Artes Gráficas BENZAL, S.A. - Virtudes, 7 - 28010 Madrid

INDICE DE MATERIAS

PROLOGO	7-8
1. INTRODUCCION A LOS ORGANIGRAMAS	9
Estudio y confección de organigramas	10
Símbolos	11
Preparación de un organigrama	12
Definición de variable y contenido de variable	13
Significación del signo igual "=" en un organigrama	14
2. LOS PRIMEROS ORGANIGRAMAS	21
Ejercicios 1 a 31	22-53
3. TRABAJANDO CON TABLAS	55
Ejercicios 32 a 65 con variables complejas	57-94
4. TRABAJANDO CON FICHEROS	95
Ejercicios 66 a 76 con ficheros en memorias auxiliares	96-112
APLICACIONES COMPLETAS	112
<i>Primera aplicación (Seis programas)</i>	<i>112</i>
1. Creación del fichero de empleados	113
2. Inserción de registros	114
3. Consulta, modificación y eliminación de registros	115
4. Creación de tablas	118

5. Modificación de la tabla general	120
6. Proceso	121
<i>Segunda aplicación (Cinco programas)</i>	127
1. Creación del fichero de alumnos	128
2. Adición de registros en el fichero	129
3. Eliminación y modificación del fichero	132
4. Listado de notas por alumno	134
5. Media de notas por evaluación y por asignatura	135
 <i>Tercera aplicación</i>	137
Estadística tipo I	137
Estadística tipo II	148
Estadística tipo III	156
 5. ORGANIGRAMAS ESTRUCTURADOS	171
Ejercicios 77 a 100	176-190

PROLOGO

Se ha comprobado la importancia del Organigrama, dentro del análisis de un problema, para la comprensión y documentación de cualquier trabajo informático.

También se ha comprobado la dificultad que supone, para cualquier persona que empieza a programar, el comprender lo que es un Organigrama y más aún el hacer dicho Organigrama. Esto es debido a que es muy difícil poder pensar y estructurar el problema a resolver, desde el punto de vista informático.

Con este libro se pretende ayudar a comprender esta nueva forma de representación. Se intenta acostumbrar paulatinamente a los principiantes a pensar que para resolver un problema debe descomponerse en sus partes y estudiar cada una de ellas por separado y paso a paso.

Para ello se empieza explicando lo que es un Organigrama y cómo se representa gráficamente. A continuación se realizan Organigramas muy sencillos, que se van complicando poco a poco.

Con ello pretendo que el lector disponga de un libro donde poder consultar varios tipos de Organigramas. Que pueda ver resueltos distintos problemas, por medio de Organigramas, sin basarse en ningún lenguaje en concreto, y que compruebe su utilidad para comprender o documentar cualquier trabajo.

Este libro se compone de:

- Una introducción, donde se explica la forma de hacer un Organigrama, indicando los símbolos y las reglas que hay que emplear para su construcción.
- 31 Organigramas resueltos, utilizando variables simples.
- 34 Organigramas resueltos, con variables subindexadas o compuestas.

- 10 Organigramas resueltos, trabajando con ficheros secuenciales, indexados y relativos.
- 5 Aplicaciones completas.
- 24 Organigramas estructurados.

Para la redacción del presente trabajo he recibido una inestimable ayuda, que no quiero dejar de agradecer públicamente, del que fue mi profesor Don Decoroso Crespo López, e igualmente el estímulo de Don Carlos Vega, Jefe de Estudios de la Facultad de Informática.

1

INTRODUCCION A LOS ORGANIGRAMAS

ESTUDIO Y CONFECCION DE ORDINOGRAMAS

Para resolver un problema por medio del ordenador, hay que realizar un estudio del problema y de la información de partida con la que contamos. Este estudio se realiza por medio del análisis orgánico y del análisis funcional del sistema.

Una vez definida claramente la información de partida, hay que hacer el estudio del tratamiento que le vamos a dar a dicha información. Tendremos que indicar:

- Las operaciones aritméticas y lógicas, que vamos a realizar sobre ella.
- Las decisiones que vamos a necesitar tomar a lo largo del tratamiento.
- La entrada de datos, cuándo y cómo se va a realizar.
- La salida de resultados, cuándo y cómo se va a producir, etc.

Todo este estudio se representa por medio de un esquema gráfico, que realiza el analista, y que denominamos ordinograma u organigrama de proceso.

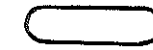
Así pues, un ordinograma es un esquema gráfico, que representa todas las operaciones a realizar por el ordenador, sobre la información de entrada, así como el orden en que deben ser realizadas dichas operaciones.

Es conveniente la realización de dicho organigrama:

- Para ver una visión global del problema.
- Para observar en qué áreas se divide dicho problema.
- Para localizar las entradas y salidas de cada una de las áreas en que hemos dividido el problema.
- Para poder observar si se han tenido en cuenta todas las posibilidades.
- Para que a la hora de realizar una corrección, sea más fácil, etc.

A continuación veremos la simbología utilizada para realizar el organigrama y las reglas que hay que seguir.

Símbolos estándar



→ Comienzo y fin



→ Proceso en general



→ Toma de decisiones



→ Entrada de datos por tarjeta



→ Entrada de datos por teclado



→ Salida de resultados por pantalla



→ Salida de datos por impresora



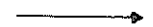
→ Unidad de cinta



→ Unidad de disco



→ Conexiones a otra página

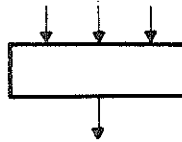


→ Líneas de conexión, dirección del flujo

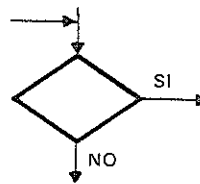
Reglas a seguir para la confección de organigramas:

1. Todos los símbolos deben de estar conectados, por medio de líneas de conexión.
2. Se debe dibujar de forma que se pueda mirar el organigrama claramente de arriba hacia abajo y de izquierda a derecha.

3. A un símbolo de proceso pueden llegarle varias líneas de conexión.



4. A un símbolo de decisión pueden llegarle varias líneas de conexión, pero de él sólo pueden salir dos líneas, que corresponden al hecho de que se cumpla la condición y al hecho de que no se cumpla la condición a evaluar.



5. A un símbolo de inicio de proceso no llega ninguna línea de conexión y de él sólo puede partir una línea de conexión.
6. A un símbolo de final de proceso, pueden llegar muchas líneas de conexión, pero de él no puede partir ninguna.

PREPARACION DE UN ORGANIGRAMA

Al realizar un programa, para resolver un problema, se pueden realizar distintos tipos de operaciones, las cuales vamos a clasificar en tres grandes grupos:

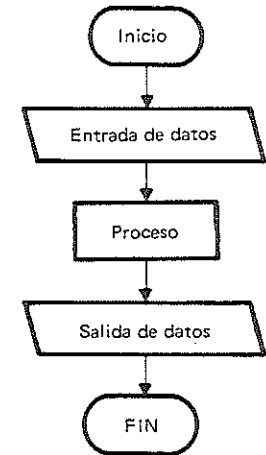
1. *Operaciones de entrada:* sirven para introducir datos desde el exterior, a las unidades de almacenamiento del ordenador (Memoria interna o auxiliar).

2. *Operaciones de proceso:* sirven para manipular los datos almacenados. Estos resultados son también almacenados en unidades de almacenamiento del ordenador.

3. *Operaciones de salida:* sirven para extraer los resultados de las unidades de almacenamiento y comunicarlás al usuario (exterior).

Por lo tanto, podemos representar cualquier problema por el organigrama básico:

Este diagrama, a pesar de ser tan general, puede servir para diseñar otros más detallados. Normalmente es recomendable hacer primero un organigrama general del problema y luego ir detallándolo más.



DEFINICION DE VARIABLE Y CONTENIDO DE VARIABLE

Definimos una variable, como un nombre o combinación de caracteres, inventado por el programador, que representa el contenido de una posición de memoria. Así pues, al referirnos en un programa a una variable determinada, nos estamos refiriendo a una determinada posición de memoria.

Las variables, según sea su contenido, se clasifican en variables numéricas y variables alfanuméricas.

Denominamos variable numérica a aquella cuyo contenido es una constante numérica. Por ejemplo, si la variable A contiene un 5, diremos que es una variable numérica. Normalmente a las variables numéricas se las denomina por medio de una letra o un conjunto de letras.

Decimos que una variable es alfanumérica cuando su contenido es una constante alfanumérica, es decir, es un conjunto de

números y letras y signos especiales. Por ejemplo "TOTAL". o "FPI" serán constantes alfanuméricas. Si la variable A\$ contiene una de estas dos constantes, será una variable alfanumérica. Normalmente las variables alfanuméricas son designadas por medio de una letra o un conjunto de letras, seguidas del símbolo del dólar.

SIGNIFICADO DEL SIGNO IGUAL "=" EN UN ORGANIGRAMA

El signo igual en un organigrama se utiliza, para asignar un contenido a una variable, es decir, para darle un valor a una variable.

Por ejemplo, si ponemos $H = 341$, estamos indicando que quremos guardar en la variable H, el valor numérico 341. Si ponemos $A\$ = \text{"SUMA Y SIGUE"}$, estamos indicando que el literal o constante alfanumérica, lo almacenamos en la variable alfanumérica A\$.

El signo igual se suele utilizar para inicializar las variables al principio del programa, es decir, para dar un valor inicial a las variables. Por ejemplo, si en el programa vamos a utilizar la variable S para alguna operación, al principio del programa le tendremos que dar un valor inicial $S = 0$

No es imprescindible inicializar las variables que van a contener el resultado de una entrada de datos, ni tampoco aquellas que van a contener el resultado de una operación. Pero será preciso inicializar todas las variables que van a intervenir en alguna operación aritmética. Por lo tanto, es recomendable inicializar todas las variables que intervienen en el proceso.

El signo igual también lo utilizamos para asignar a una variable el resultado de una expresión aritmética o lógica. Por ejemplo:

$$X = 4 * A * C/D$$

Siendo las expresiones aritméticas un conjunto de variables y constantes, relacionadas por medio de operadores aritméticos, los cuales son:

- + para la suma
- para la resta
- * para el producto
- ** para la potencia
- / para la división

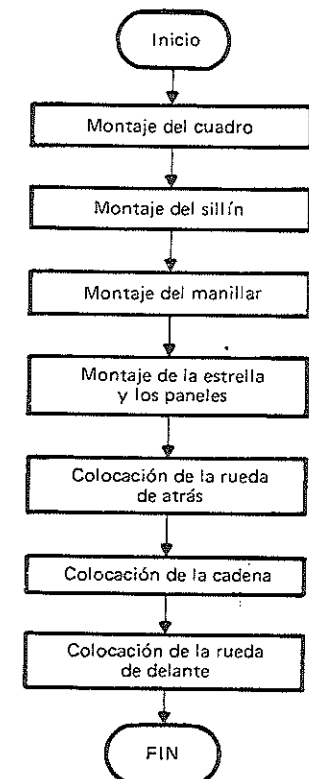
También se puede almacenar en una variable el resultado de una expresión lógica. Una expresión lógica es un conjunto de variables y constantes, relacionadas por medio de los operadores lógicos, los cuales son:

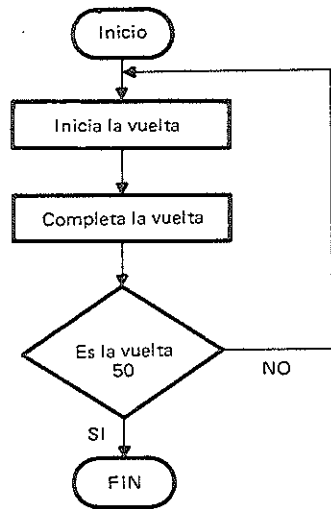
- > para mayor que
- < para menor que
- = para igual
- ≠ para menor o mayor
- ≤ para menor o igual
- ≥ para mayor o igual

A continuación veremos un ejemplo sencillo de organigrama: vamos a ver los pasos que tenemos que dar para el montaje de una bicicleta:

Las instrucciones en un programa se ejecutan secuencialmente, una detrás de otra, en el mismo orden en que se escriben. Por lo tanto, para realizar cualquier programa u organigrama debemos de ir colocando las órdenes tal y como queremos que se ejecuten, sin saltarnos ninguna.

Algunas veces, en un programa, necesitamos ejecutar una instrucción o varias, un número determinado de veces, en el mismo lugar del programa. Para no tener que escribir dichas instrucciones varias veces, lo que hace es crear un bucle, de forma que se pase varias veces por el mismo conjunto de instrucciones. Un ejemplo sería el de un escalectrix, en el que queremos que un coche dé 50 vueltas:

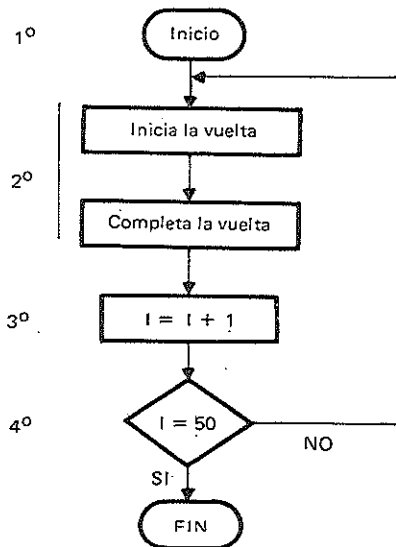




Como vemos en este organigrama ejecutamos las órdenes secuenciales como en el anterior, pero en un momento determinado, tenemos que romper esta secuencia y volver atrás para volver a ejecutar un número de instrucciones que queríamos repetir. Por lo tanto, en este organigrama existe un bucle o ciclo.

Para completar este organigrama, podemos ponerle un contador, que nos vaya indicando por qué vuelta vamos. Este contador lo haremos con la variable I, de tal forma que cada vez que completemos una vuelta al circuito, la sumamos uno a la variable I.

En este caso, para saber si se ha terminado el proceso, sólo tenemos que preguntar si la variable I vale ya 50.

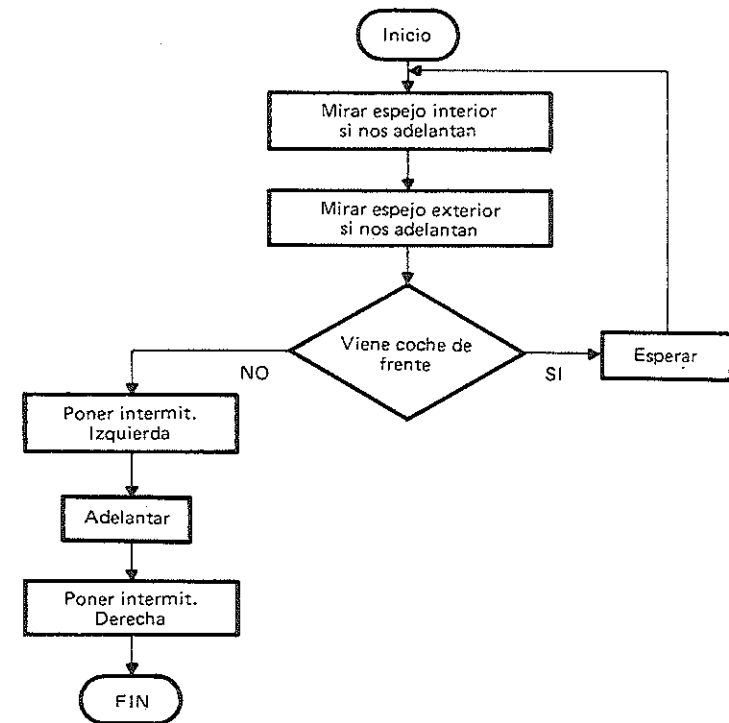


Normalmente un bucle consta de cuatro partes bien diferenciadas:

- Inicio del bucle.
- Progreso del bucle.
- Cuerpo del bucle.
- Decisión.

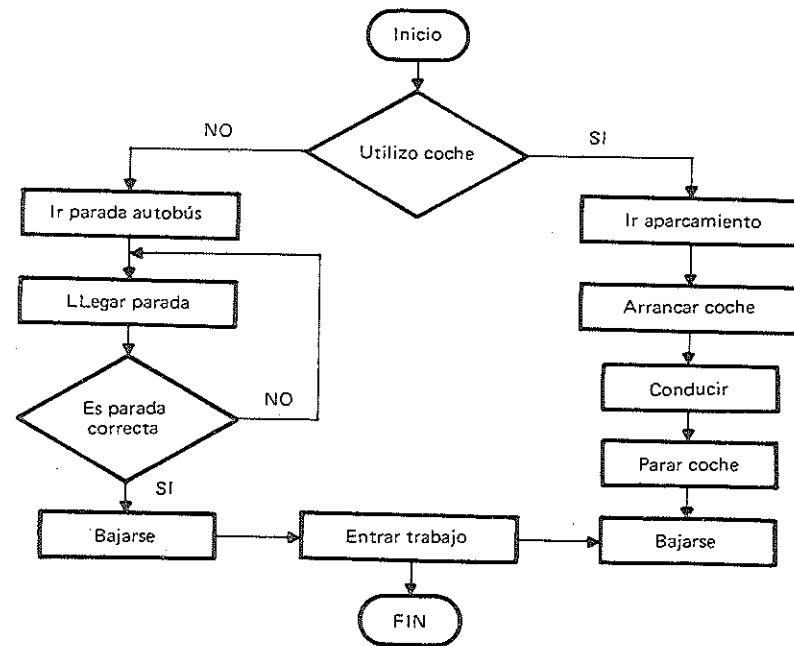
Estas cuatro partes las hemos señalado con un número en nuestro ejemplo.

Existen otros programas u organigramas que nos permiten continuar con fases diferentes, de acuerdo con una determinada condición. Se caracterizan por tener diferentes ramas, de las cuales sólo se ejecuta en un momento determinado una de ellas. Vamos a ver un ejemplo: queremos hacer el organigrama necesario para ver los pasos que hay que dar para realizar un adelantamiento con un coche en una carretera.



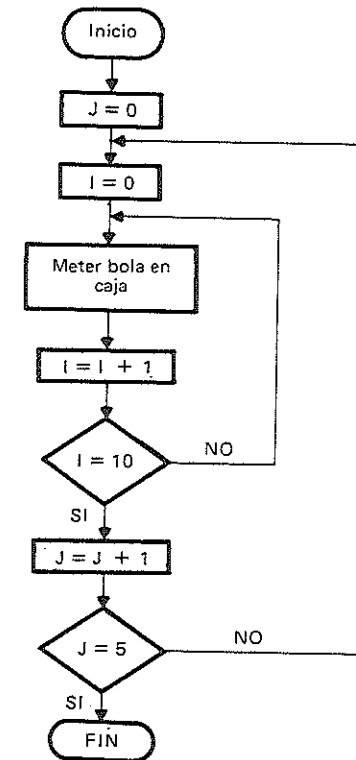
En este organigrama vemos que en un momento determinado rompemos el orden secuencial de las instrucciones, volviendo hacia atrás para ejecutar otra vez una instrucción ya ejecutada. Vemos también que en un momento determinado evaluamos una condición, y dependiendo de si se cumple o no, tomamos dos caminos diferentes; entonces decimos que se ha producido una bifurcación. Una bifurcación consiste en romper la secuencia de ejecución, para saltar a otro lugar del programa, bien hacia atrás o bien hacia adelante.

A continuación veremos un ejemplo, en el cual tenemos dos formas diferentes y dentro de una de ellas un bucle. Supongamos que una persona sale de casa para ir a trabajar, de forma que puede ir en su coche o coger el autobús. Según decida hacer una cosa u otra tendrá que dar determinados pasos, por lo tanto tendremos dos ramas diferentes:



Hemos visto lo que era un bucle. Estos bucles pueden estar encerrados unos dentro de otros, formando lo que se llama bucles anidados. Se tiene que completar el bucle más interno, antes de que se acabe el bucle más externo. Normalmente se ejecuta entero el bucle más interno por cada vuelta que de el bucle más externo. Por ejemplo, si los dos bucles se tienen que ejecutar N veces, por cada vez que se ejecute el bucle externo, se debe de ejecutar N veces el bucle más interno.

Vamos a ver los bucles anidados con un ejemplo. Supongamos que queremos llenar 5 cajas con 10 bolas cada una, de forma que 1.^o llenemos la 1.^a caja, después la 2.^a, etc.



En este organigrama hemos utilizado la variable I para contar las bolas que vamos introduciendo en cada una de las cajas y la variable J, para contar las cajas que vamos llenando. Por lo tanto I tendrá que ir variando desde uno hasta 10 y J variará desde 1 hasta 5.

El bucle interno (el de la variable I) se debe ejecutar 10 veces para introducir las 10 bolas en la 1.^a caja. Cuando se haya ejecutado esas 10 veces, pasaremos al bucle externo (el de la variable J) aumentando uno J, con lo cual indicamos que ya hemos llenado la 1.^a caja, y preguntaremos si ya hemos llenado las 5 cajas, es decir, si J ha llegado a 5. Si nos dice que sí, es el final; si nos dice que no, tendremos que volver al bucle interno para llenar otra caja, para lo cual tendremos que volver a poner I a cero para que nos cuente las bolas de la 2.^a caja, y así sucesivamente.

Como vemos tendremos que realizar 10 veces el bucle interno, por cada vez que realicemos el bucle externo.

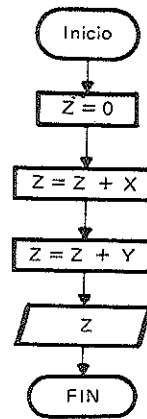
A continuación veremos una serie de organigramas, con el fin de acostumbrarnos a su manejo. Empezaremos por los más sencillos y los iremos complicando poco a poco.

2 LOS PRIMEROS ORGANIGRAMAS

① Hallar la suma de dos números: X e Y.

Lista de variables

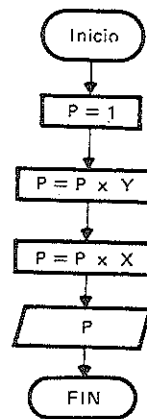
Z = contendrá la suma acumulada.



② Hallar el producto de dos números, X e Y.

Lista de variables

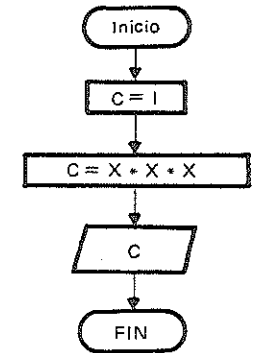
P = contendrá el producto acumulado.



③ Hallar el cubo de un número X.

Lista de variables

C = contendrá el cubo de X

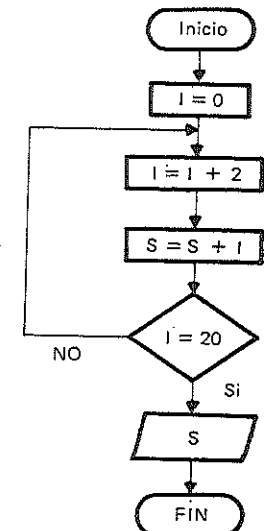


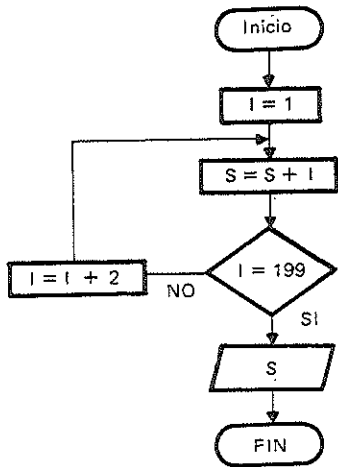
④ Hallar la suma de los diez primeros números pares.

Lista de variables

I = va generando los números pares.
Valdrá 0, 2, 4, 6 etc.

S = contendrá la suma acumulada.



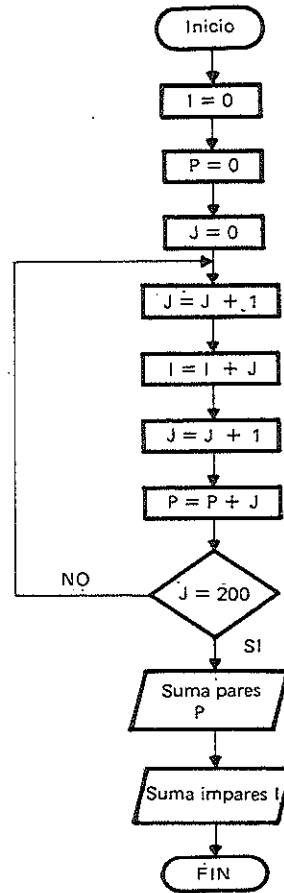


⑤ Hallar la suma de los 100 primeros números impares.

Lista de variables

I = va generando los números impares, valdrá: 1, 3, 5 ... 199

S = contendrá la suma acumulada.



⑥ Hallar la suma de los 200 primeros números pares y de los 200 primeros impares.

Lista de variables

I = contendrá la suma acumulada de los números impares.

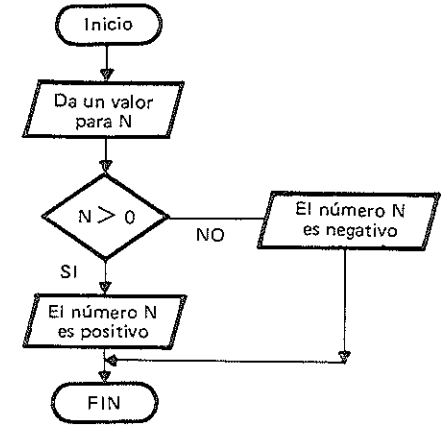
P = contendrá la suma acumulada de los números pares.

J = irá generando los números pares y los impares, de forma que valdrá: 1, 2, 3, 4, 5, etc. Es decir, le iremos sumando una unidad, empezando desde cero.

⑦ Dado un número, contestar si es positivo o negativo.

Lista de variables

N = va a contener el número a investigar. Pediremos su valor por pantalla.

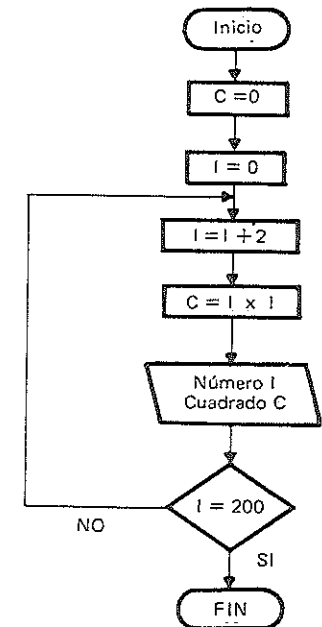


⑧ Hallar el cuadrado de los 100 primeros números pares.

Lista de variables

I = va generando los números pares, valdrá 2, 4, 6... 2000.

C = iremos guardando el cuadrado del número correspondiente.



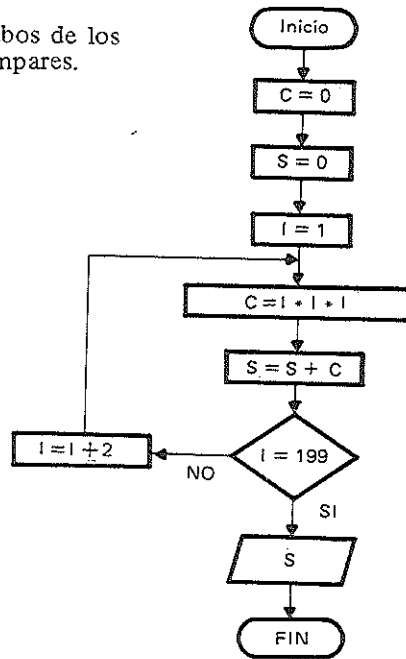
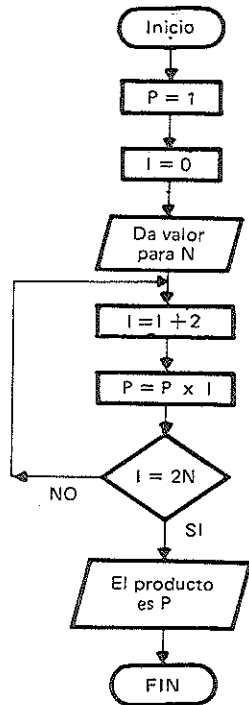
- 9 Hallar la suma de los cubos de los 100 primeros números impares.

Lista de variables

I = va generando los números impares, valdrá: 1, 3, 5 ... 199.

C = en esta variable guardaremos el cubo de cada uno de los números.

S = en esta variable guardaremos la suma acumulada.



- 10 Hallar el producto de los N, primeros números pares.

Lista de variables

N = contendrá el número total de números que tenemos que sumar. Su valor no lo conocemos al empezar, por lo tanto lo tendremos que pedir por pantalla.

I = ira generando los números pares, valdrá: 2, 4 ... 2N.

P = en esta variable iremos acumulando el valor del producto acumulado.

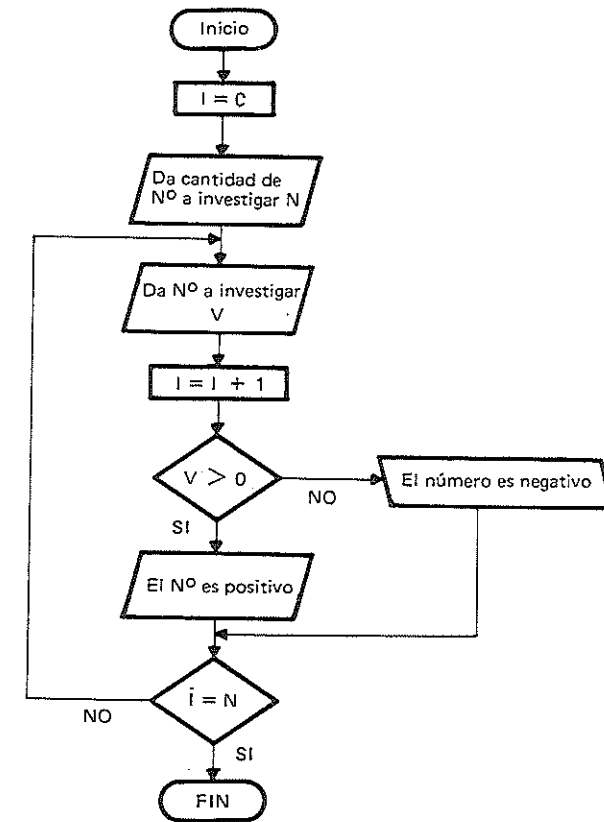
- 11 Dados N números por pantalla, contestar si son positivos o negativos.

Lista de variables

N = esta variable contendrá el número total de números que queremos investigar. Como no conocemos su valor al iniciar el proceso, lo pediremos por pantalla.

I = irá generando los números pares, valdrá 2, 4, 6 ... 2N

P = en esta variable iremos acumulando el valor del producto acumulado que vamos investigando. El proceso terminará cuando I = N.

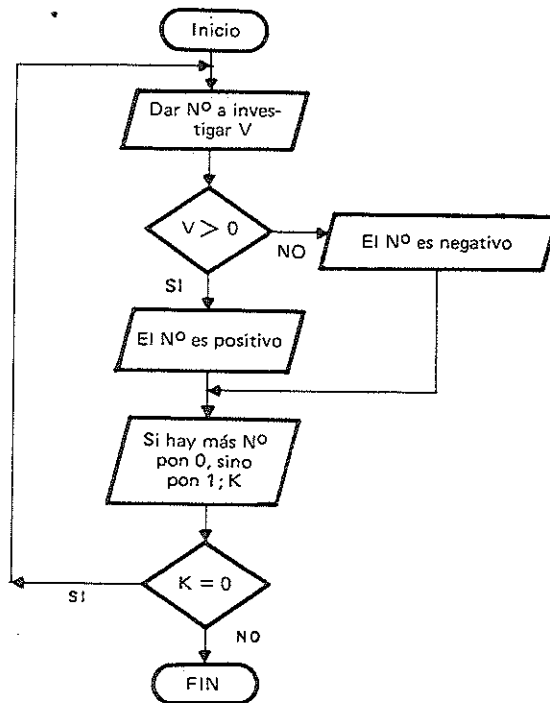


- 12 A continuación vamos a realizar el ejercicio anterior, pero en este caso no ponemos límite de números a investigar, sino que introduciremos un parámetro con el cual pararemos cuando nos interese. Para ello vamos a utilizar la variable K, la cual pondremos a 1, para parar y a 0, para seguir investigando números.

Lista de variables

K = variable para preveer el final del proceso.

V = en esta variable iremos guardando los números a investigar, los cuales iremos pidiendo por pantalla.

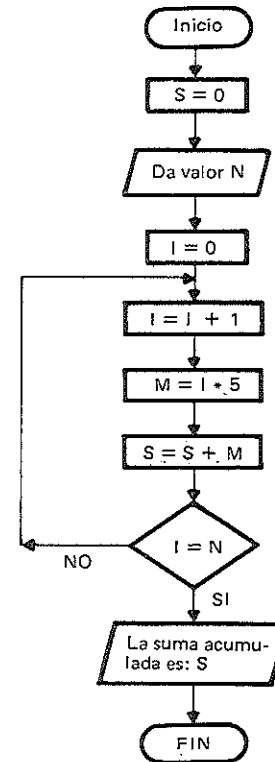
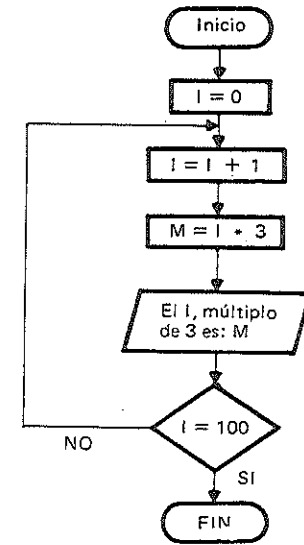


- 13 Hallar los 100 primeros múltiplos de tres.

Lista de variables

I = esta variable la vamos a utilizar para obtener los múltiplos de 3. Pues si multiplico tres por 1, 2, 3, 4 ... 100 voy obteniendo sus múltiplos. De forma que I valdrá 1, 2, 3 ... 100.

M = en esta variable guardará el múltiplo de tres obtenido.



- 14 Sumar los N primeros múltiplos de cinco.

Lista de variables

N = esta variable me indicará cuántos múltiplos de 5 tengo que sumar. Como no conozco su valor, lo pediré por pantalla.

I = esta variable me servirá para hallar el múltiplo de cinco, valdrá 1, 2, 3, 4 ... N.

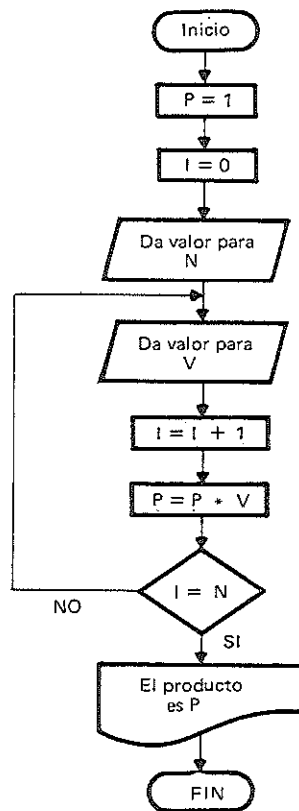
M = en esta variable guardaré el múltiplo de cinco.

S = en esta variable iremos almacenando la suma acumulada.

- 15) Hallar el producto de N números que entren por pantalla.

Lista de variables

N = esta variable nos indica los números a efectuar el producto. Como no la conocemos la pedimos por pantalla.
 V = en esta variable almacenamos el número a multiplicar. La pediremos por pantalla.
 I = esta variable irá contando los números pedidos.
 P = en esta variable iremos acumulando el producto.



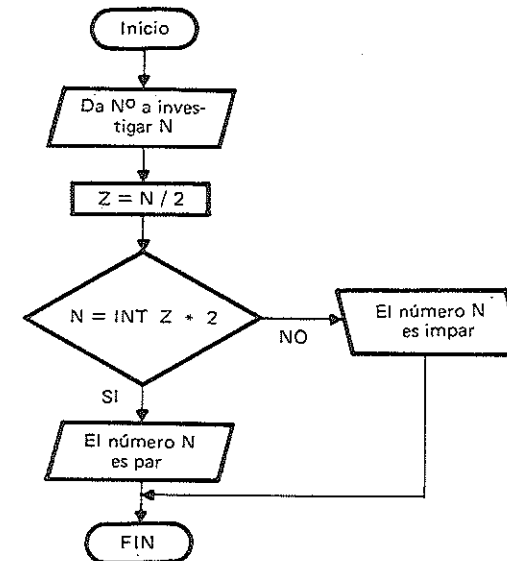
- 16) Dado un número por pantalla, contestar si es par o impar

Lista de variables

N = en esta variable almacenaremos el número a investigar. Como no lo conocemos lo pediremos por pantalla.

Z = para saber si es par un número, nos basta con dividirlo por dos. Si la división es exacta es que el n.º es par. En la variable Z, guardaremos el resultado de la división. A continuación cogeremos la parte entera de Z, para saber si la división es exacta o no.

Ejemplo: N = 5; Z = N/2 = 2,5. Si el número es par N = 2 x parte entera de Z. En nuestro ejemplo N ≠ 2 x INT Z. Por el contrario si N = 6; Z = N/2; Z = 3; N = 2 x INT Z = 2 x 3.



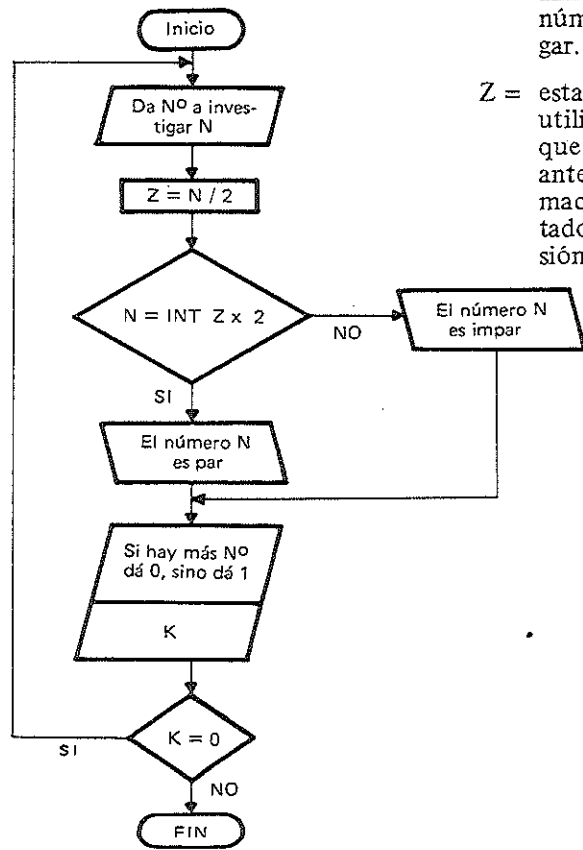
- 17 A continuación vamos a realizar el mismo ejercicio anterior, pero no para un solo número, sino para todos los que queramos, es decir, vamos a preveer un final.

Lista de variables

K = esta variable la vamos a utilizar, para preveer un final. De forma que pondremos en ella un 0, cuando queramos seguir investigando números y un 1, cuando queramos parar.

N = en esta variable almacenaremos el número a investigar.

Z = esta variable la utilizaremos igual que en el ejercicio anterior, para almacenar el resultado de la división.



- 18 Hallar el factorial de un número.

Lista de variables

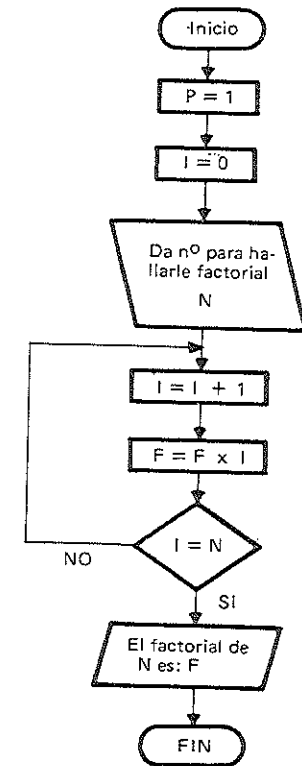
La fórmula para hallar el factorial de un número es:

$$N! = N (N-1) (N-2) \dots 4 3 2 1$$

N = en esta variable guardaremos el número al que le tenemos que hallar el factorial. Como no lo conocemos lo pediremos por pantalla.

I = esta variable valdrá 1, 2, 3, 4 ... (N-1). De forma que será la que utilizaremos para multiplicarla por si misma para conseguir el factorial.

F = en esta variable almacenaremos los productos acumulados al final del proceso. Contendrá el factorial del número.

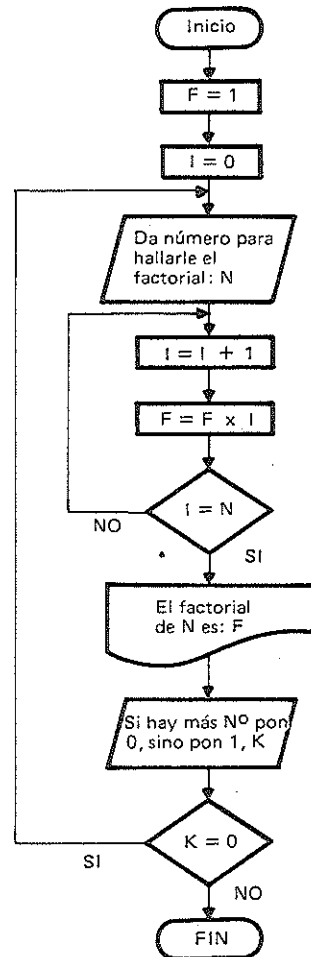


- 19) A continuación vamos a realizar el mismo ejercicio anterior pero en vez de hallar el factorial de un solo número, lo vamos a hacer para varios; preveemos un final.

Lista de variables

Las mismas del ejercicio anterior.

K = con esta variable vamos a preveer el final. Valdrá 0 si hay más números y 1 si no hay más números.



- 20) Dado un número y un tanto por ciento, contestar:

- El número y el tanto por ciento.
- La cantidad obtenida con el tanto por ciento de descuento y la cantidad a pagar.

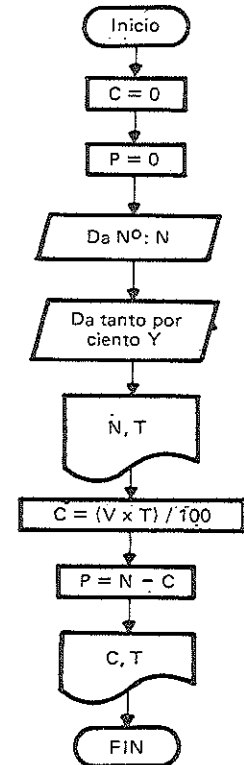
Lista de variables

N = en esta variable meteremos el número al que hay que hallarle el tanto por ciento. La pediremos por pantalla.

T = en esta variable meteremos el tanto por ciento. La pediremos por pantalla.

C = en esta variable meteremos la cantidad obtenida con el tanto por ciento.

P = en esta variable meteremos la cantidad a pagar.



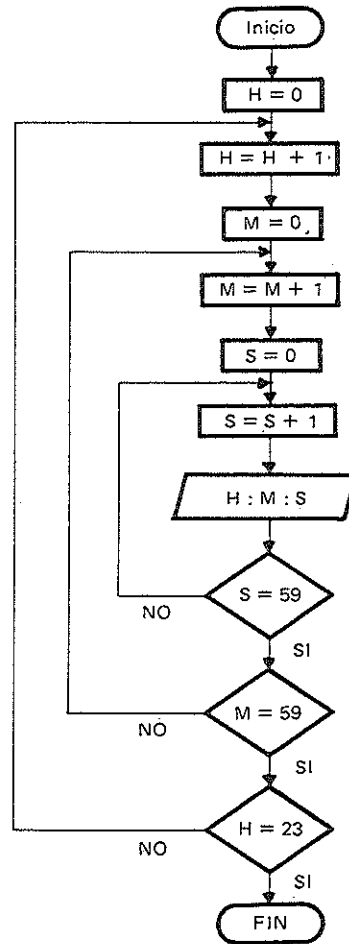
21) Hacer que el ordenador funcione como un reloj.

Lista de variables

H = esta variable va a ir generando las horas. Valdrá 1, 2, ... 23.

M = esta variable va a ir generando los minutos. Valdrá 1, 2, 3, 4, 5 ... 59.

S = esta variable va a ir generando los segundos. Valdrá: 1, 2, 3, 4, 5 ... 59.



22) Una empresa paga a sus cien operarios semanalmente de acuerdo con el número de horas que han trabajado, a razón de D ptas. por hora y de 1,5 D ptas. por todas aquellas horas que exceden de 40. Indicar el total de salario a percibir por cada uno de los operarios.

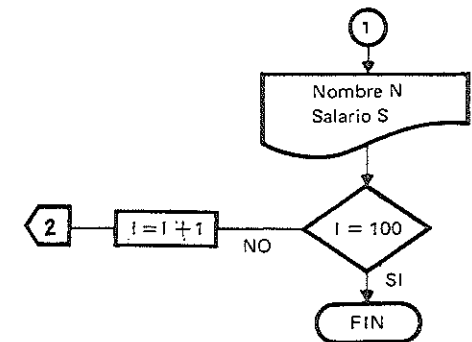
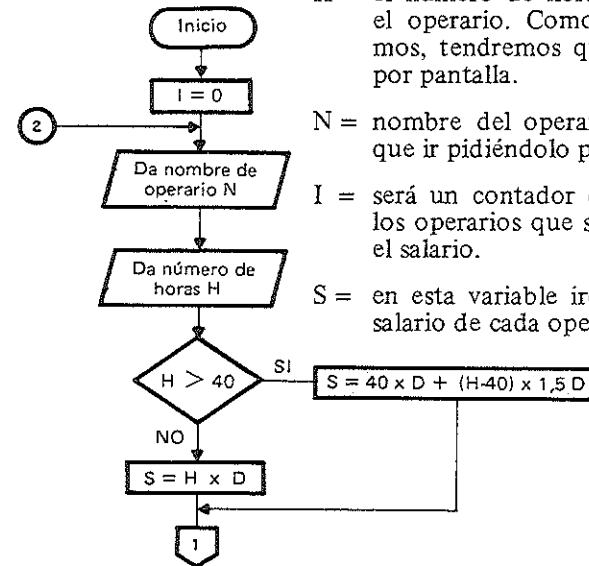
Lista de variables

H = el número de horas trabajadas por el operario. Como no las conocemos, tendremos que ir pidiéndolas por pantalla.

N = nombre del operario, también hay que ir pidiéndolo por pantalla.

I = será un contador que irá contando los operarios que se les va hallando el salario.

S = en esta variable iremos hallando el salario de cada operario.

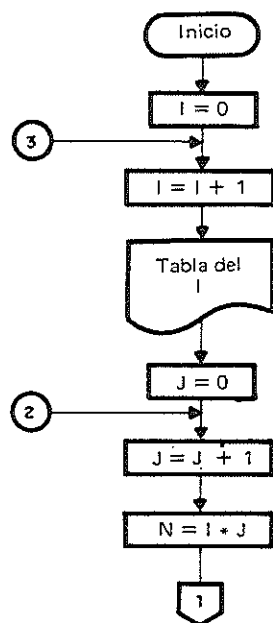
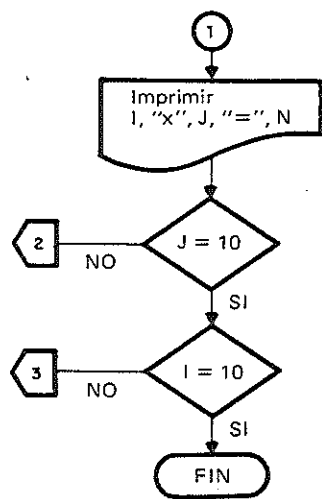


23 Programar las 10 primeras tablas de multiplicar.

Lista de variables

I = esta variable me indicará la tabla en que estoy. La tabla del 1, la del 2, etc.

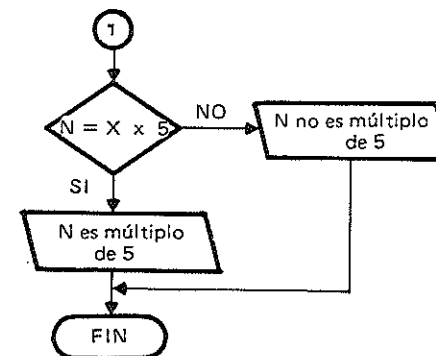
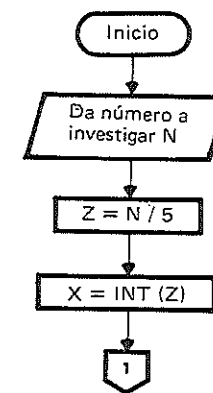
J = esta variable me indica en qué número de una tabla estoy.



24 Dado un número, contestar si es múltiplo de cinco.

Lista de variables

Z = Para que un número sea múltiplo de cinco, tiene que ser divisible por cinco. En la variable Z, vamos a guardar el resultado de dividir dicho número por 5. Y a continuación cogemos la parte entera de Z y la guardaremos en X.



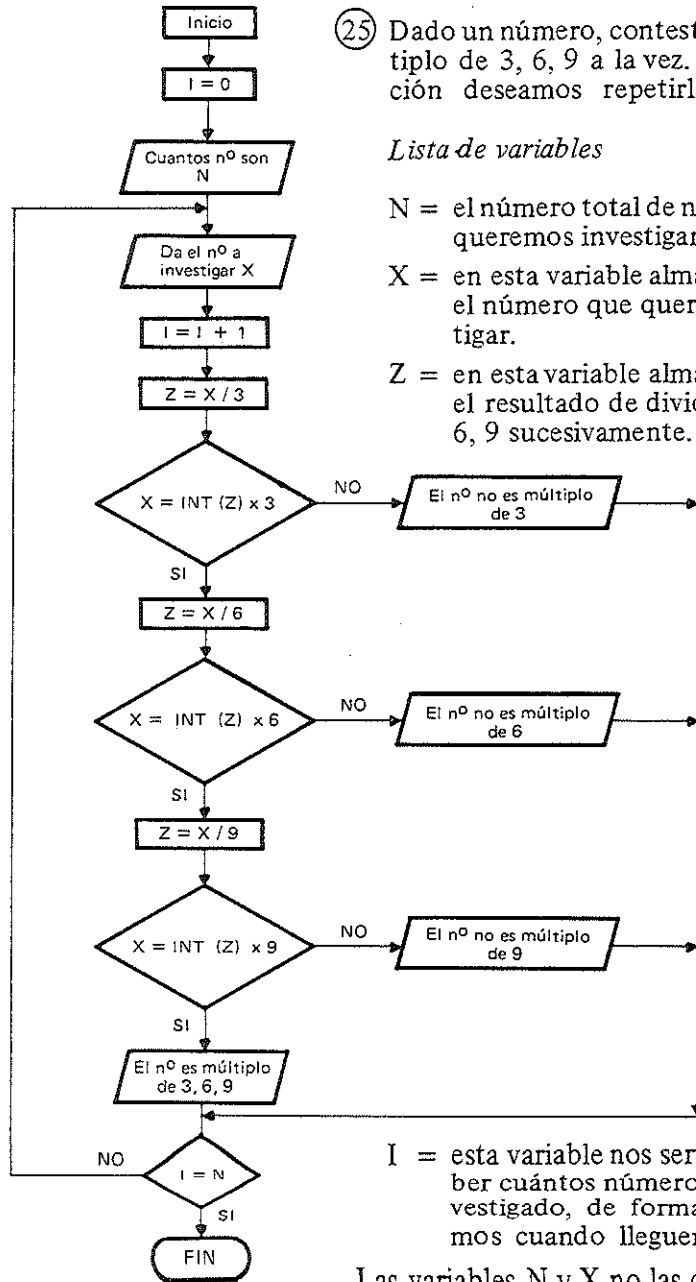
25) Dado un número, contestar si es múltiplo de 3, 6, 9 a la vez. Esta operación deseamos repetirla N veces.

Lista de variables

N = el número total de números que queremos investigar.

X = en esta variable almacenaremos el número que queremos investigar.

Z = en esta variable almacenaremos el resultado de dividir N por 3, 6, 9 sucesivamente.



I = esta variable nos servirá para saber cuántos números hemos investigado, de forma que paremos cuando lleguemos a 100.

Las variables N y X no las conocemos, luego las tendremos que pedir en pantalla.

26) Realizar el organigrama para resolver una ecuación de segundo grado, teniendo en cuenta las soluciones imaginarias. La fórmula de una ecuación de segundo grado es:

$$Ax^2 + Bx + C = 0$$

$$X_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$$

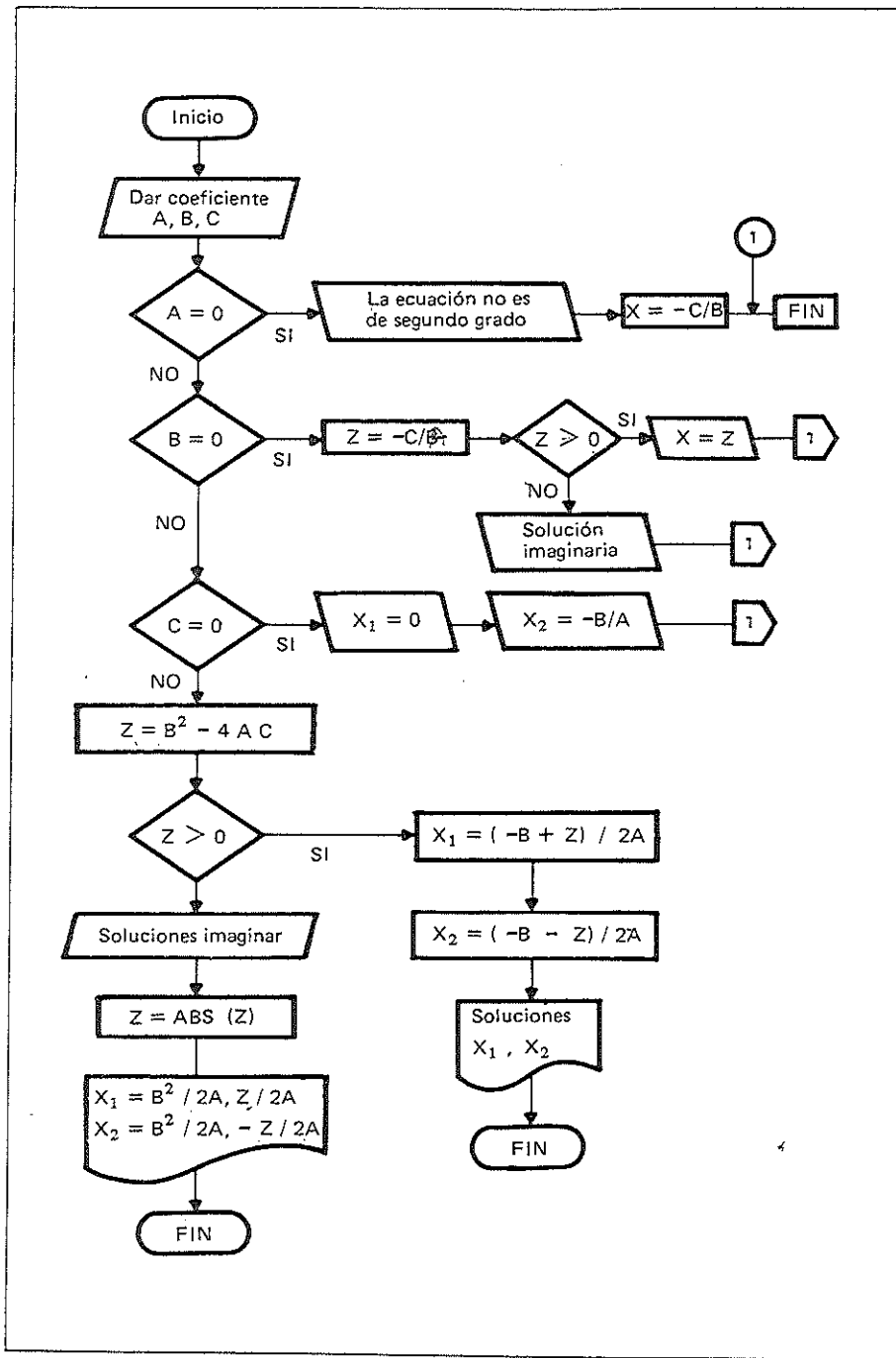
$$X_2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A}$$

Lista de variables
 Si: $A = 0$ $x = -C/B$
 $B = 0$ $Z = \frac{-C}{A}$ si $Z > 0$ $x = \sqrt{Z}$
 $C = 0$ $Ax^2 + Bx = 0$
 $x(A + B/A) = 0$

A, B, C = serán los coeficientes de x^2 , x y el término independiente respectivamente.

Z = en esta variable guardaremos el resultado de la operación $B^2 - 4AC$, pues nos interesará para ver si las soluciones son imaginarias o no. Si la variable Z es mayor que cero, las soluciones son reales; si la variable es menor que cero, las soluciones son imaginarias.

Para realizar el estudio de la ecuación más profundamente, tendremos en cuenta la posibilidad de que alguno de los coeficientes sea cero.



27) Hacer un organigrama que realice el horóscopo.

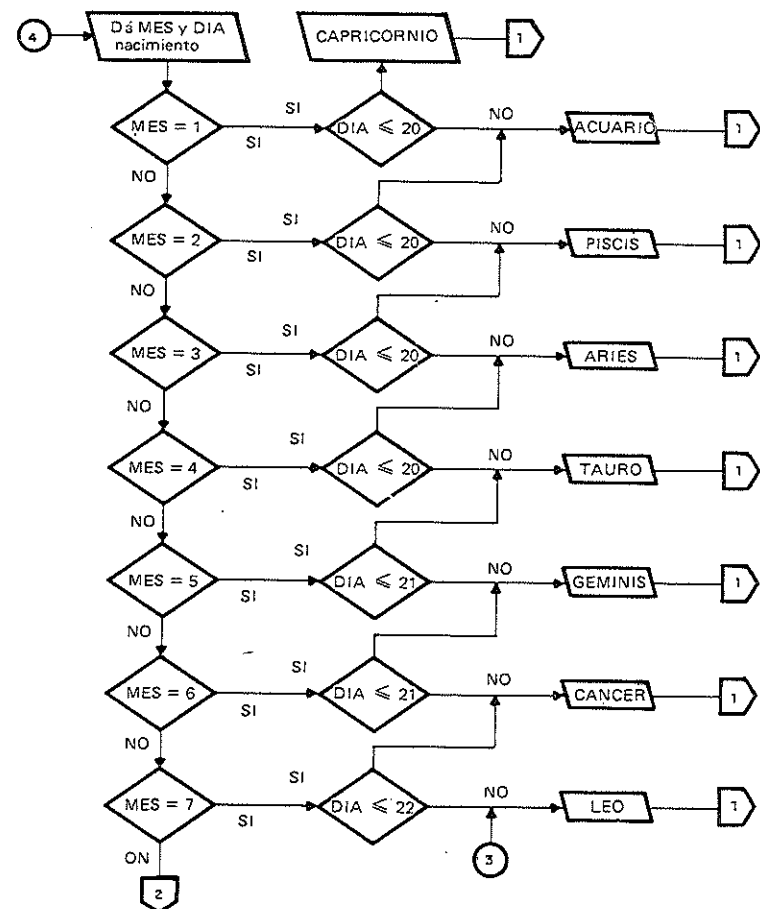
Lista de variables

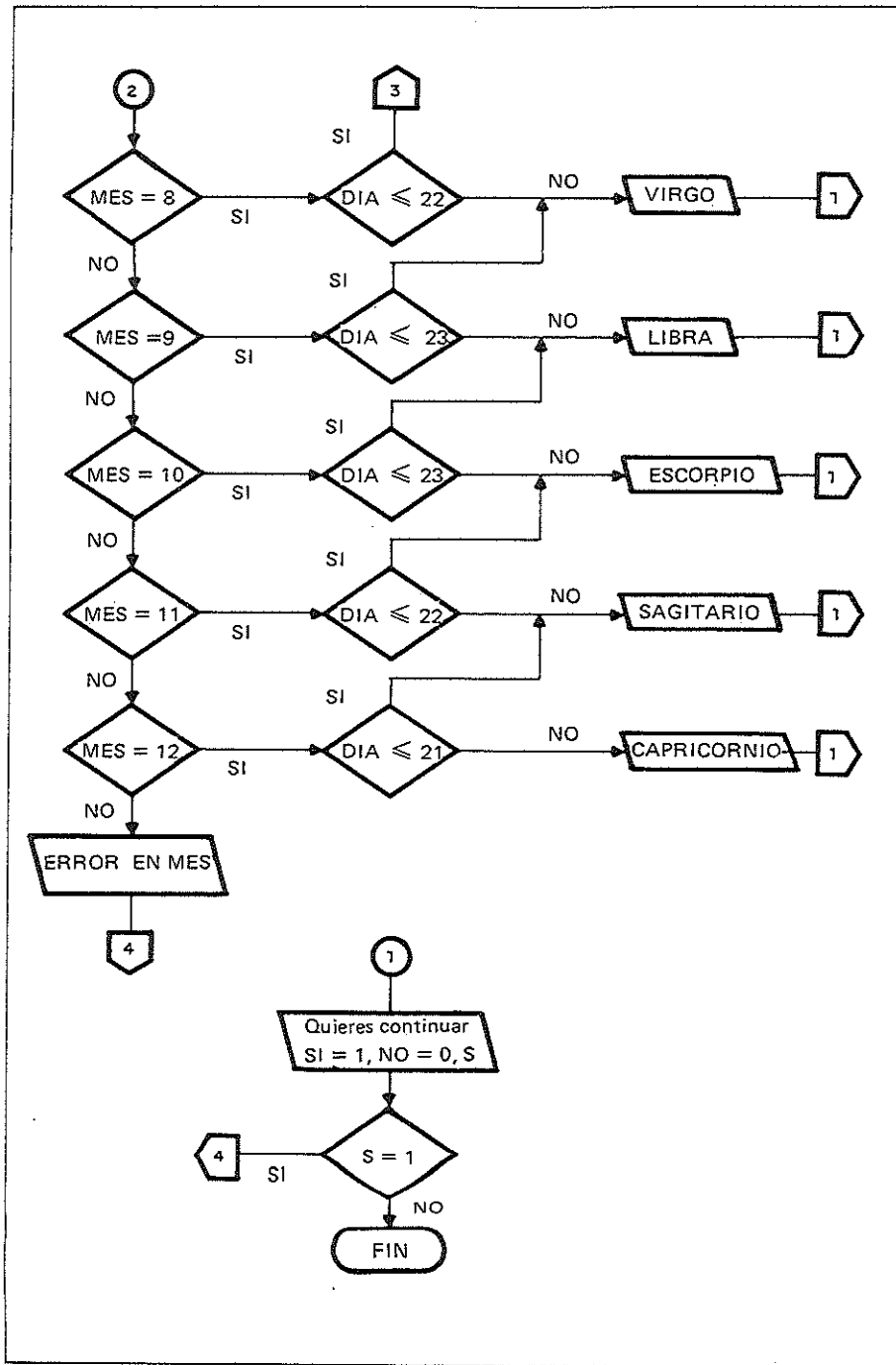
Mes = esta variable servirá para guardar el mes de nacimiento.

Día = servirá para guardar el día de nacimiento.

S = en esta variable guardaremos un 1, si queremos seguir investigando fechas y un 0 si no queremos seguir.

MES 0:0
DIA 20





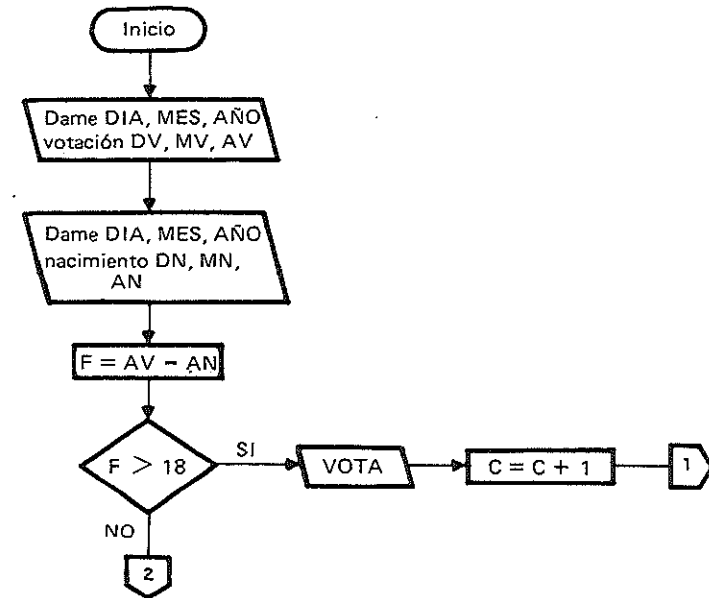
28) Vamos a realizar un sondeo para las votaciones, de forma que salga un mensaje de vota o de no vota, dependiendo de que se tenga dieciocho años o no, contando los que votan. Pedimos la fecha de la votación y la fecha de nacimiento de cada individuo.

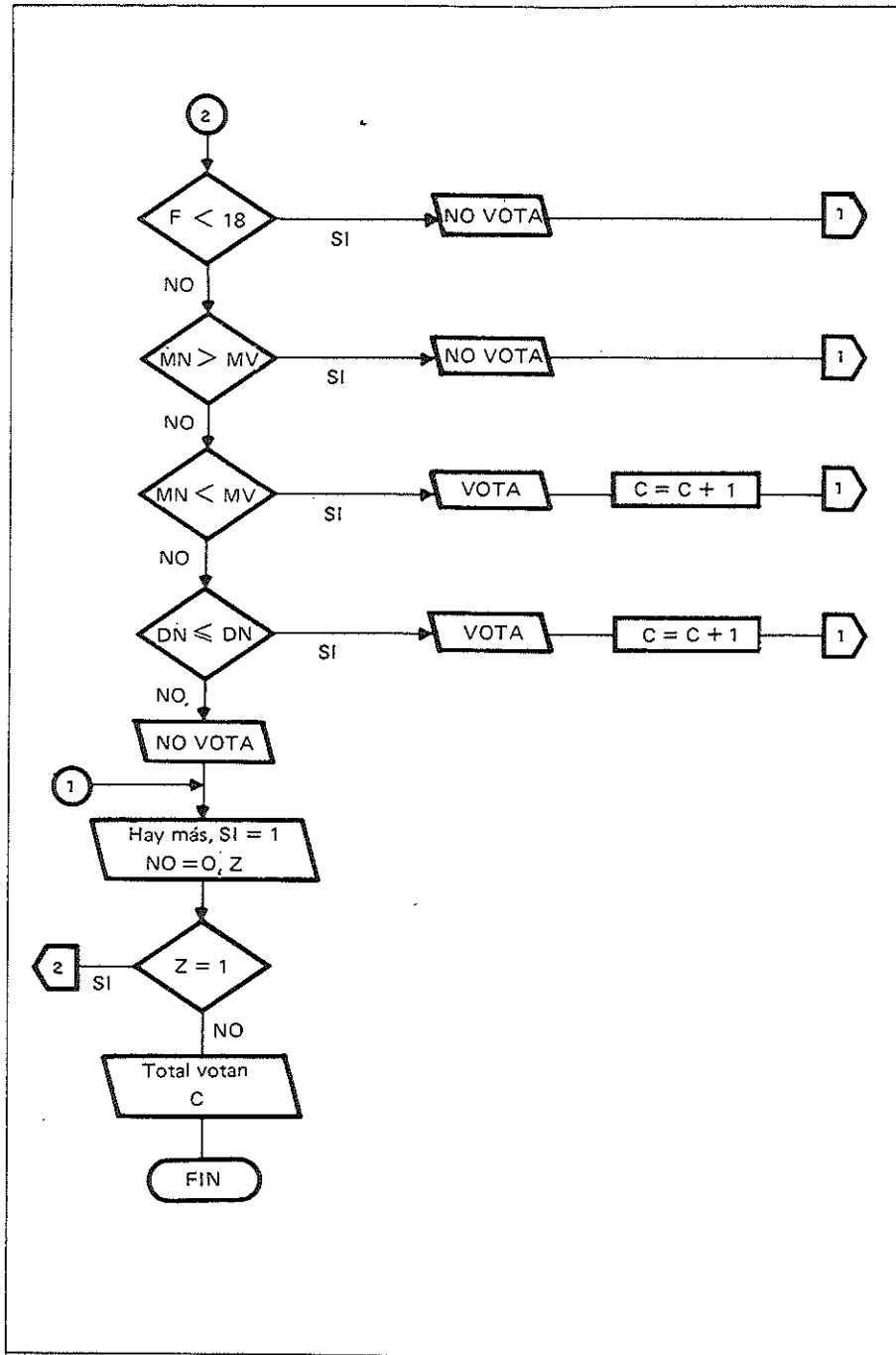
Lista de variables

AV, MV DV = estas variables contendrá el año de la votación, el mes y el día respectivamente.

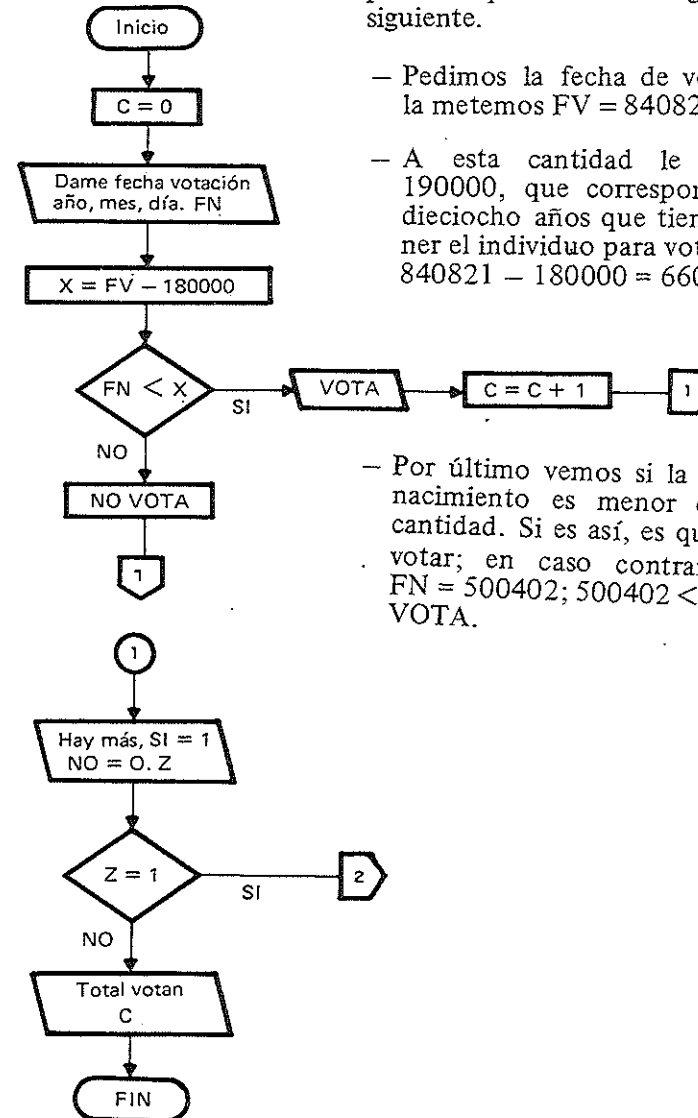
AN, MN, DN = estas variables contendrá el año de nacimiento, el mes y el día respectivamente.

F = esta variable la utilizamos para guardar la diferencia entre el año de nacimiento y el año de votación





29) A continuación vamos a realizar el mismo organigrama anterior, pero utilizando otro método. Es decir, vamos a considerar las fechas como números, pidiendo primeramente el año, a continuación el mes y después el día, y metiéndolo todo en una misma variable. El proceso que vamos a seguir es el siguiente.



– Pedimos la fecha de votación y la metemos $FV = 840821$.

– A esta cantidad le restamos 190000, que corresponde a los dieciocho años que tiene que tener el individuo para votar:
 $840821 - 180000 = 660821$.

– Por último vemos si la fecha de nacimiento es menor que esta cantidad. Si es así, es que puede votar; en caso contrario, no:
 $FN = 500402$; $500402 < 660821$
 VOTA.

30) A través del ordenador queremos hacer la siguiente encuesta:

- Tienen televisión o no la tienen. Contar los que la tienen y los que no la tienen.
- Es en color o no es en color. Contarlos.
- Piensan comprarla o no. Contarlos.
- Hallar los tantos por ciento de todo.

Lista de variables

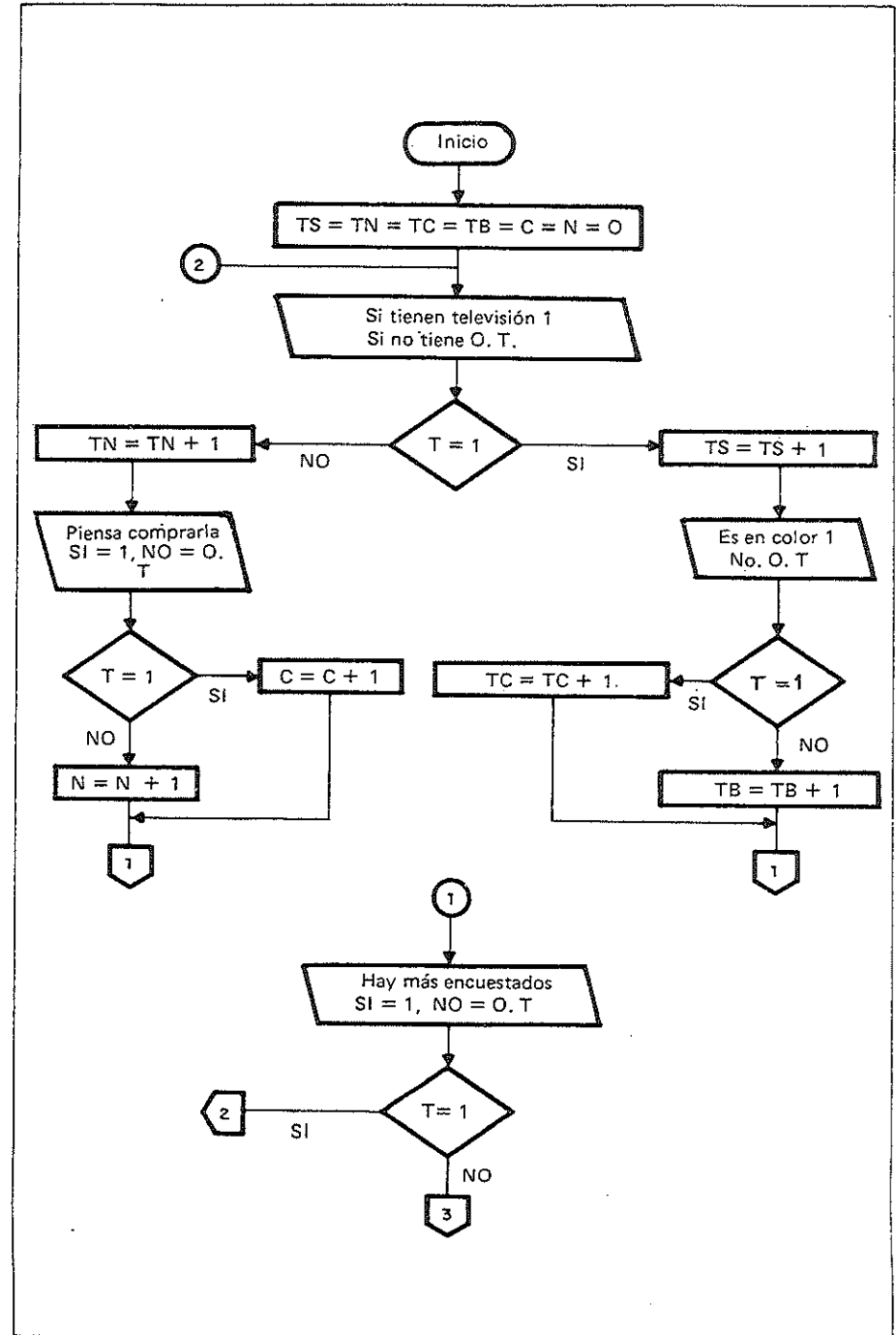
TS = en esta variable contamos los que tienen televisión.

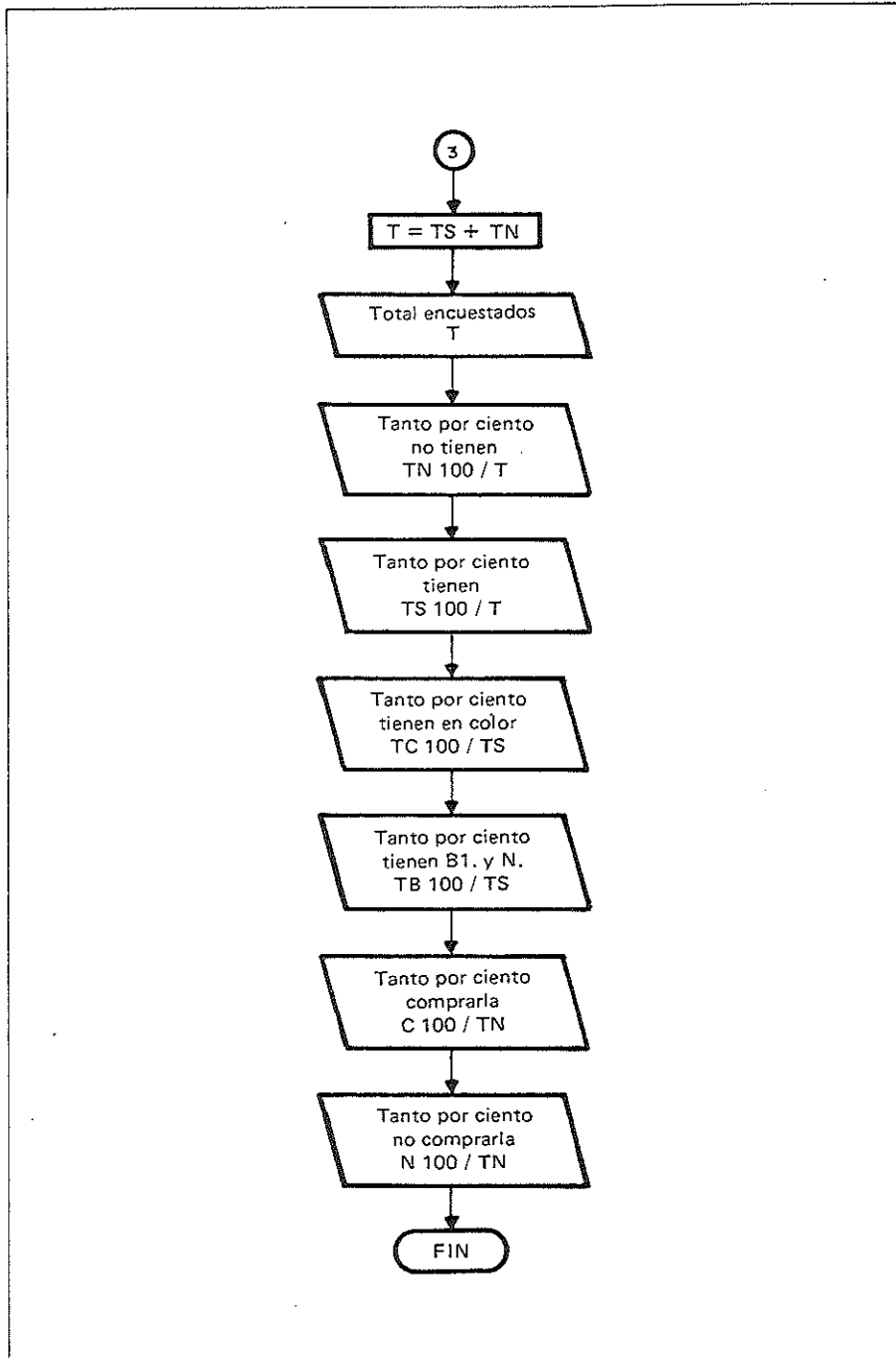
TN = en esta variable contamos los que no tienen televisión.

TC = en esta variable contamos los que no tienen televisión color.

C = en esta variable contamos los que no la tienen pero piensan comprarla.

N = en ésta contamos los que no piensan comprarla.



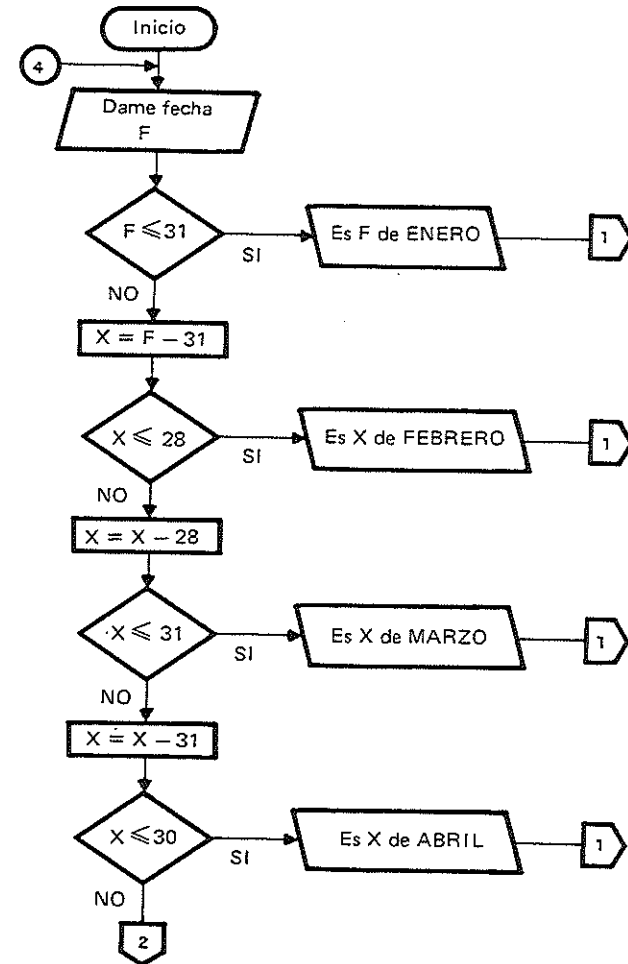


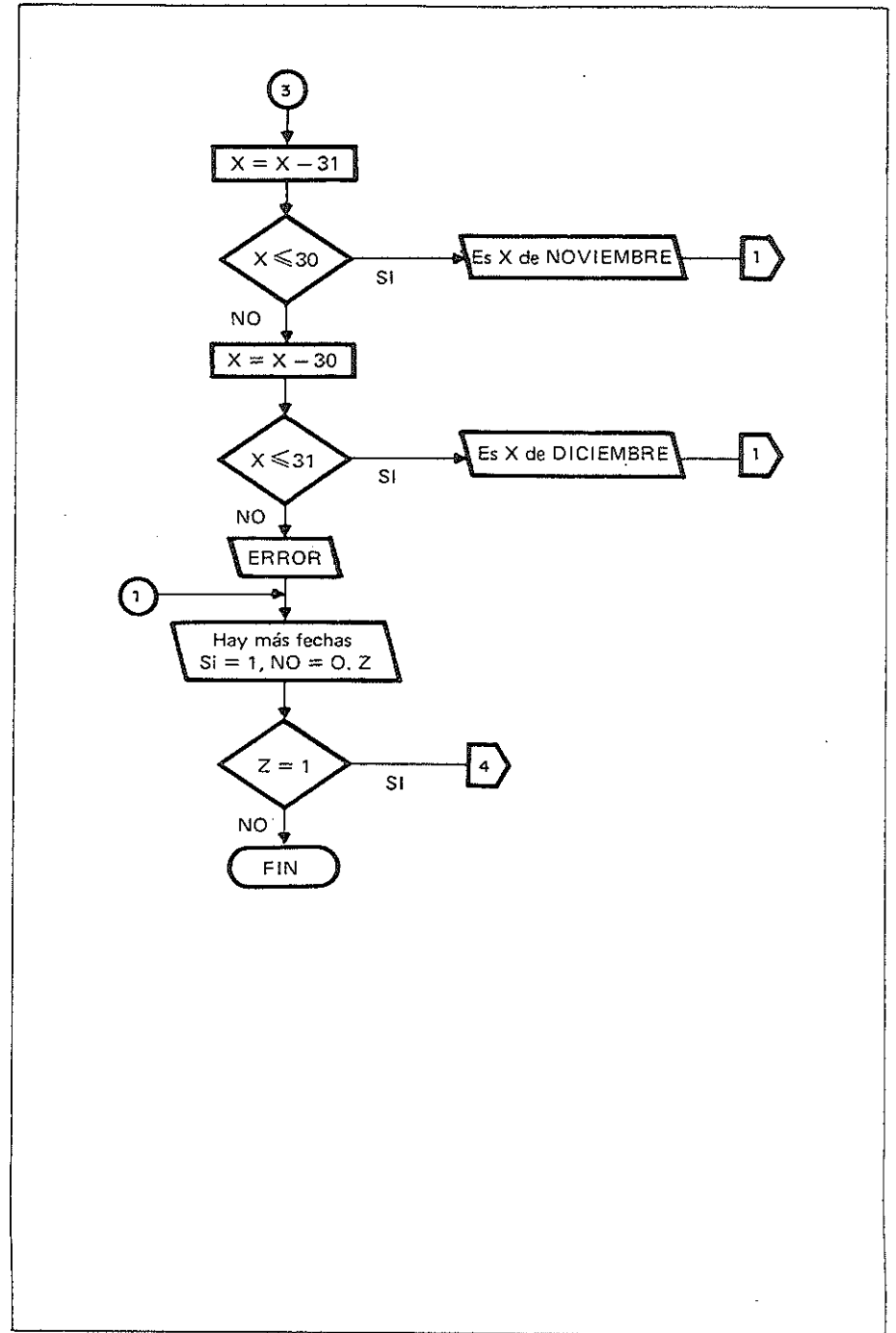
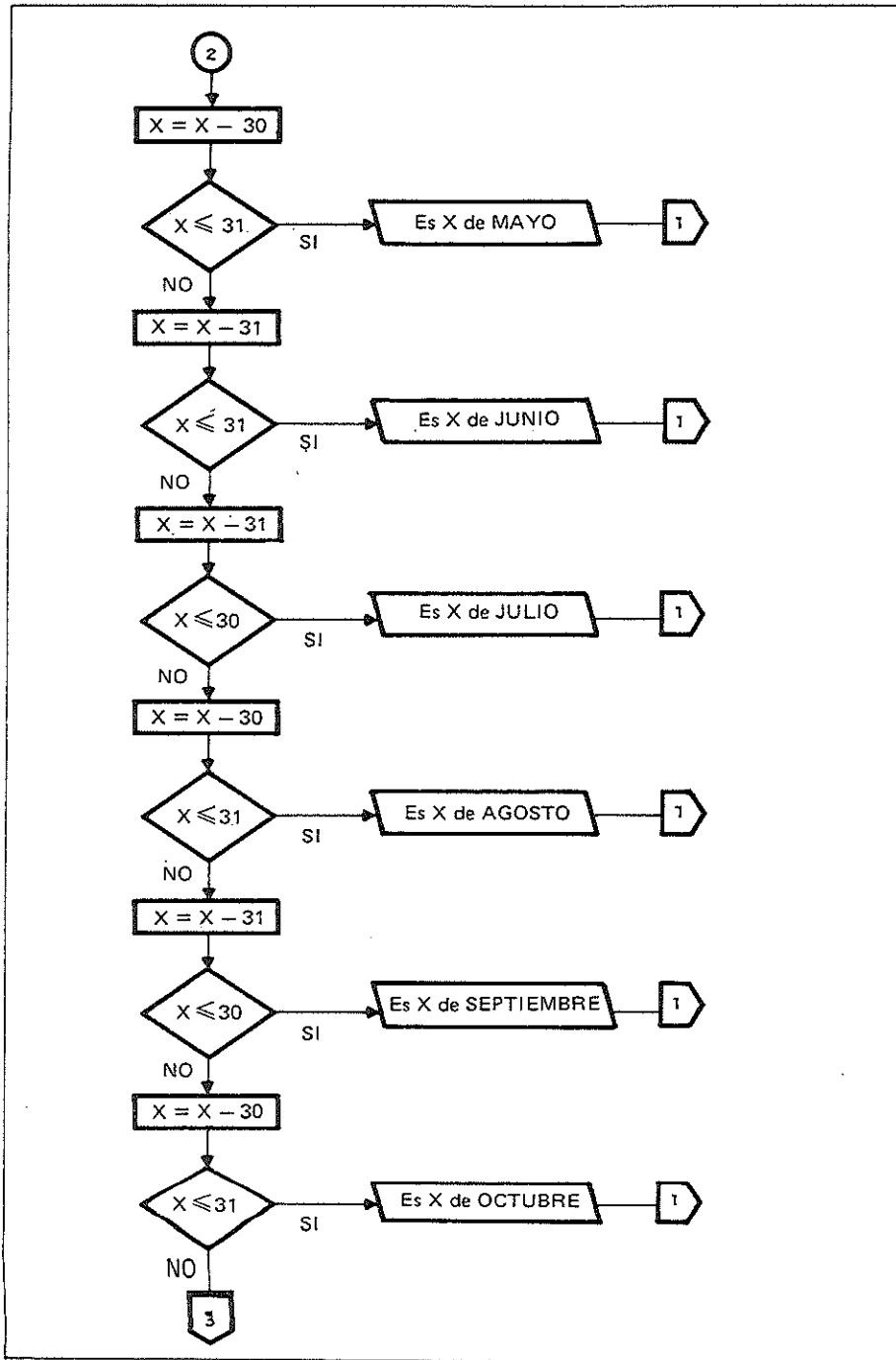
- 31) Dada una fecha, juliana, contestar con la fecha que le corresponde.

Lista de variables

F = en esta variable meteremos la fecha juliana.

X = en esta variable meteremos el día del mes que le corresponda.





3

TRABAJANDO
CON TABLAS

Hasta ahora sólo hemos visto ejemplos con variables simples, pero existen otras variables complejas, las cuales se utilizan para guardar una serie de valores, relacionados entre sí.

Estas variables se denominan variables subindexadas: se utilizan para guardar una serie de valores en una sola variable, de tal forma que un conjunto de números enteros, denominados subíndices, nos indican la situación de cada uno de dichos valores.

Por ejemplo, supongamos que tenemos una lista de números, como: 1, 5, 7, 32, 40. Dichos números los guardaremos en una variable denominada A(I), de tal forma que en la posición:

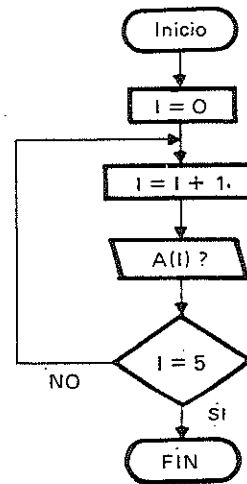
- En la posición A(1) se guardará el elemento 1.
- En la posición A(2) se guardará el valor 5.
- En la posición A(3) se guardará el valor 7.
- En la posición A(4) se guardará el valor 32.
- Y en la posición A(5) se guardará el valor 40.

También puede ser una variable subindexada alfanumérica, de tal forma que su contenido será una lista de valores alfanuméricos, por ejemplo una serie de nombres.

Con esta lista de valores o variable subindexada, se pueden realizar una serie de operaciones:

1. *Se debe crear la lista*, es decir, se deben de almacenar los elementos en sus posiciones. Esta operación se realizará por medio de un bucle. El organismo correspondiente será:

Petición de valor para A(I), por medio del teclado y la pantalla.

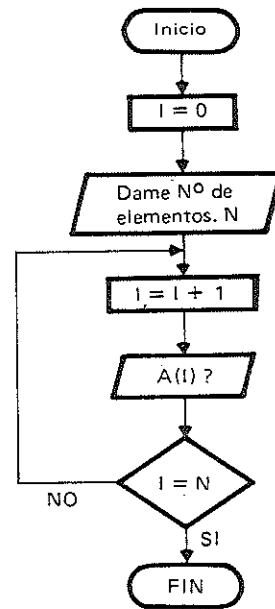


Se puede crear secuencialmente, es decir, guardando los elementos con un orden ascendente o descendente, bien aleatoriamente, es decir, sin seguir ningún orden, guardando los elementos según van llegando.

2. *Inserción de un nuevo elemento*: consiste en añadir un nuevo elemento a la lista.
3. *Eliminación de un elemento*: consiste en borrar un elemento de la lista que ya no nos interesa.
4. *Búsqueda de un elemento determinado*: esta búsqueda se puede hacer secuencialmente, es decir, ir leyéndolos todos hasta encontrar el elemento que nos interesa, o bien por el método de las dicotomías o búsqueda binaria, el cual veremos más adelante con un ejemplo.
5. *Modificaciones*: consiste en acceder a un elemento de la lista para modificar su contenido.
6. *Clasificación*: consiste en ordenar una lista que esté desordenada, en un orden ascendente o descendente.

A continuación veremos una serie de ejemplos con variables complejas.

32 Crear una lista en memoria para almacenar diez elementos

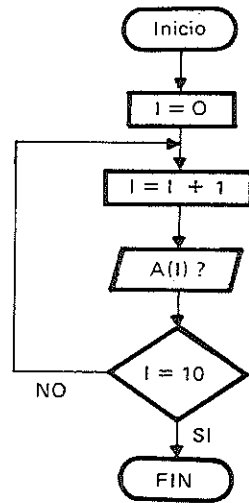


33 Crear una lista en memoria para N elementos.

Lista de variables

N = en esta variable se guardarán el número total de elementos que componen la tabla. Como no lo conocemos pediremos su valor en pantalla.

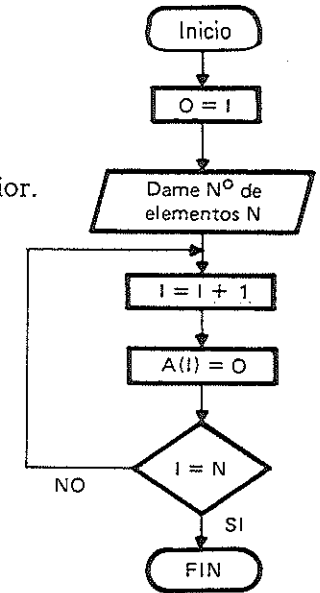
I = será el subíndice que va indicando qué elemento estoy pidiendo.



34 Crear una lista en memoria para N elementos, rellenándola con ceros.

Lista de variables

Las mismas que el ejercicio anterior.

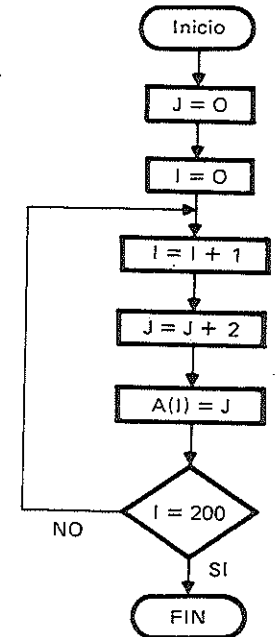


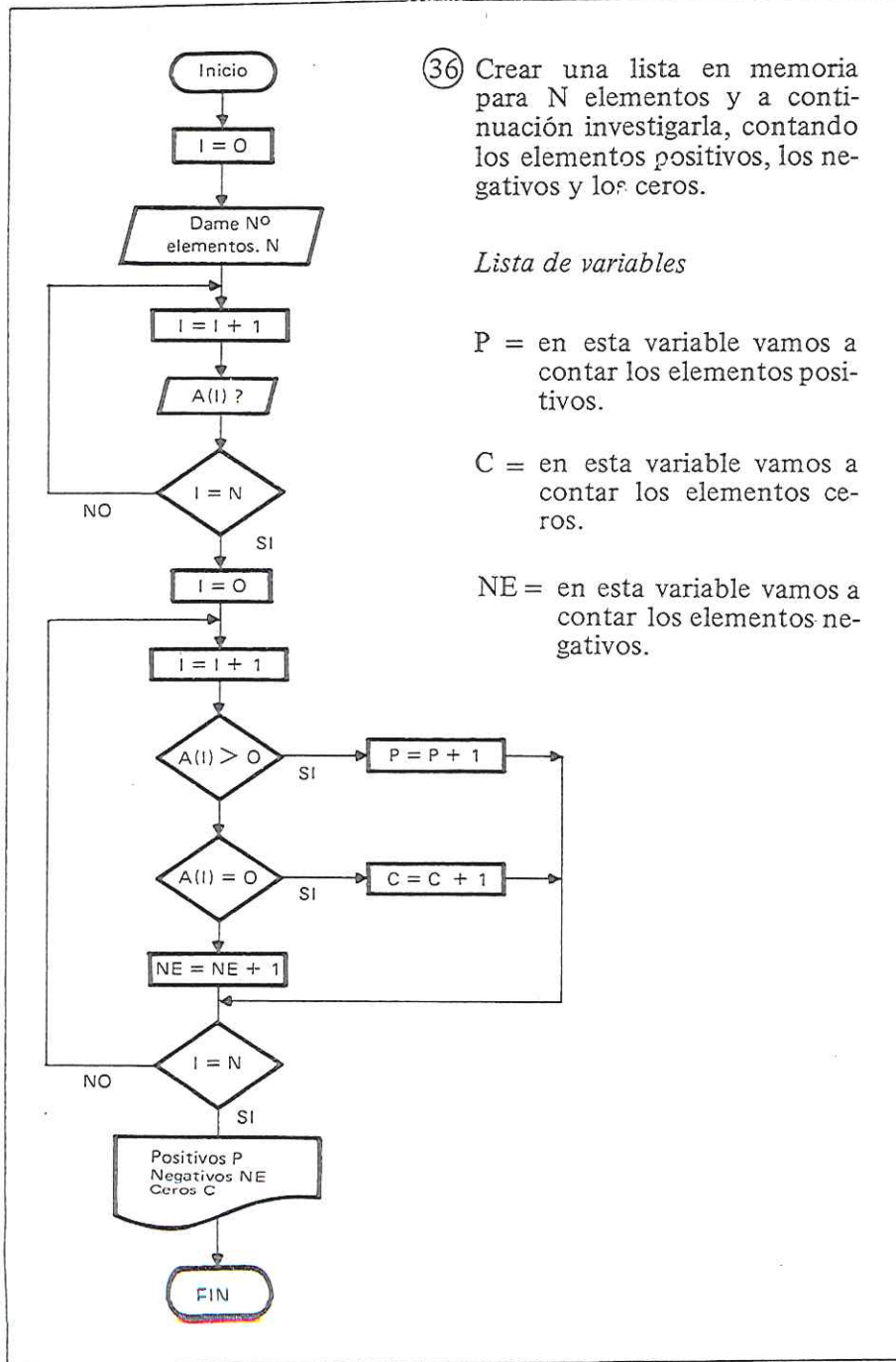
35 Crear una tabla en memoria para 200 elementos, y rellenarla con los números pares.

Lista de variables

I = será el subíndice de la tabla, y me irá indicando por qué elemento de la tabla voy.

J = esta variable la voy a utilizar para generar los números pares, es decir, valdrá 2, 4, 6, 8 etc.



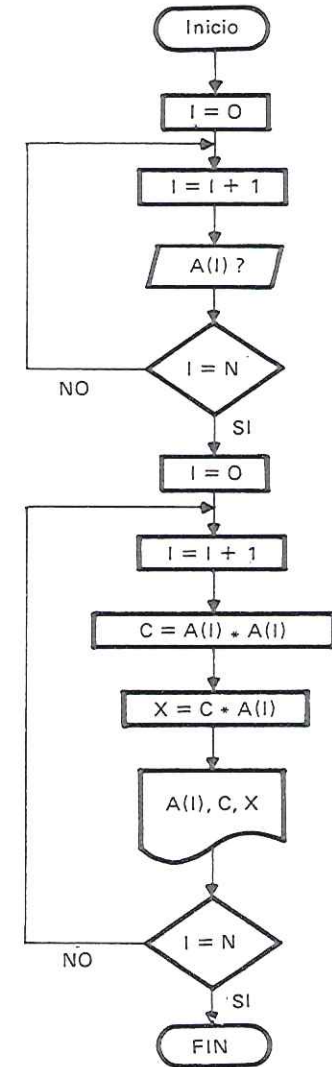


- 37) Crear una tabla en memoria para N elementos. A continuación se quiere tener un listado con todos los elementos, sus cuadrados y sus cubos.

Lista de variables

C = en esta variable guardaremos el cuadrado del elemento.

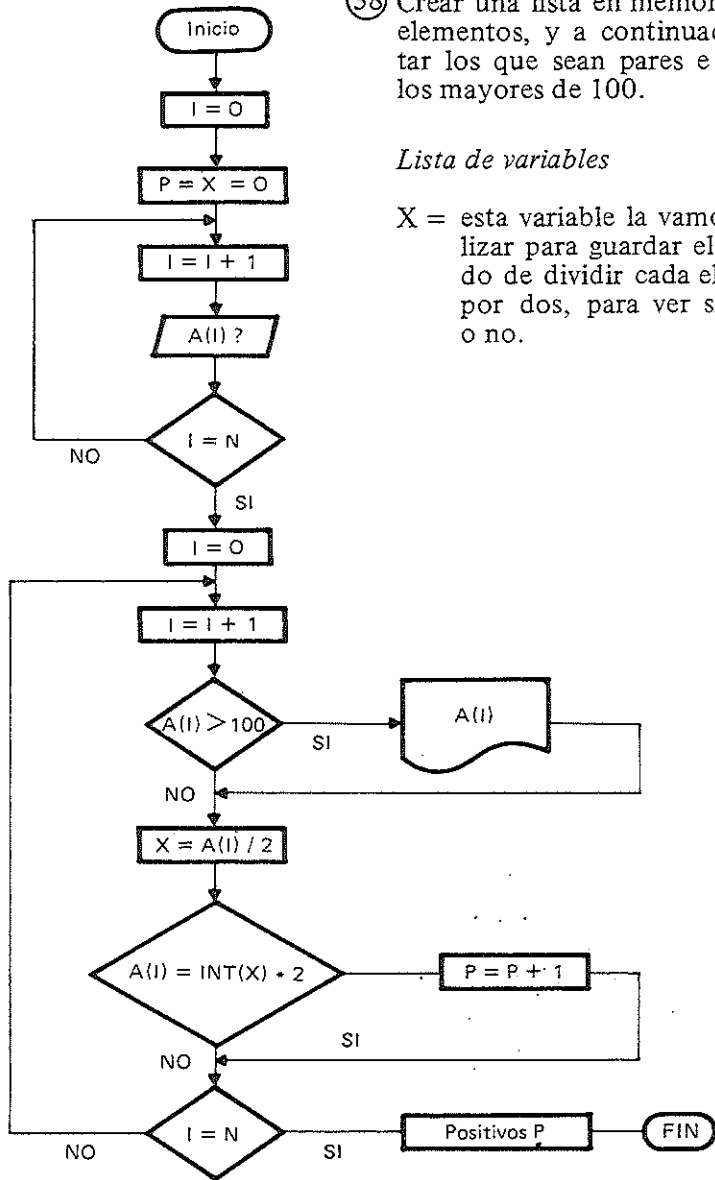
X = en esta variable guardaremos el cubo del elemento.



- 38 Crear una lista en memoria para N elementos, y a continuación contar los que sean pares e imprimir los mayores de 100.

Lista de variables

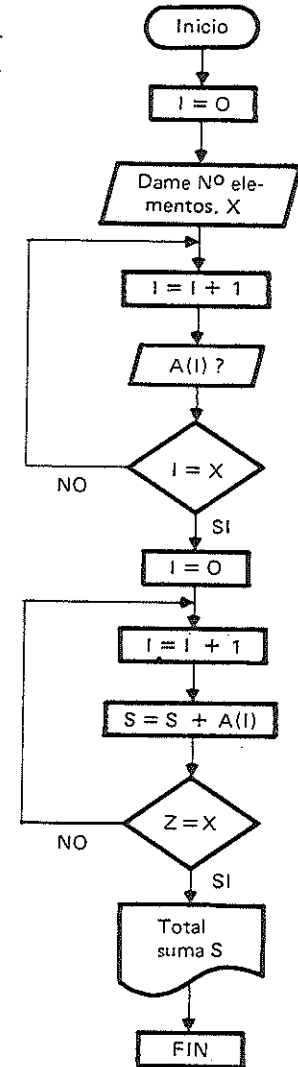
X = esta variable la vamos a utilizar para guardar el resultado de dividir cada elemento por dos, para ver si es par o no.



- 39 Crear una lista en memoria con X elementos. A continuación hallar la suma de los X elementos e imprimirla.

Lista de variables

S = en esta variable iremos guardando la suma acumulada de los X elementos.



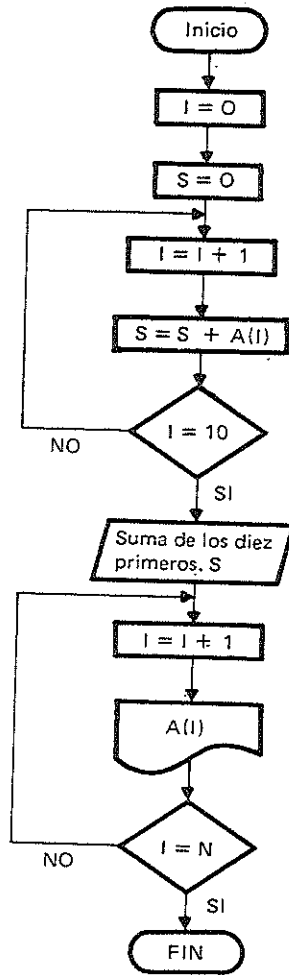
- 40) Dada una lista en memoria de N elementos, queremos investigarla, sumando los diez primeros elementos e imprimiendo los restantes. La tabla ya está creada.

Lista de variables

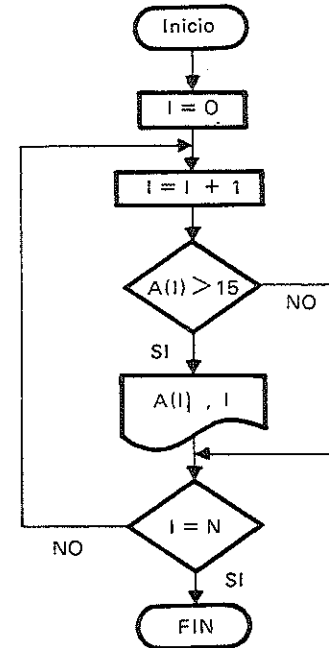
S = esta variable la utilizamos para almacenar la suma.

I = esta variable es el subíndice de la tabla.

N = es el número total de elementos de la tabla.



- 41) Dada una lista en memoria de N elementos, investigarla de forma que aparezcan en pantalla todos aquellos elementos mayores de 15, y el subíndice o lugar que ocupan en la lista.



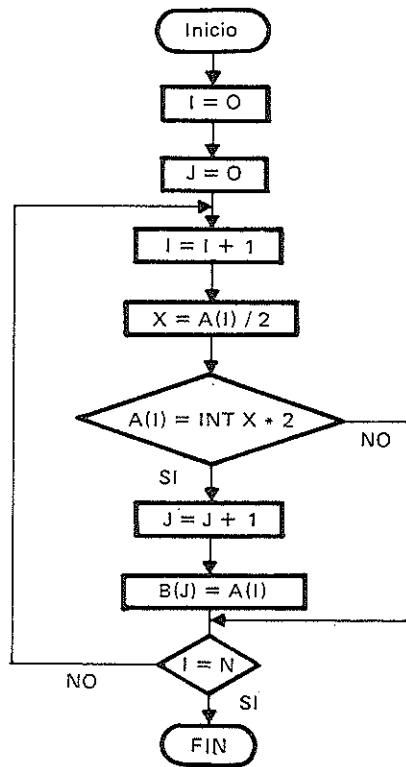
- ④2 Dada una lista en memoria de N elementos, investigarla de forma que vayamos metiendo en otra lista, todos aquellos elementos que sean pares.

Lista de variables

A(I) = ésta será la lista original.

B(J) = en esta otra lista iremos metiendo todos los elementos pares de la lista A(I).

X = esta variable contendrá el resultado de dividir cada elemento por dos, para ver si es par.



- ④3 Pedir cien valores por pantalla, repasarlo de forma que vayamos metiendo en distintas tablas los valores positivos y los negativos, y vayamos contando los ceros.

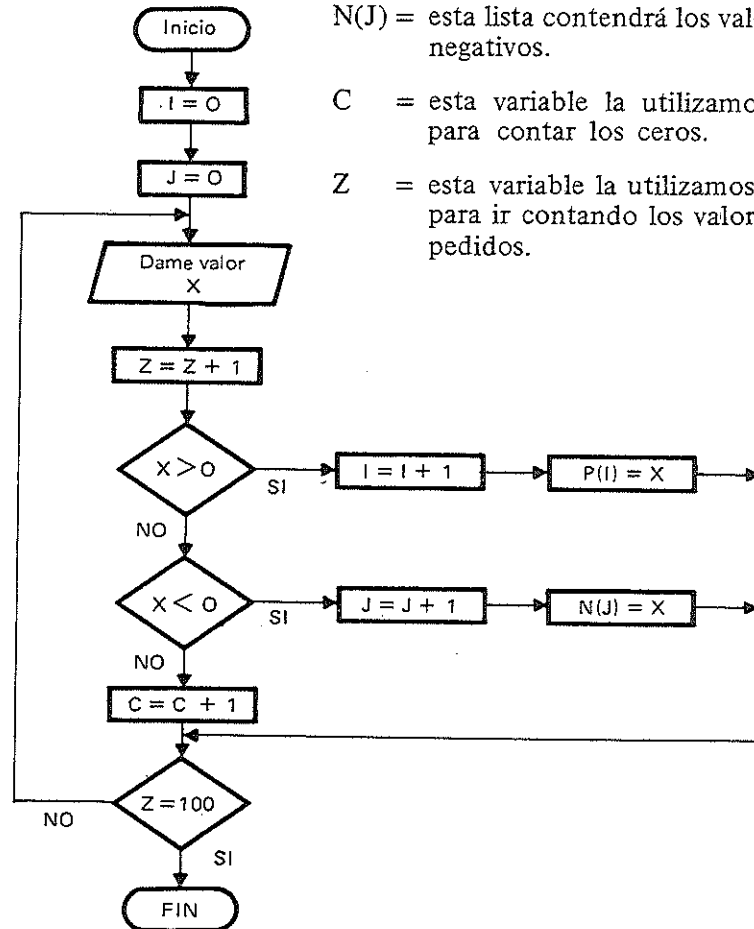
Lista de variables

P(I) = esta lista contendrá los valores positivos.

N(J) = esta lista contendrá los valores negativos.

C = esta variable la utilizamos para contar los ceros.

Z = esta variable la utilizamos para ir contando los valores pedidos.



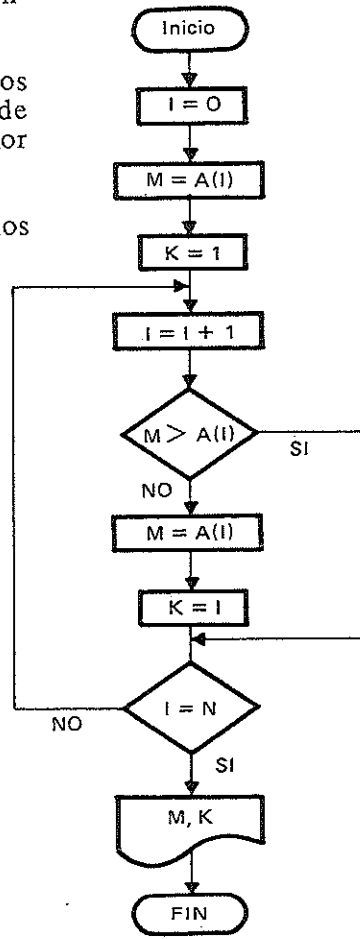
- 44 Dada una tabla en memoria de N elementos, queremos obtener el valor más grande y la posición donde se encuentra.

Lista de variables

M = en esta variable iremos metiendo el valor más grande que vamos encontrando.

K = en esta variable vamos metiendo el lugar donde se encuentra dicho valor mayor.

A(I) = es la tabla que tenemos en memoria.



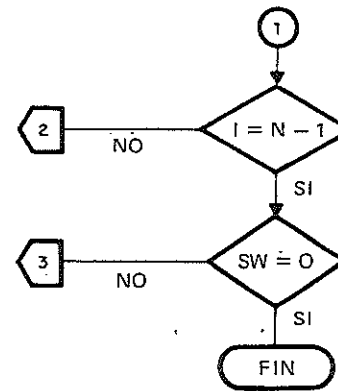
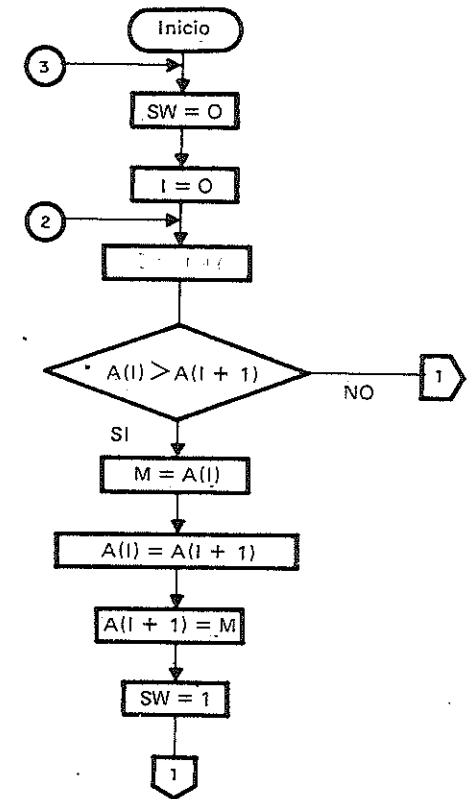
- 45 Dada una tabla en memoria de N elementos, ordenarla de menor a mayor.

Lista de variables

A(I) = lista a ordenar.

M = esta variable la vamos a utilizar para ir haciendo el cambio de elementos, el mayor con el menor.

SW = esta variable la pondremos a uno cuando se produzca un cambio y a cero cuando no se produzca un cambio, de tal forma que para saber si la tabla está ordenada, sólo tendremos que ver si esta variable está a cero.



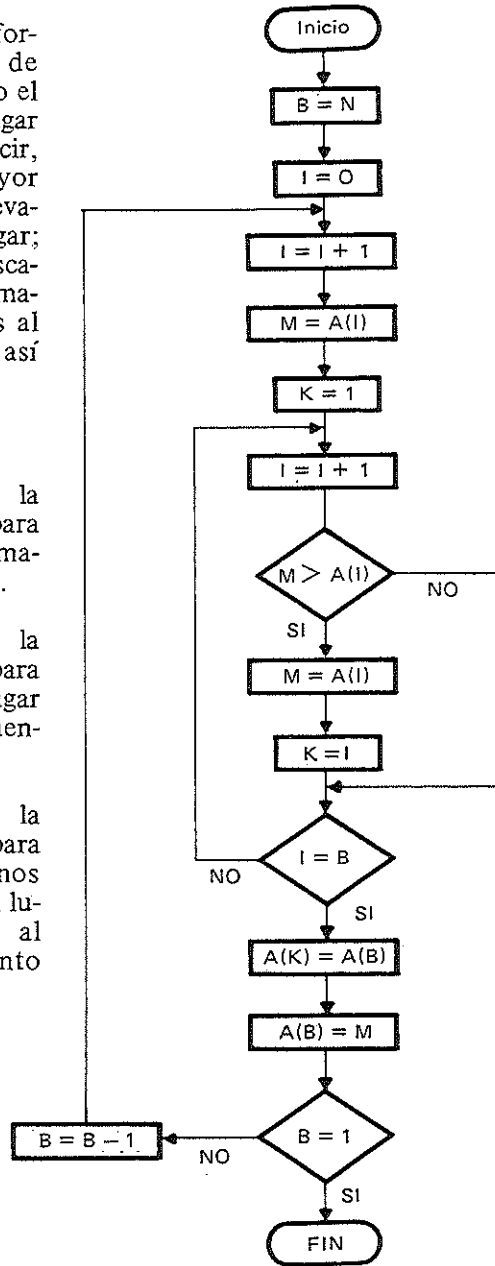
- 46) Vamos a ver otra forma de ordenación de una lista utilizando el intercambio en lugar del cambio. Es decir, buscamos el mayor de la lista y lo llevamos al último lugar; a continuación buscamos el siguiente mayor y lo llevamos al penúltimo lugar y así sucesivamente.

Lista de variables

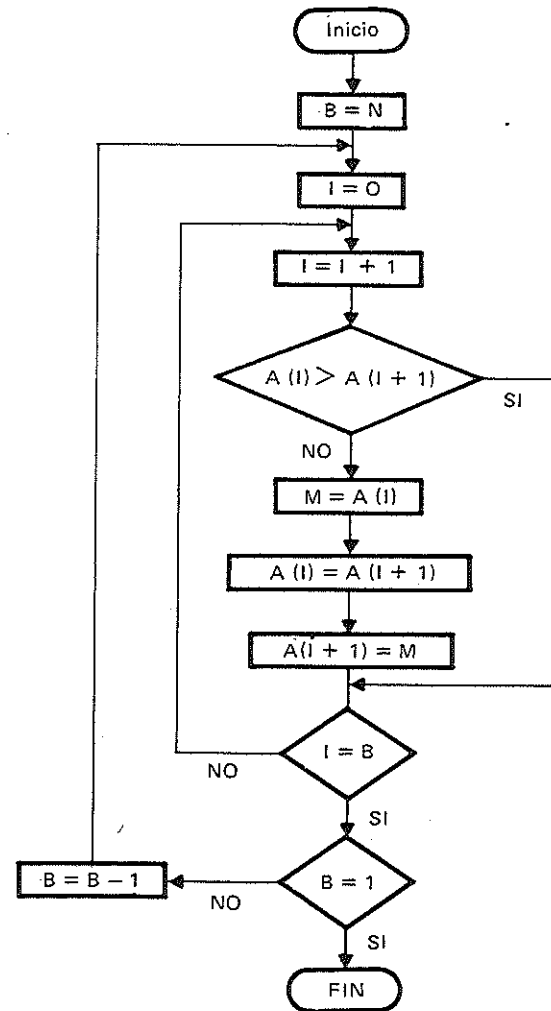
M = esta variable la utilizamos para guardar el mayor de la lista.

K = esta variable la utilizamos para guardar el lugar donde se encuentra el mayor.

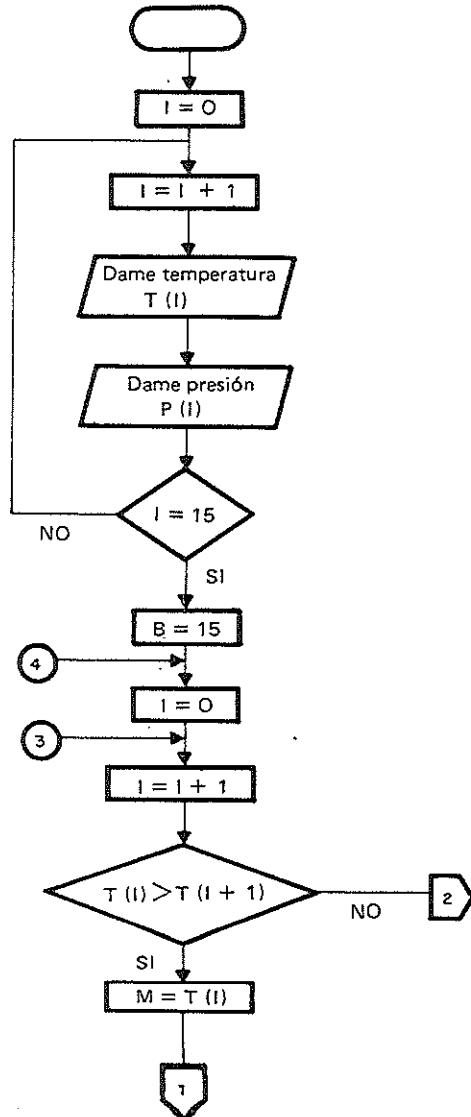
B = esta variable la utilizamos para ir colocándonos siempre en el lugar anterior al último elemento ordenado.



- 47) La tabla también la podemos ordenar de mayor a menor de esta forma.



- 48) Queremos crear dos listas, una con las temperaturas y otra con las presiones; cada lista tendrá 15 elementos. A continuación debemos ordenar las temperaturas en orden creciente, de forma que a la vez se vayan ordenando las presiones en el mismo orden.

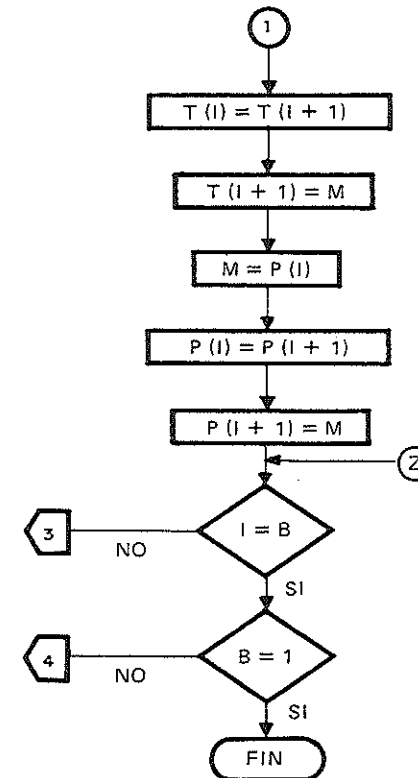


- 49) Dadas dos listas ordenadas, sin repeticiones ni lagunas, intercalarlas de forma que obtengamos una sola lista ordenada.

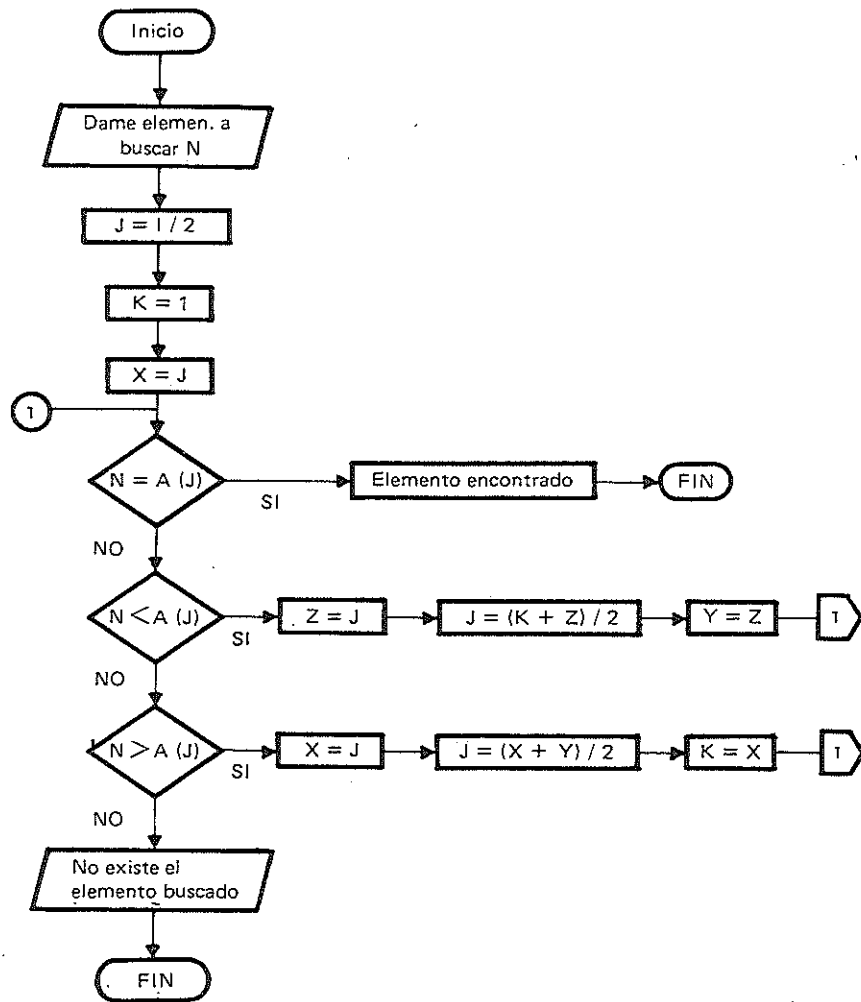
Lista de variables

A (I) = 1.^a lista ordenada. B (J) = 2.^a lista ordenada.
C (K) = Será la lista que obtenemos de la unión de A y B.

La lista A tiene N elementos, y la lista B, M elementos.



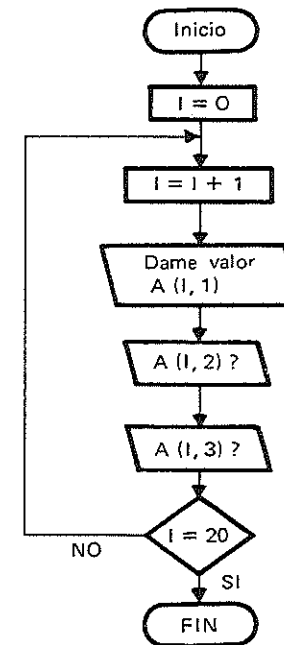
- 50) Dada una lista ordenada, realizar una búsqueda binaria de cualquier elemento pedido por teclado.



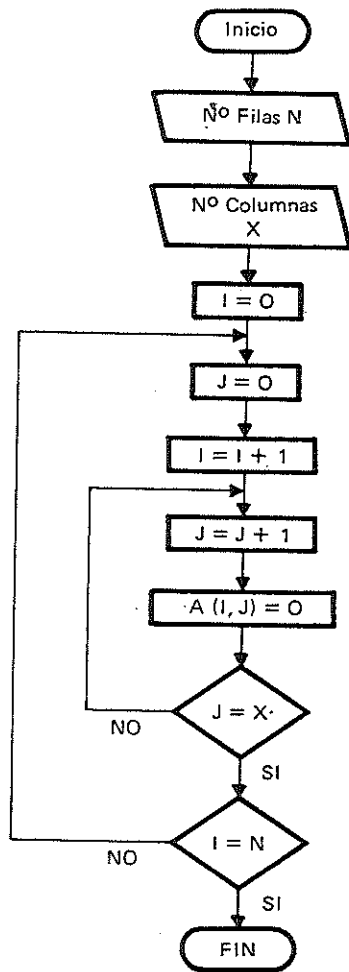
- 51) Crear una matriz de tres columnas y veinte filas.

Lista de variables

I = es el subíndice que nos indica las filas.



- 52 Crear una matriz de X columnas y N filas, rellenándolas a ceros.



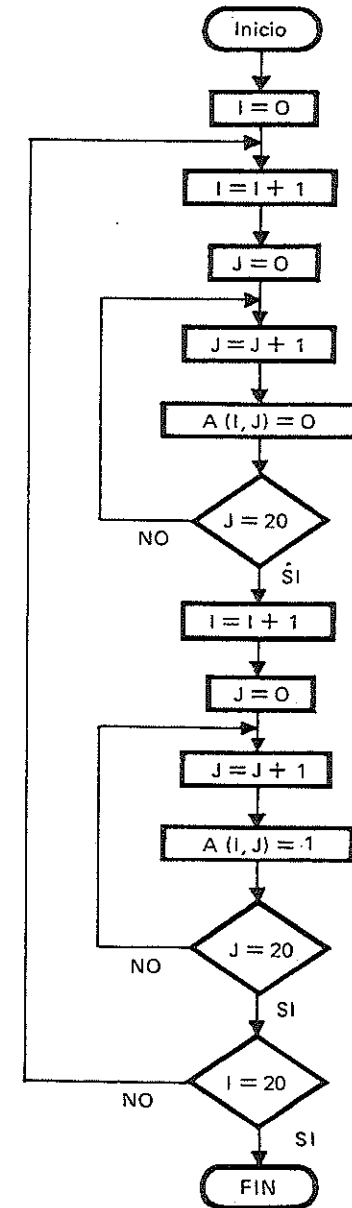
- 53 Generar una matriz de 20 filas y 20 columnas de forma que las filas pares se llenen a unos y las impares a ceros.

Lista de variables

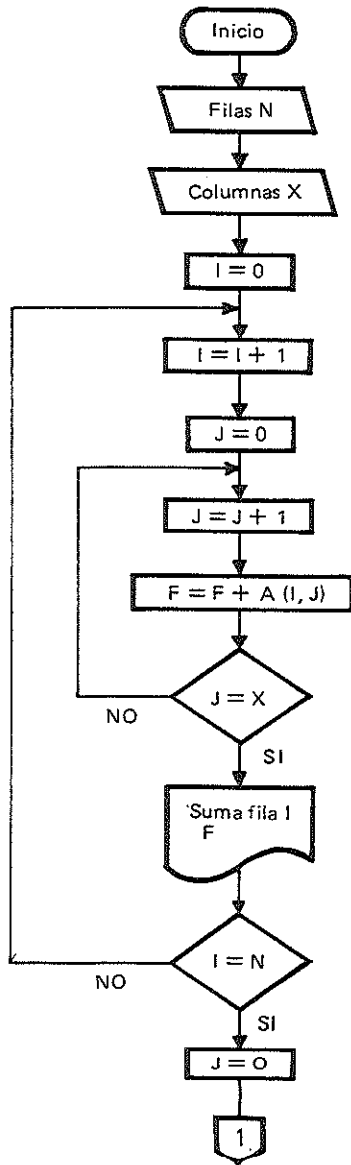
I = Subíndice de las filas.

J = Subíndice de las columnas.

g. par = 1
g. impar = 0



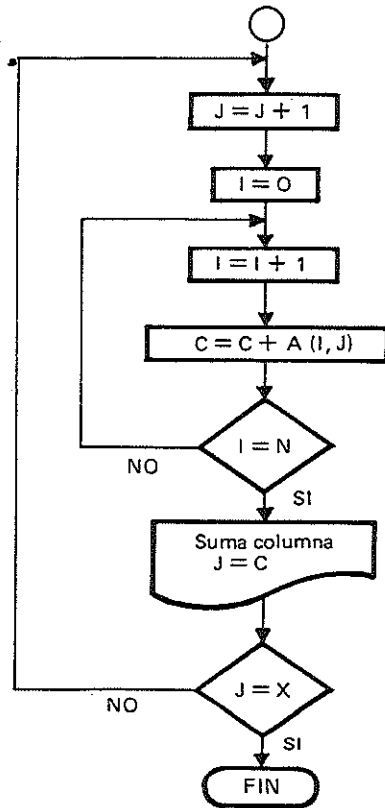
56) Sumar todos los elementos de cada fila de una matriz y todos los elementos de cada columna.



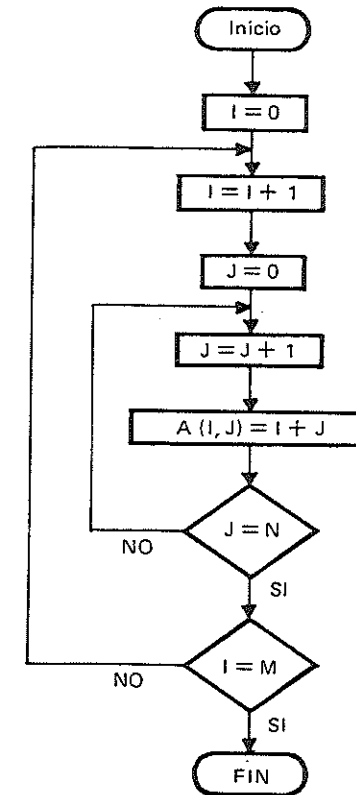
Lista de variables

F = Suma acumulada de cada fila

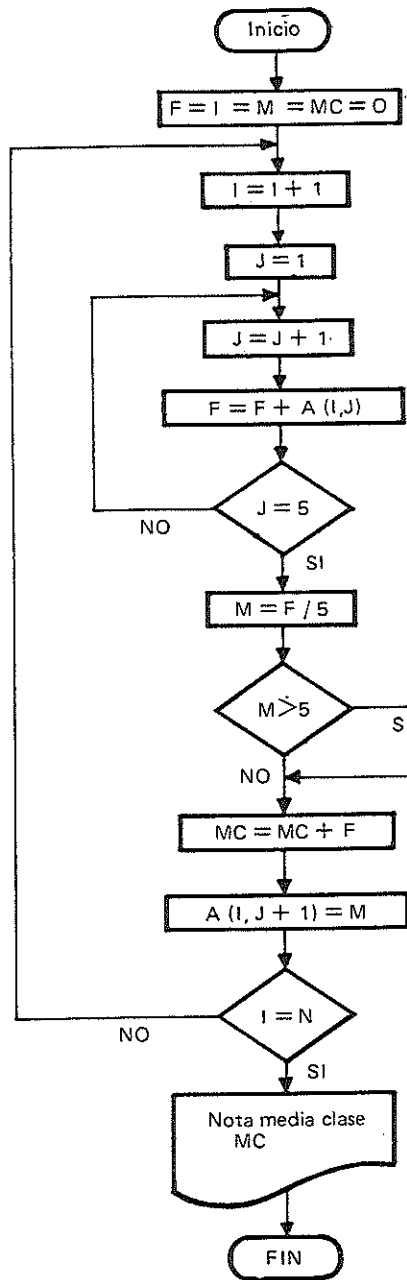
C = Suma acumulada de cada columna.



57) Estando dimensionada una matriz en memoria de N por M elementos, rellenarla de tal forma que el valor de cada elemento sea la suma del número de su fila y del número de su columna.



58) Tenemos una matriz con las puntuaciones sacadas por los estudiantes en una clase de Cobol (cinco notas por estudiante). En la primera columna tenemos el nombre, en las cinco siguientes las notas, y la última columna está vacía para guardar la media del estudiante. Queremos hallar la puntuación media de cada estudiante, hallar la nota media de la clase y la lista de los aprobados.



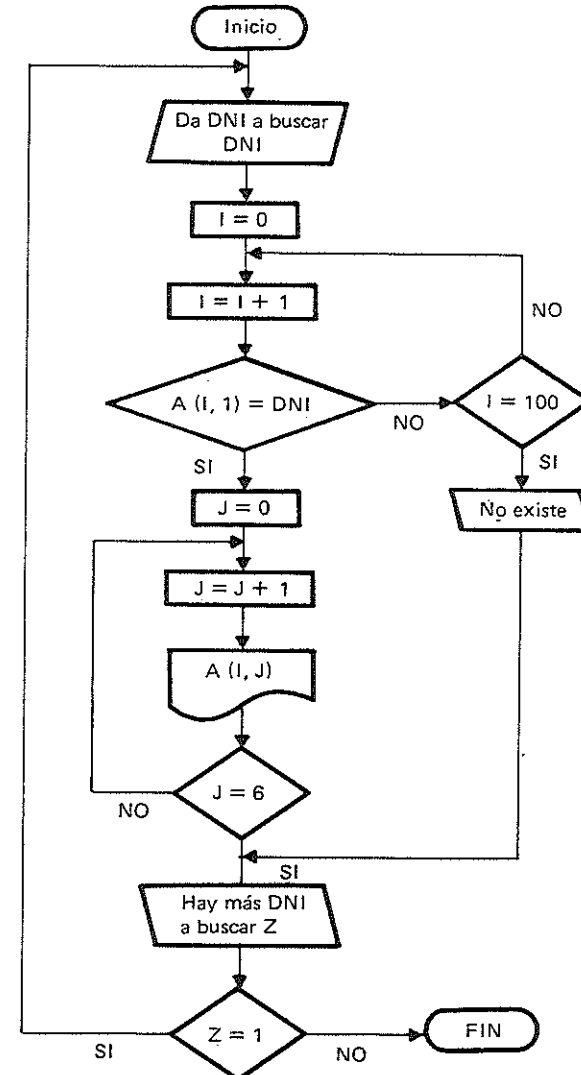
Lista de variables

F = Suma fila matriz

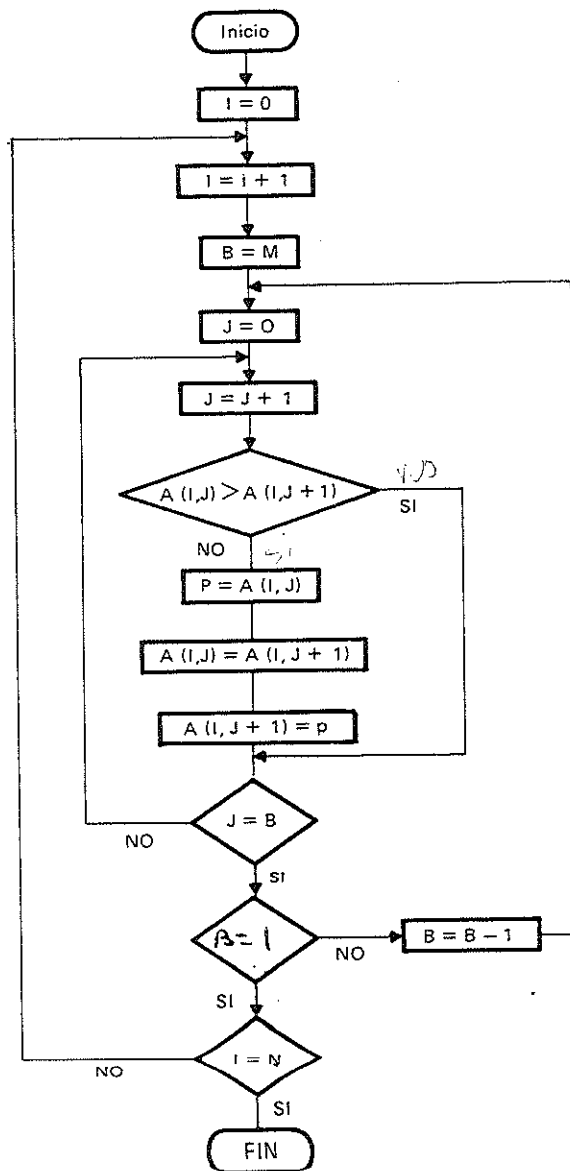
M = La media del estudiante.

MC = Media de la clase.

59) Dado un fichero en memoria central, en el que cada registro va a contener la información referente a un individuo con los siguientes campos: DNI, EDAD, SEXO, PROFESION, ESTADO CIVIL, INGRESOS. Queremos, que dado un DNI, por el teclado se busquen los datos correspondientes y se impriman. Suponemos que el fichero tiene 100 registros.



- 60 Dada una matriz de N por M elementos, ordenar de menor a mayor cada una de sus filas.

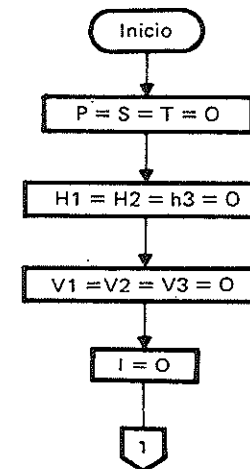


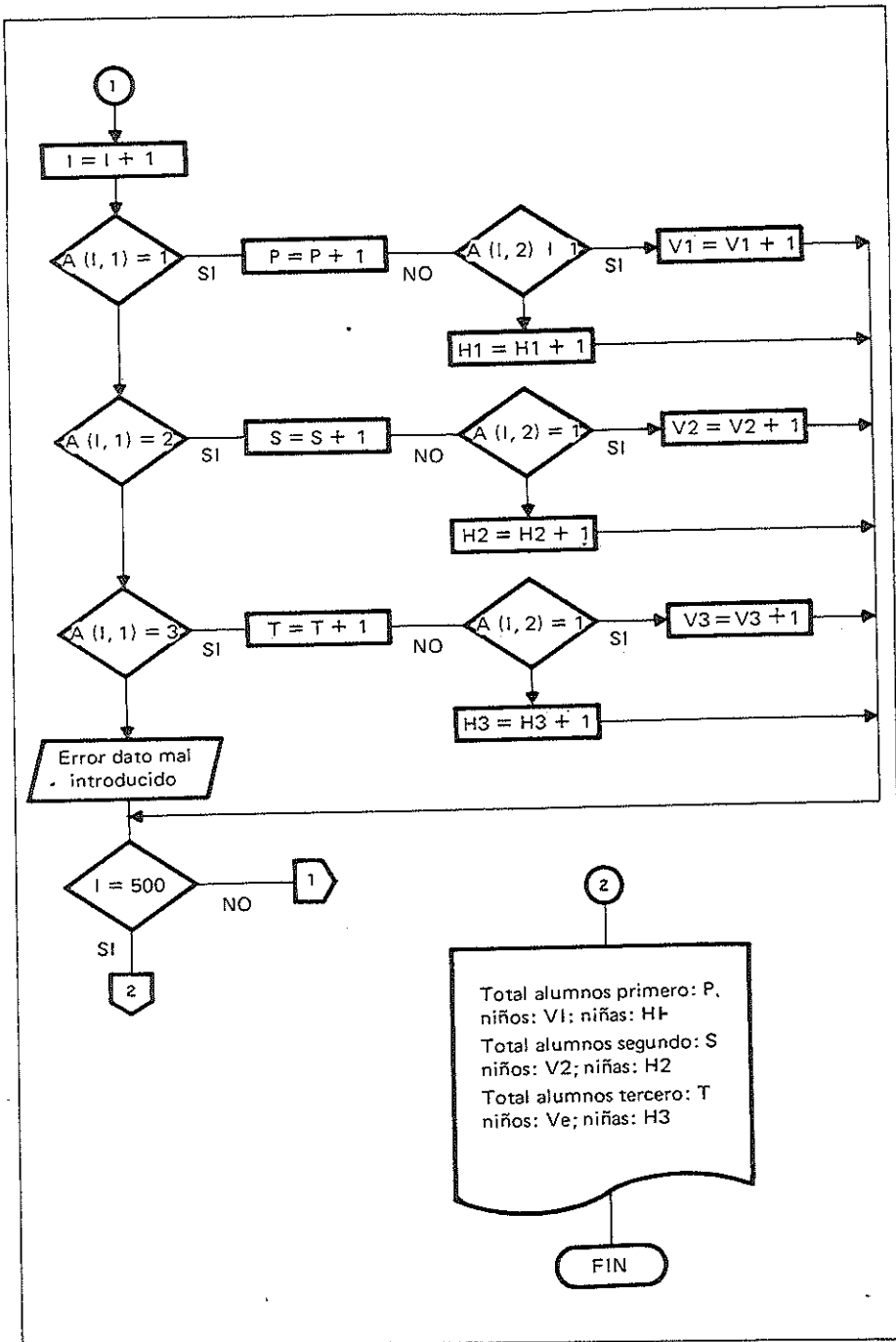
- 61 Tenemos una matriz en memoria con los datos de los alumnos de un colegio. En cada fila tenemos los datos de un alumno: 1.^a columna curso, 2.^a columna sexo, 3.^a columna núm. de matrícula, 4.^a columna nombre.

Queremos investigar la tabla para saber de los 500 alumnos cuántos hay de cada curso, y dentro de cada curso cuántos son niños y cuántos niñas. Existen tres cursos diferentes en el colegio.

Lista de variables

- P = en esta variable iremos sumando los alumnos de primero.
- V1 = en esta iremos sumando los niños de 1.^o
- H1 = en esta iremos sumando las niñas de 1.^o
- S = esta irá sumando los alumnos de 2.^o.
- V2 y H2 serán los niños y niñas de 2.^o respectivamente.
- T = en esta iremos sumando los alumnos de 3.^o.
- V3 y H3 serán los niños y niñas de 3.^o respectivamente.

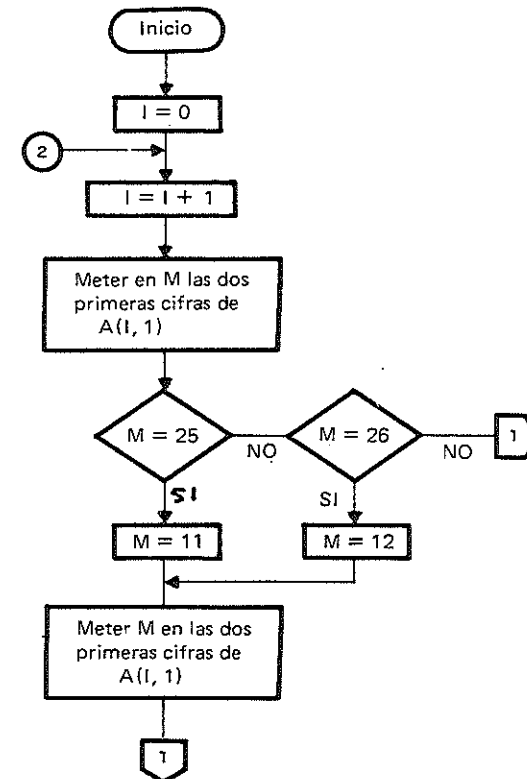


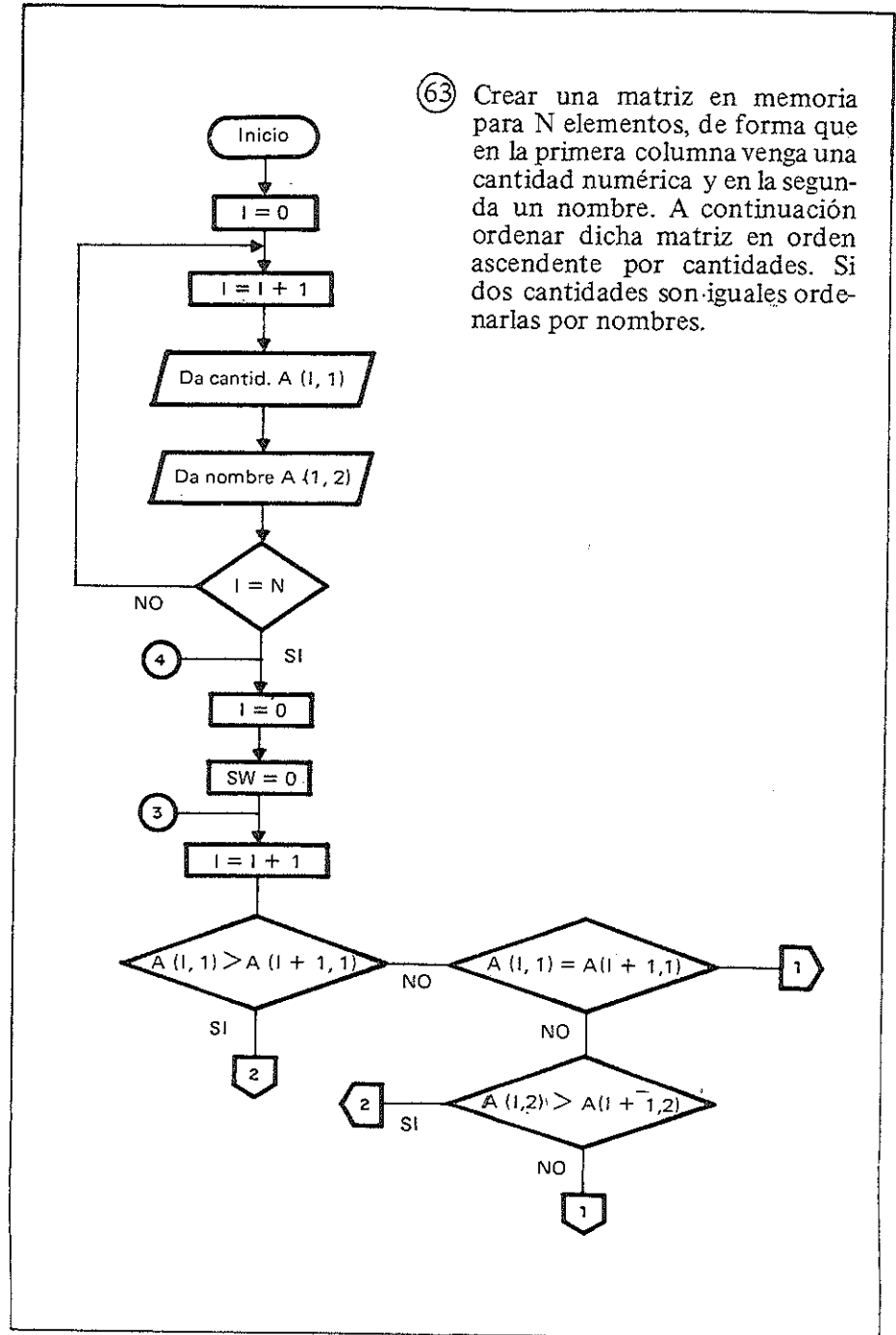
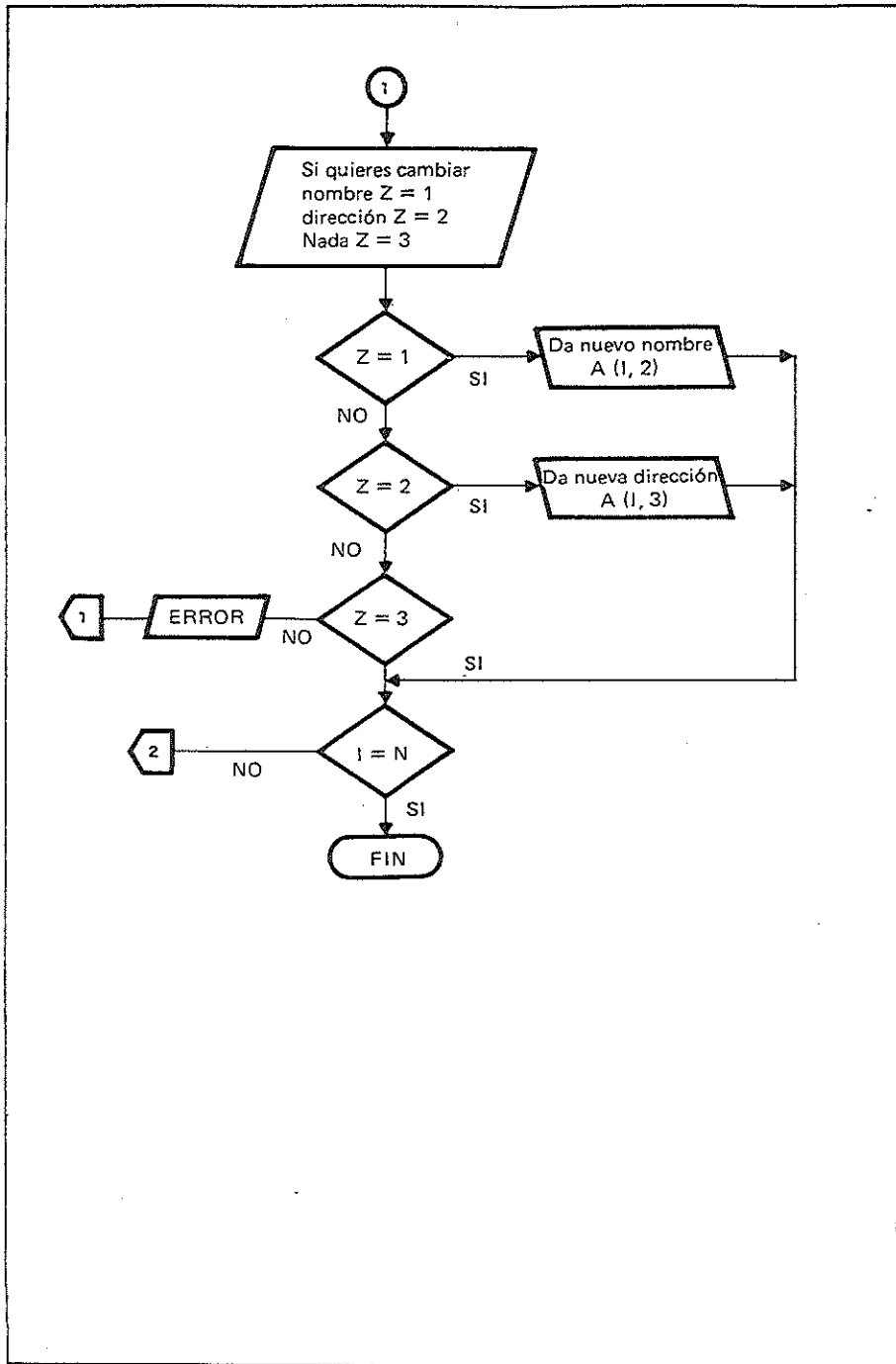


- 62) Tenemos una matriz en memoria con los datos de un abonado: número de teléfono, nombre, dirección y zona. Queremos cambiar todos los teléfonos que empiecen por 25 y 26, para que empiecen por 11 y 12. También queremos dar la posibilidad de cambiar cualquier otro campo.

Lista de variables

M = en esta variable meteremos las dos primeras cifras de los números de teléfono. Como dependiendo del lenguaje utilizado se realiza de una forma u otra, esta operación la dejaremos indicada.





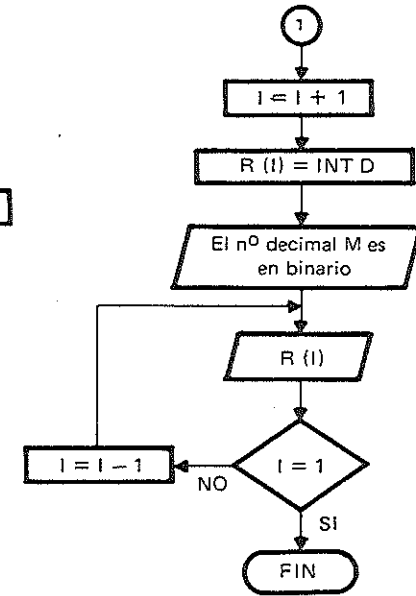
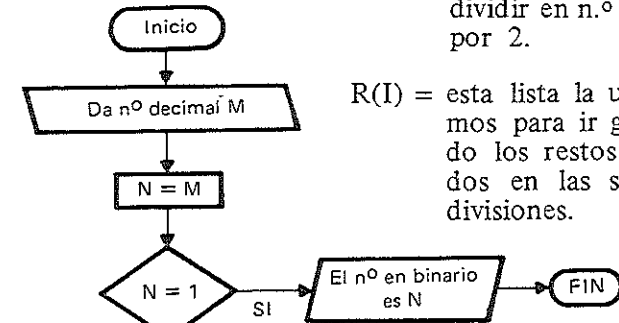
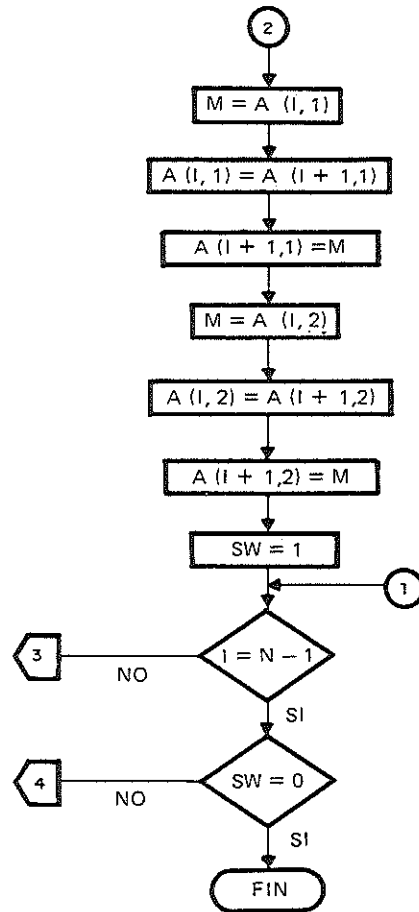
64 Realizar el organigrama para pasar un número de decimal a binario.

Lista de variables

M = en esta variable introduciremos el n.º decimal.

D = en esta variable meteremos el resultado de dividir en n.º decimal por 2.

R(I) = esta lista la utilizaremos para ir guardando los restos obtenidos en las sucesivas divisiones.



65) Vamos a realizar un organigrama que realice la factura detallada de la compañía telefónica, de forma que el cliente debe pagar 3 ptas. por cada paso del contador y a continuación una tarifa de alquiler que depende de la zona donde está situado el teléfono. Si es zona 1 pagará 1.000 ptas., si es zona 2 pagará 1.200 ptas., si es zona 3 pagará 1.500 pesetas.

Partimos de la base de que los datos del cliente ya están en la memoria en una matriz. 1.^a columna N.^o de clientes, 2.^a columna nombre del cliente, 3.^a dirección, 4.^a lectura anterior del contador y 5.^a zona. Tenemos que pedir por pantalla la lectura actual y el n.^o de cliente. Prever un final.

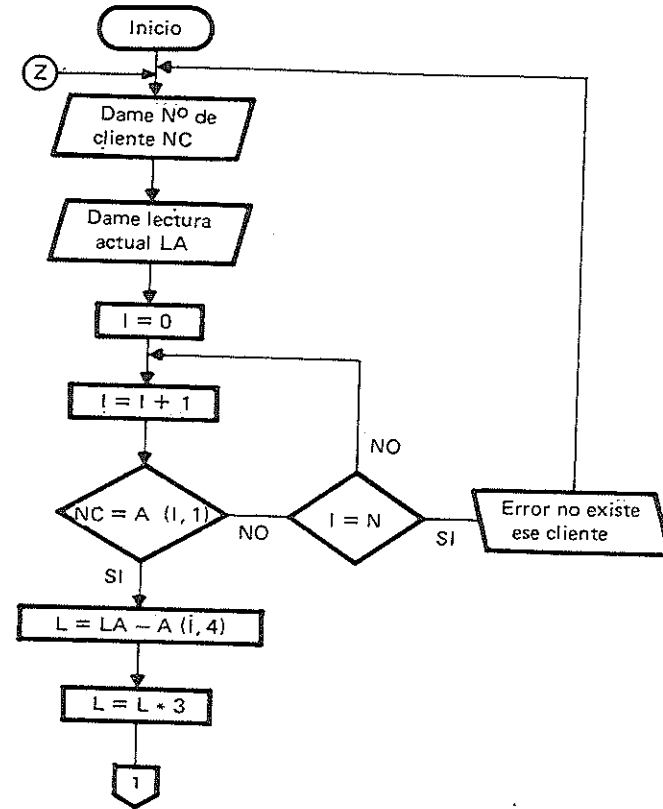
Lista de variables

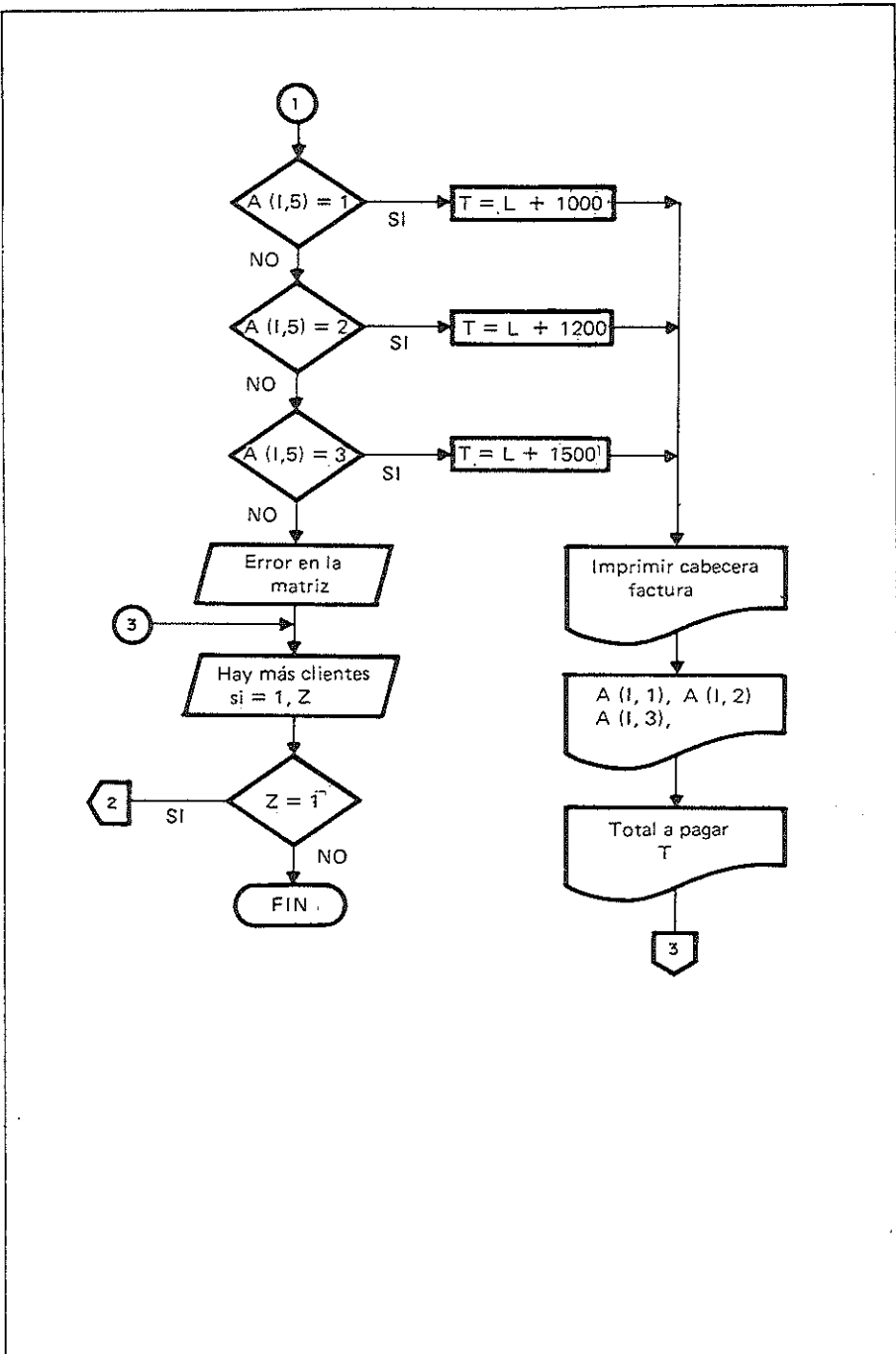
LA = en esta variable introducimos la lectura actual.

NC = en esta variable introducimos el n.^o del cliente.

L = esta variable la utilizamos para hallar los pasos consumidos.

T = esta variable la utilizamos para hallar el total a pagar.





4 TRABAJANDO CON FICHEROS

Hemos estado trabajando con variables simples y compuestas. Es decir, hasta ahora todos nuestros datos o información de partida, estaban almacenadas en la memoria interna.

Como ya sabemos, la memoria interna tiene una capacidad limitada, para almacenar información. Si para resolver nuestro problema necesitamos muchos datos básicos, no nos interesa guardarlos en la memoria interna. En tal caso nos conviene guardarlos en la memoria externa. En tal caso nos conviene guardarlos en la memoria externa. En tal caso nos conviene guardarlos en la memoria externa.

En este capítulo vamos a trabajar con Ficheros en memorias auxiliares.

Un fichero es un conjunto de elementos que contienen información relativa a un tema común, estando esta información en cada elemento de acuerdo con unos formatos preestablecidos.

Así pues, podemos tener el fichero de los alumnos de un centro escolar, en el que cada elemento contendrá la información relativa a un alumno: el número de matrícula, el nombre, la dirección, el teléfono, el curso, el grupo, etc. A cada uno de los elementos de un fichero lo conocemos con el nombre de registro.

Sobre estos ficheros podemos realizar una serie de operaciones, como:

Creación del fichero

Consiste en el almacenamiento en un soporte externo, de los registros que lo constituyen.

Consulta de un fichero

Es la operación que permite la búsqueda de un determinado registro, bien para su copia o para la utilización posterior de sus campos.

Actualización de un fichero

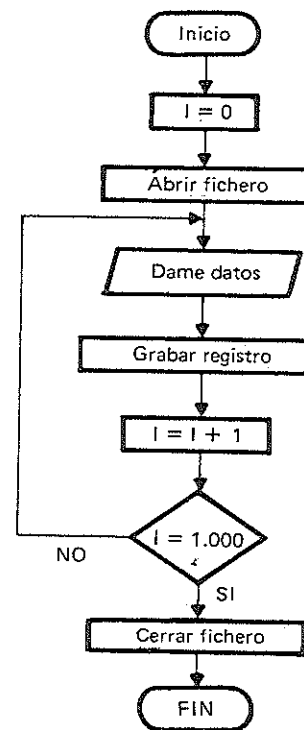
Es la operación que permite renovar el contenido de los campos de un fichero. Normalmente al actualizar un fichero se pueden realizar tres operaciones distintas:

- 1.º Inserciones: incluir nuevos registros en el fichero.
- 2.º Modificaciones: modificar el contenido de algún registro de dicho fichero.
- 3.º Eliminaciones: suprimir algún registro que ya no es necesario.

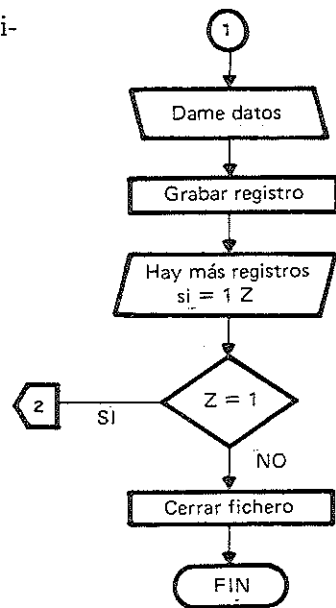
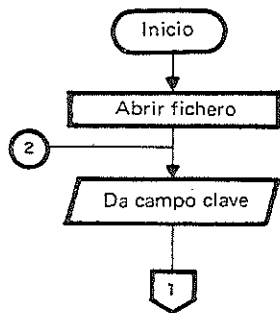
Estas operaciones las podemos realizar secuencialmente, es decir, tratar un registro detrás de otro, hasta que se hallan tratado todos. O bien aleatoriamente, que consiste en tratar los registros sin un orden determinado y no siendo necesario tratar todos los registros del fichero.

También se han tenido en cuenta en este capítulo las diferentes organizaciones que existen de los ficheros.

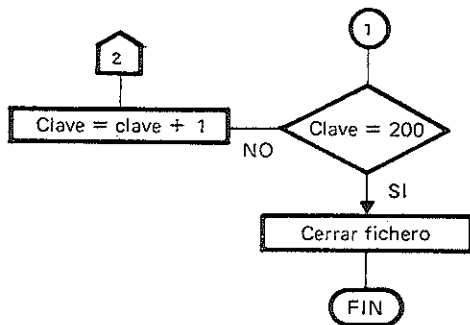
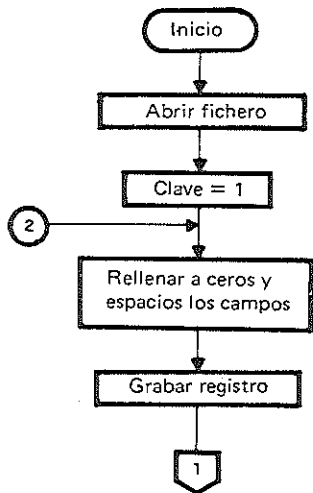
- 66 Crear un fichero con organización secuencial, para 1.000 registros.



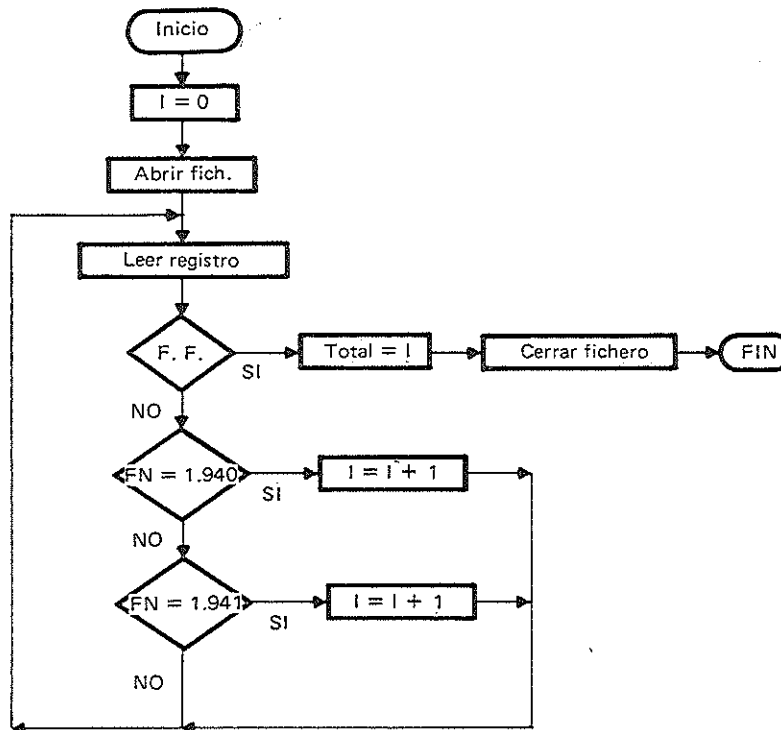
67 Crear un fichero con organización indexada.



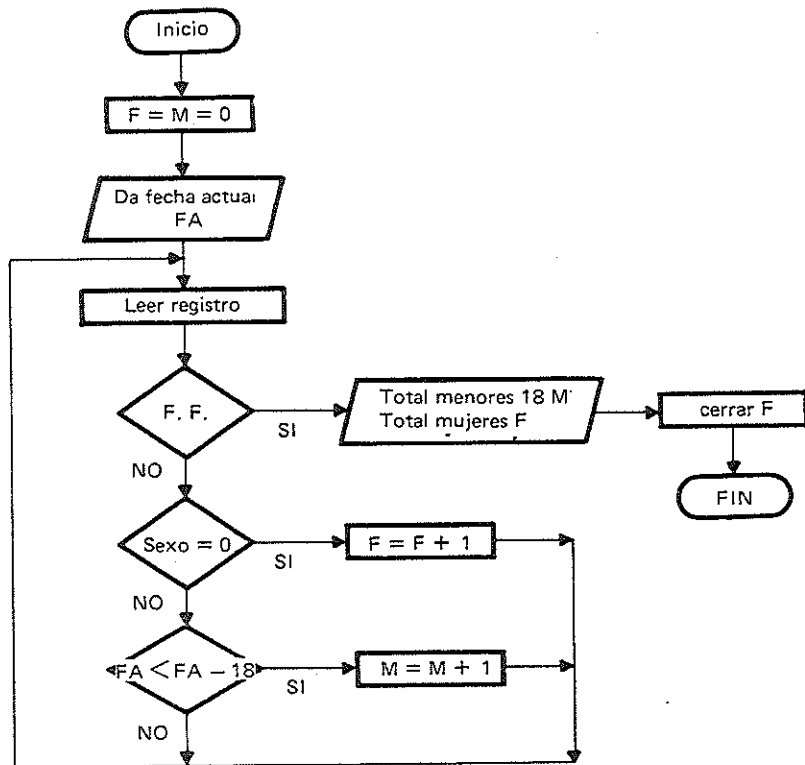
68 Crear un fichero de acceso directo, con organización relativa para 200 registros, rellenándolo a ceros y espacios.



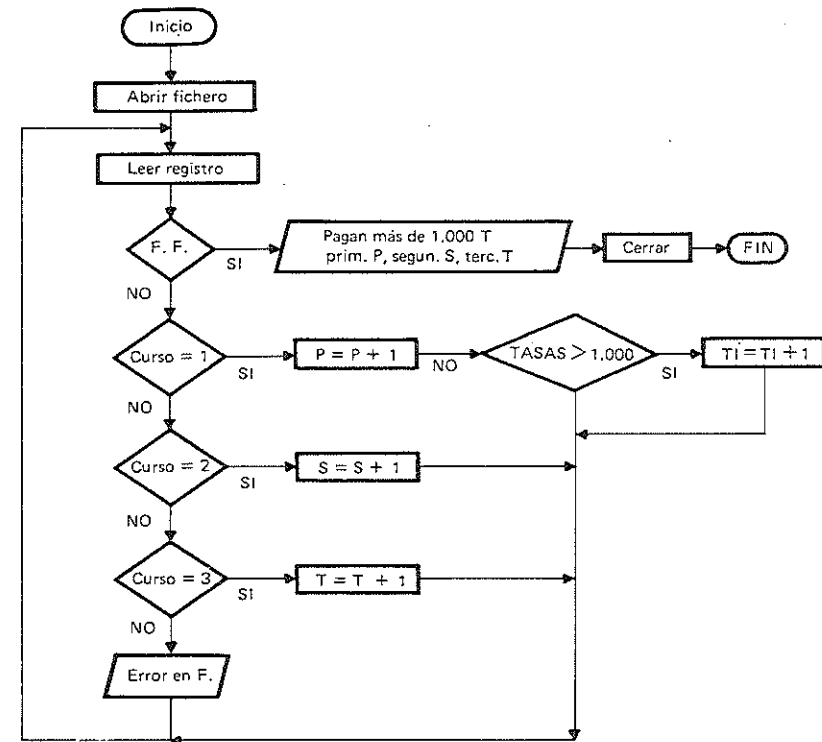
69 Dado un fichero secuencial con los datos de los empleados de una empresa, contar todos aquellos empleados que hayan nacido en 1940 y 1941.



- 70 Dado un fichero con organización secuencial, de los empleados de una empresa, contar todos los empleados que sean menores de 18 años y todos los de sexo femenino.



- 71 Dado un fichero con organización y acceso secuencial, cuyos registros son de la forma: n.º, nombre, tasas, rama, especialidad, información. Contar el n.º de alumnos matriculados en cada curso, y dentro de los de 1º los que pagan más de 1.000 ptas.



- 72 Consideramos una empresa que vende tres artículos diferentes, por medio de una serie de vendedores. Se perforará en una tarjeta las ventas que semanalmente realiza cada vendedor de cada producto. C_1 C_2 C_3 .

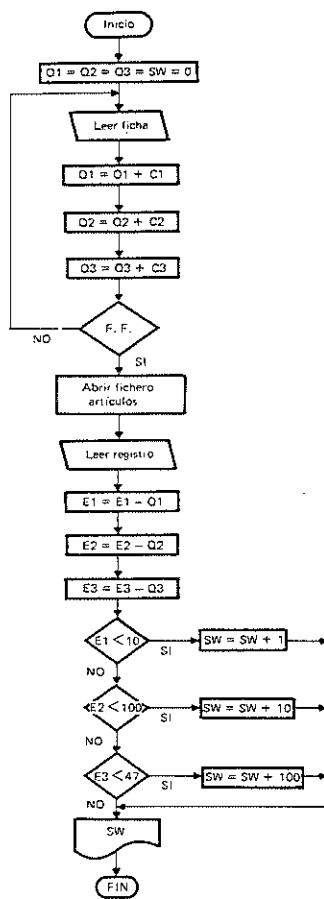
Al final de cada mes hay que actualizar las existencias del fichero de artículos y a la vez comprobar si dichas existencias han llegado al stock mínimo. En tal caso hacer un nuevo pedido. Stock mínimo artículo 1 = 10 piezas, stock mínimo artículo 2 = 100 piezas, y stock mínimo artículo 3 = 47.

Lista de variables

Q_1, Q_2, Q_3 = al total vendido al mes por cada uno de los artículos.

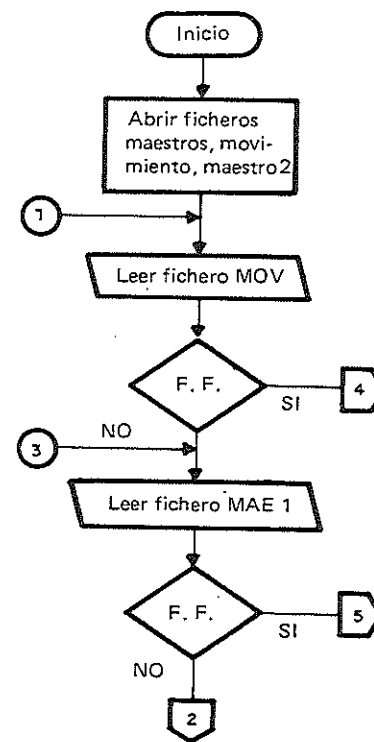
E_1, E_2, E_3 = existencias de cada uno de los artículos.

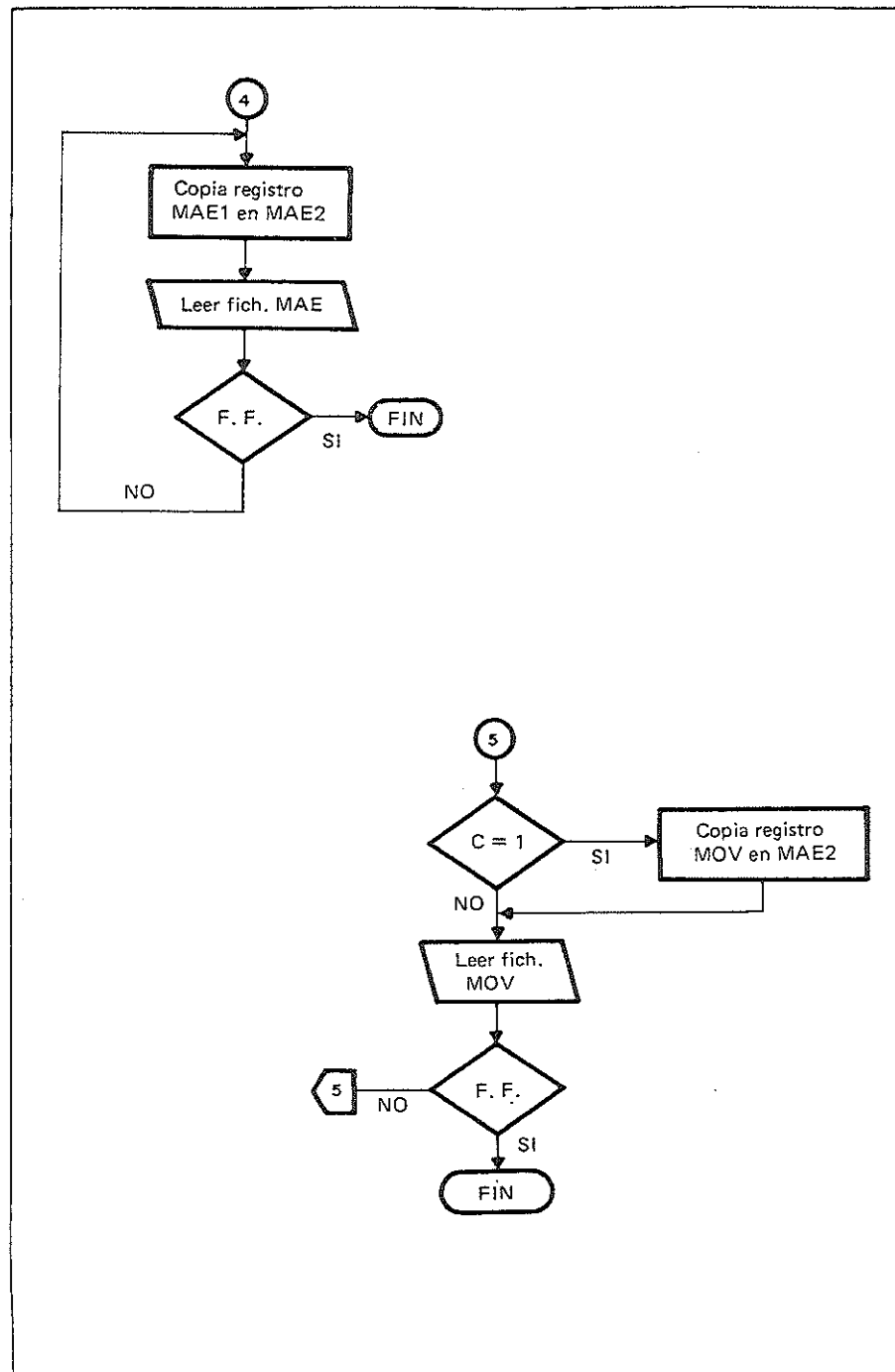
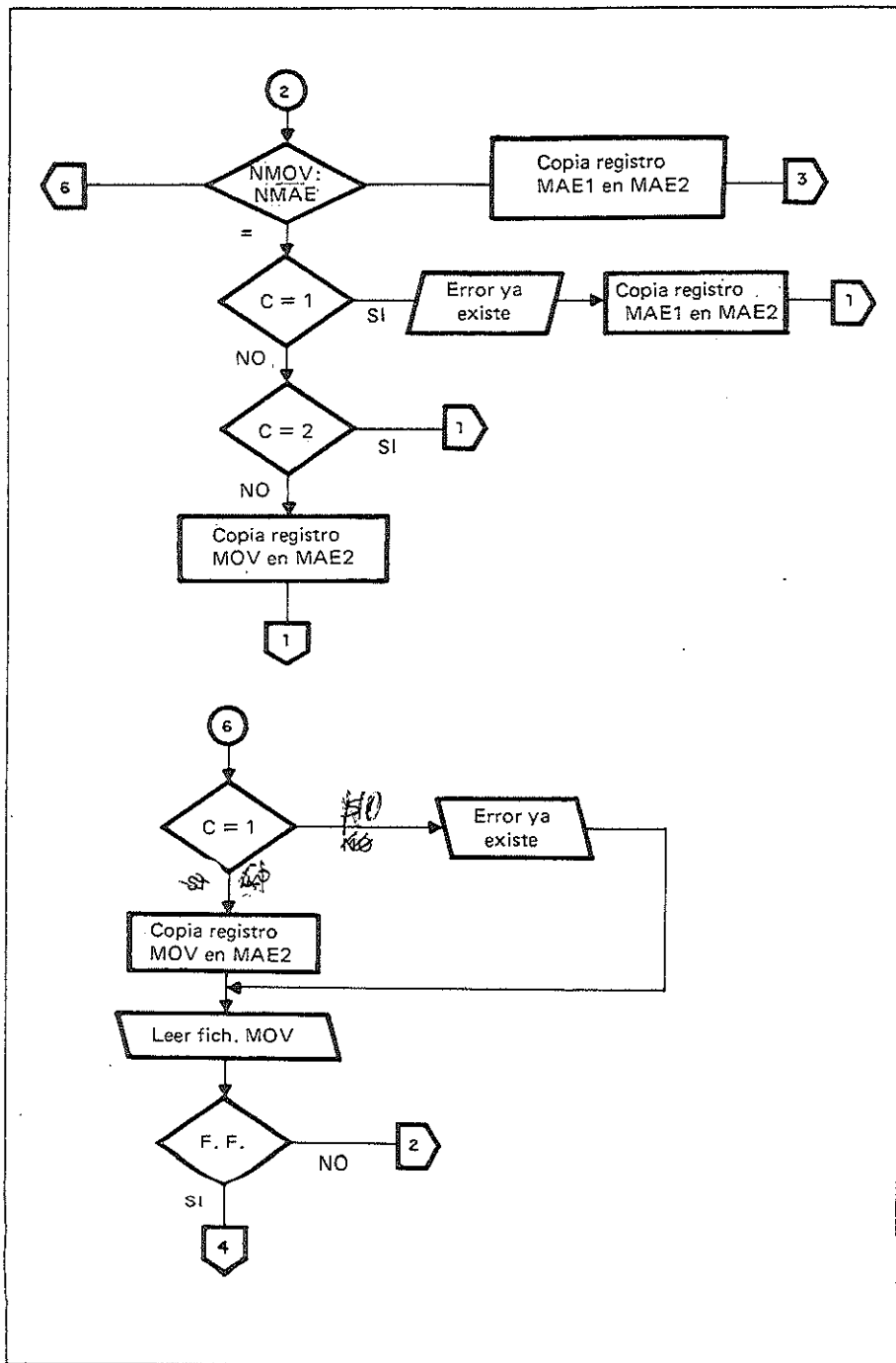
SW = vamos a usar esta variable de forma que si aparece un 1 en sus unidades, hay que comprar del artículo uno. Si aparece el uno en las decenas hay que comprar del artículo dos, y si aparece en las centenas habrá que comprar el artículo tres. De esta forma, imprimiendo al final el contenido de esta variable, sabremos de qué artículo hay que hacer un nuevo pedido.



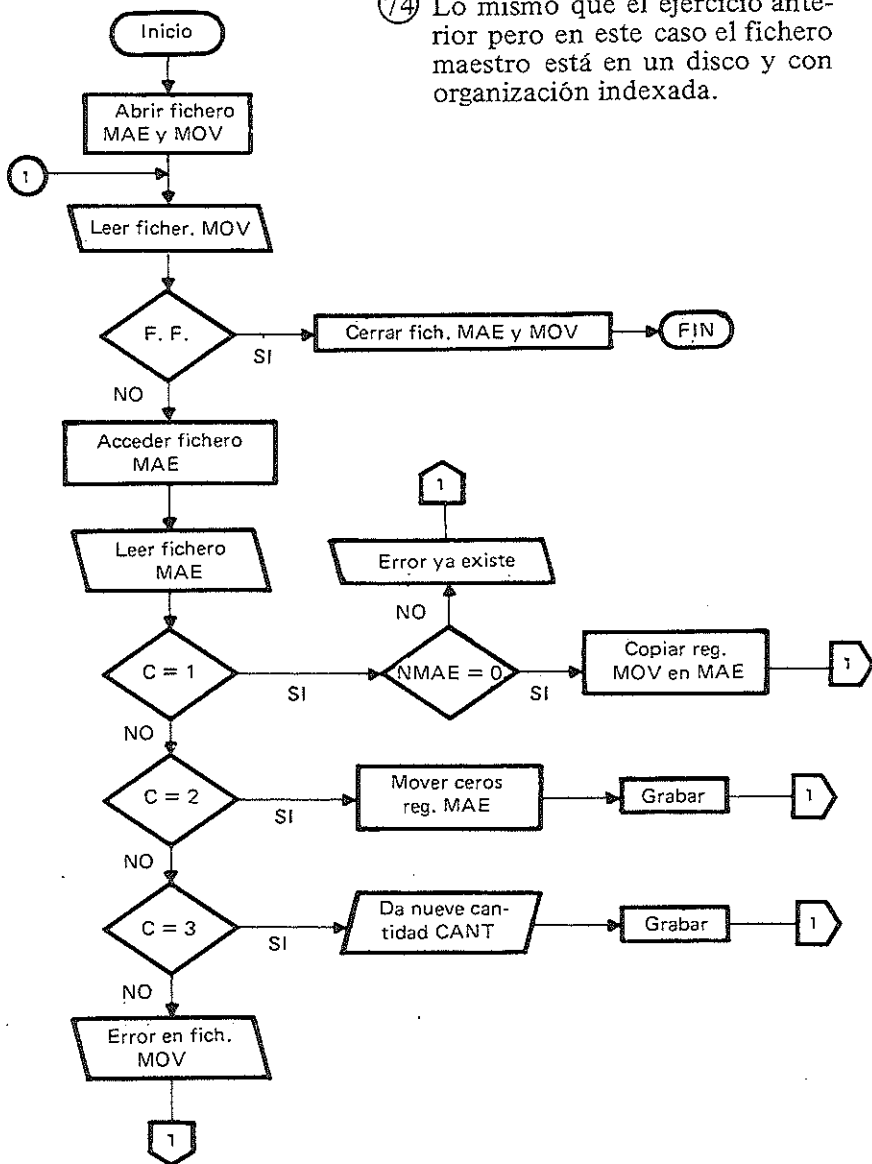
- 73 Tenemos un fichero maestro con dos campos: n.º de artículo (NMAE) y cantidad (CANT). Tenemos también un fichero de movimientos con tres campos, n.º de artículo (NMOV), cantidad (CAN) y clave de movimiento (C). Las claves de movimientos son 1 si es una alta, 2 si es una baja y 3 si es una modificación.

El fichero maestro y el de movimientos estarán ordenados por el n.º de artículo en orden ascendente. Se pretende actualizar el fichero maestro por medio del fichero de movimientos, teniendo en cuenta que los dos ficheros son secuenciales. Por lo tanto habrá que hacer un nuevo fichero maestro.

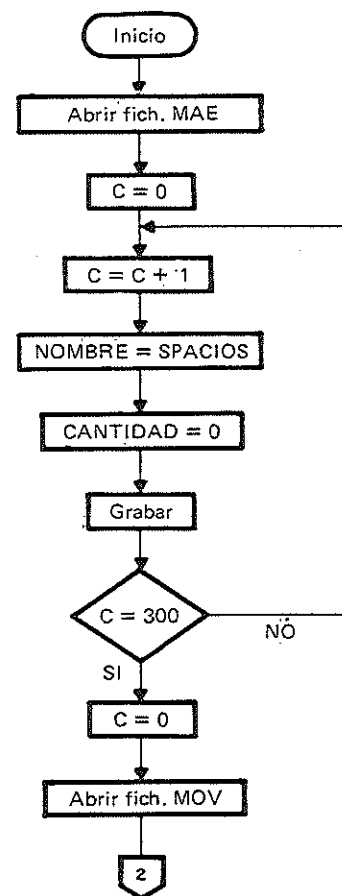


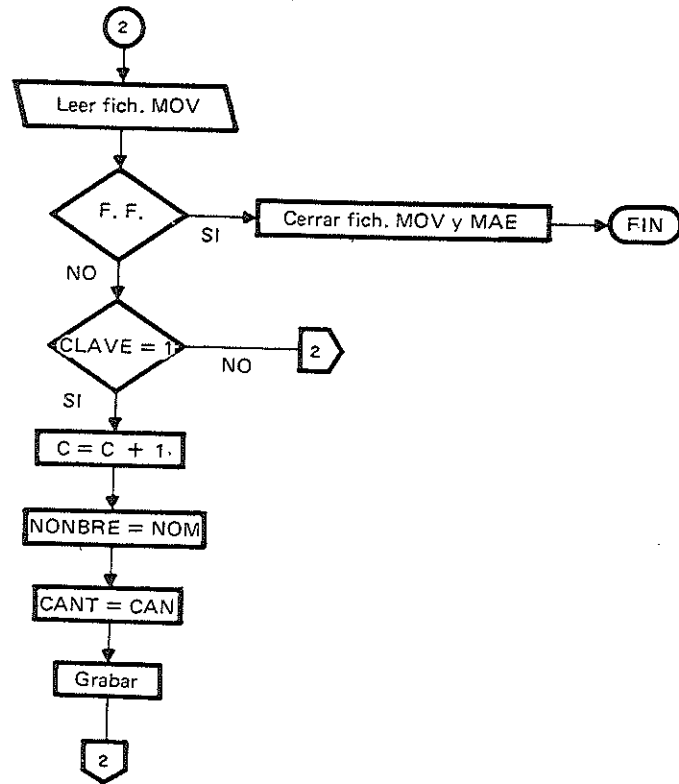


74) Lo mismo que el ejercicio anterior pero en este caso el fichero maestro está en un disco y con organización indexada.



75) Crear un fichero en memoria auxiliar con acceso directo, con trescientos registros, llenándolos de ceros. Cada registro constará de tres campos clave (C), nombre (NOMBRE) y cantidad (CANT). A continuación abrir un fichero de movimientos secuencial, que ya está creado, el cual consta de tres campos: nombre (NOM) cantidad (CAN) y un 1 si el registro debe ser copiado y un cero si no debe ser copiado. Queremos ir leyendo registros de fichero de movimientos e ir copiando los necesarios en el fichero maestro.





76 Tenemos creados tres ficheros, con las siguientes características:

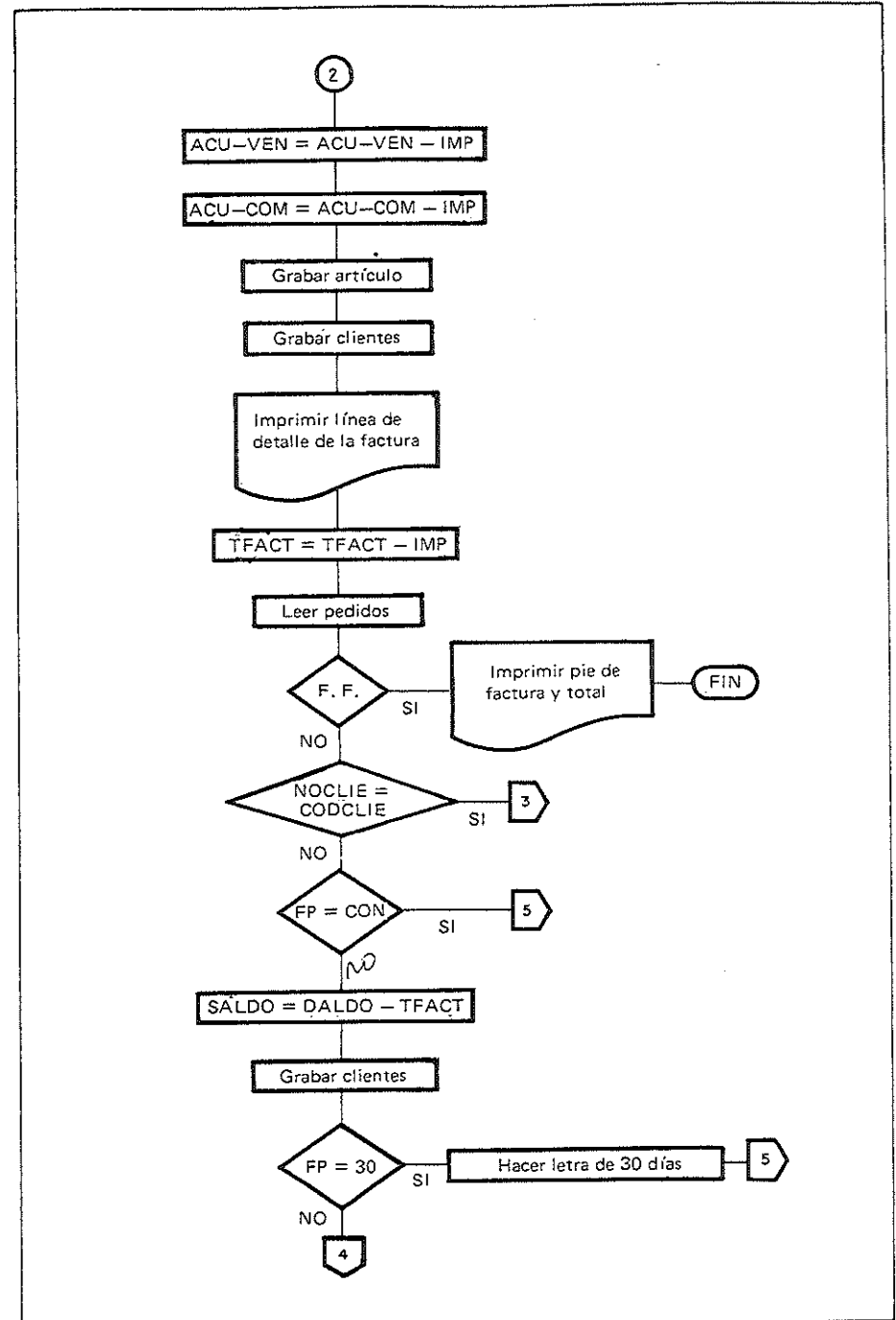
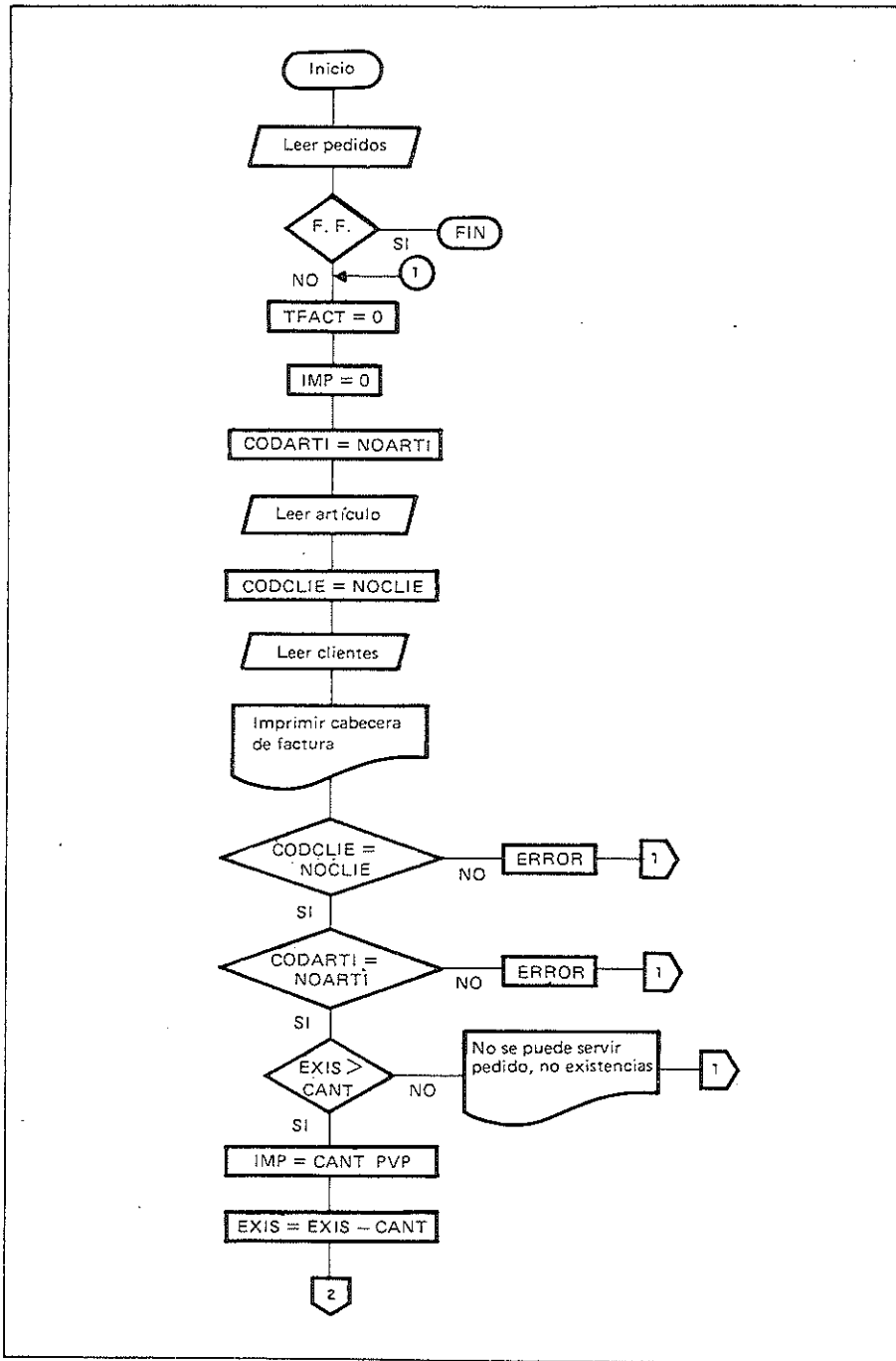
El de pedidos, ficheros secuencial cuyo registro es de la forma / *NOCLIE* / *NOARTI* / *CANT* /

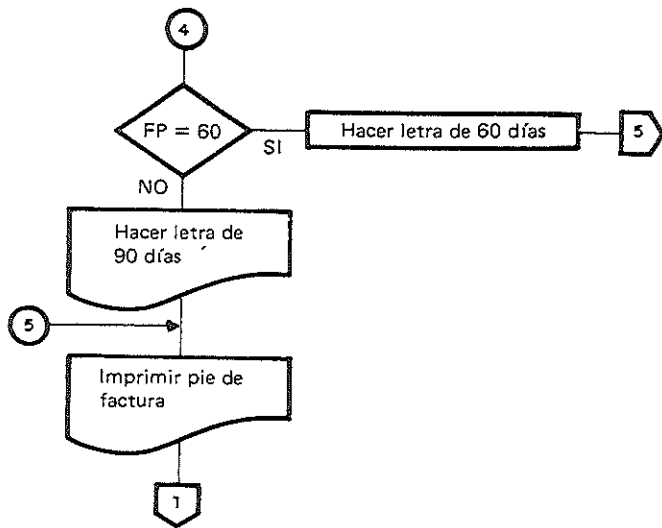
En el artículo, con organización relativa y acceso directo:
/ *CODARTI* / *DESCRIPCION* / *EXIS* / *ACU-VEN* / *STOCK* / *PVP* / *PC* /

El de Clientes, con organización relativa y acceso directo:
CODCLIE / *NOMBRE* / *DIRECCION* / *TELEFONO* / *ACU-COM* / *SALDO* / *FP* / *DESCUEN* /

Se desea crear la factura de todos los clientes que tenemos en el fichero de pedidos y a la vez actualizar el fichero de clientes y el de artículos. En el fichero de artículos actualizaremos las existencias y el acumulado de ventas y en el de clientes el saldo cuando el pago no sea al contado y el acumulado de compras.

En el campo FP, forma de pago, tendremos CON si el pago es al contado, 30 si el pago es a los treinta días, 60 si el pago es a los 60 días y 90 si el pago es a los 90 días.





A continuación vamos a realizar una serie de aplicaciones completas.

Esta primera aplicación consiste en la gestión de la nómina para una empresa. Realizaremos los seis programas siguientes: >>

1. Creación del fichero de empleados

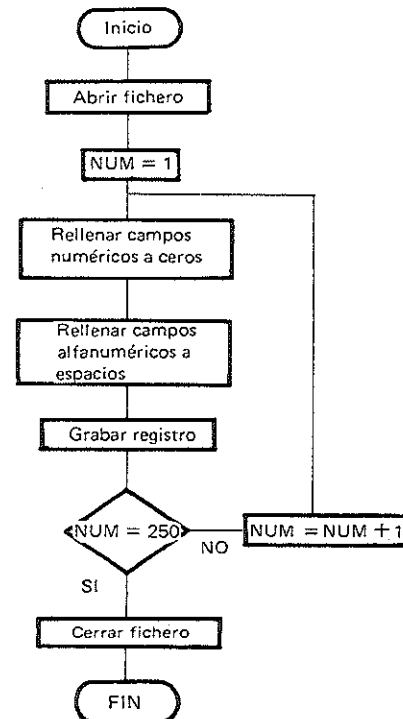
Lo creamos rellenándolo a ceros y espacios, para evitar errores a la hora de la grabación. Guardamos espacio para 250 registros. El formato es:

/ CLAVE / NOMBRE / DIRECCION / CATEGORIA /
 / FECHA / N.º DE SEGURIDAD SOCIAL / SUELDO BASE /
 / ESTADO CIVIL / NUMERO DE HIJOS /
 / DOCUMENTO NACIONAL DE IDENTIDAD /

Lista de variables

CAT = Categoría N.º CALLE = dirección. CATI = Categoría
 NSS = N.º de la seguridad social. SB = Sueldo base.
 EC = Estado civil. NH = N.º de hijos. DNI = D. N. I.
 NUM = en esta variable llevamos la clave del registro.

ORGANIGRAMA:



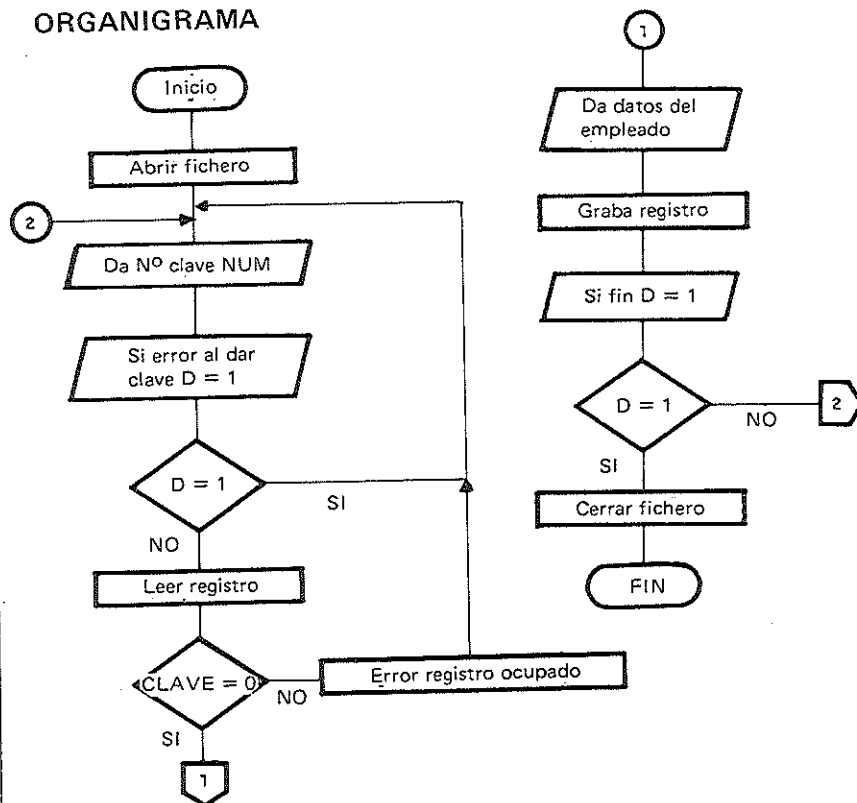
2. Inserción de registros en el fichero de empleados

En este programa introduciremos los datos del empleado nuevo por pantalla y a continuación lo grabaremos en el fichero. Como el fichero es de acceso directo, sabemos que el registro está vacío cuando el campo clave está a cero; así, si en la dirección indicada por NUM, el campo clave no está a cero, es que ese registro ya está creado; por lo tanto saldrá un mensaje de error y volverá a pedirnos la clave.

Lista de variables

D = En esta variable introduciremos el valor 1, si queremos volver a introducir la clave, porque nos hallamos equivocado. También empleamos esta variable para indicar que hemos terminado el proceso de inserción.

ORGANIGRAMA



3. Consulta, modificación y eliminación de registros del fichero de empleados

Pedimos la clave del registro a consultar, modificar o eliminar, leemos dicho registro y hacemos que aparezca por pantalla; a continuación preguntamos si queremos modificarlo o eliminarlo.

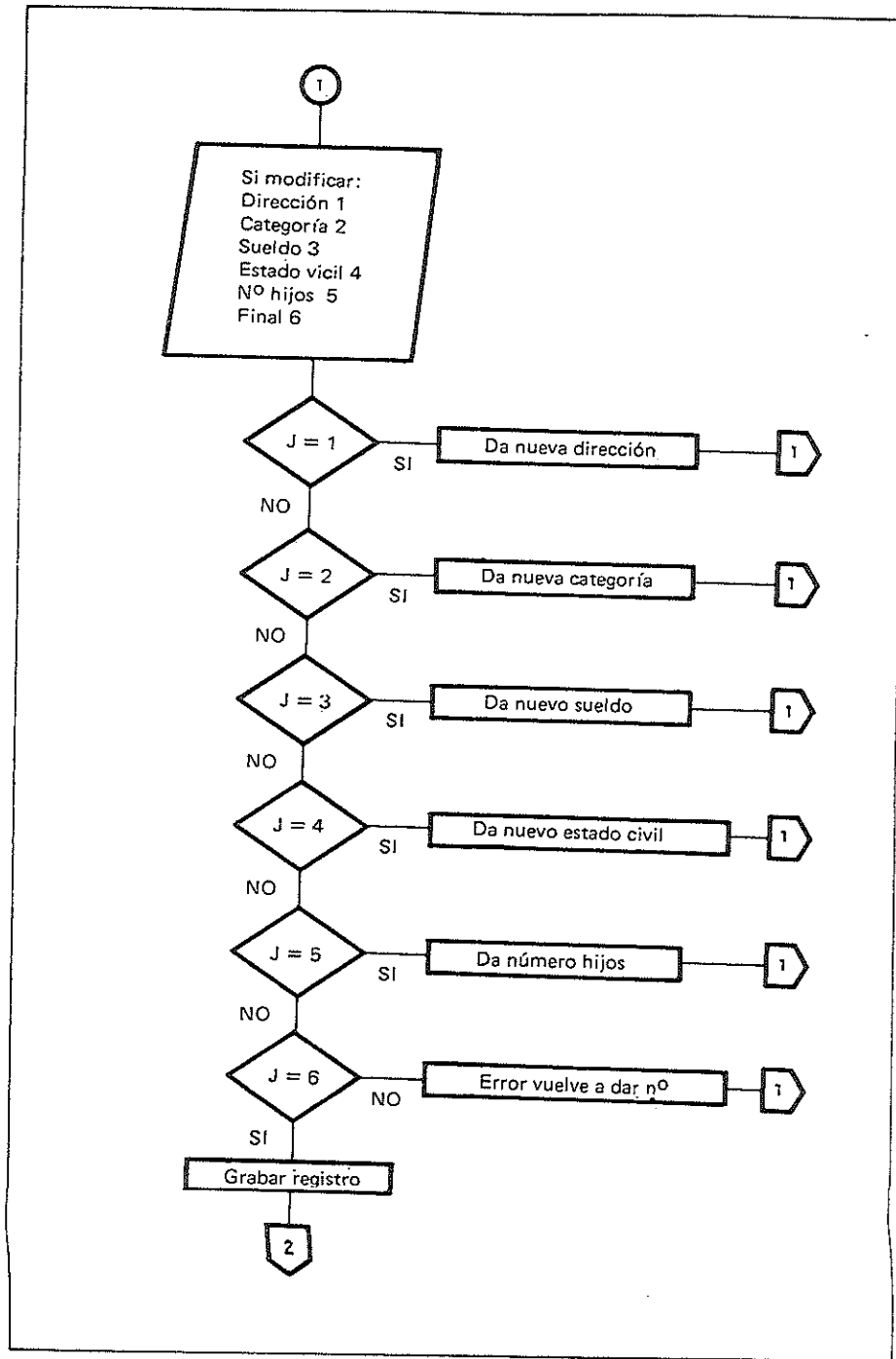
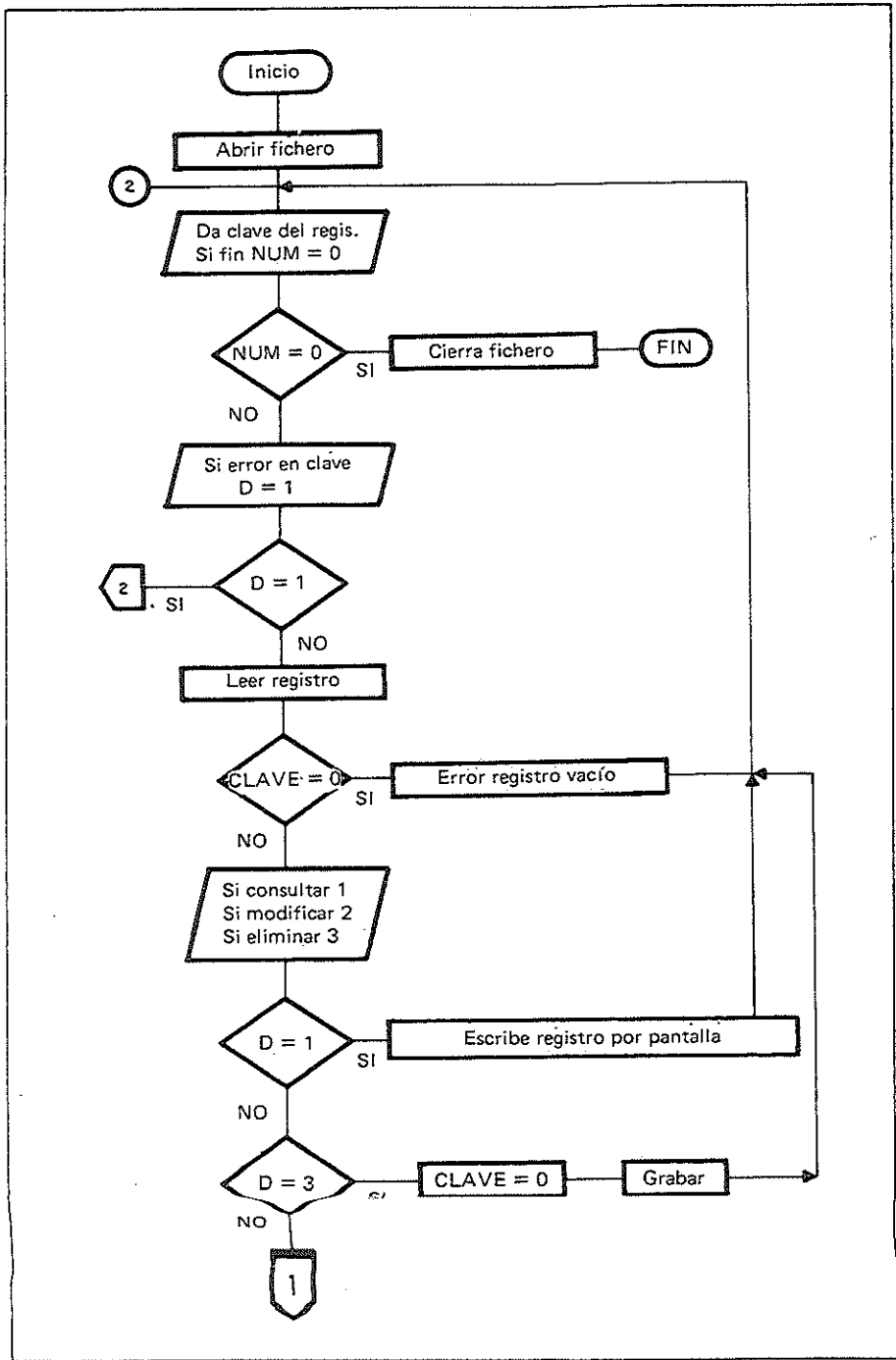
Hemos confeccionado un menú de forma que nos aparecen por pantalla las distintas opciones. Nosotros indicamos con un número la que nos interesa en ese momento.

Lista de variables

D = Esta variable la utilizamos para seleccionar la opción deseada.

J = Esta variable la utilizamos para indicar en caso de modificación qué campo es el que queremos modificar.

Ver diagramas en
pág. 116 y 117



4. Creación de tablas

Mediante este programa grabamos dos tablas en dos registros respectivamente. Una de ellas referida a las categorías, al plus de antigüedad, al plus de transporte y a las bases mínima y máxima para la cotización al régimen de la seguridad social, es decir, consta de 7 filas y 5 columnas. La otra contendrá la retención correspondiente para el I.R.T.P.F. dependiendo de el sueldo y del número de hijos. Es decir, constará de 33 filas y 9 columnas.

Lista de variables

F = indica la fila de la primera tabla.

CA = categoría.

ANT = plus de antigüedad.

TRAN = plus de transporte.

BMIN = base mínima.

BMAX = base máxima.

FL = filas de segunda tabla

IMPORTE = Columna del importe.

Hijos = n.º de hijos.

NUM1 = clave física del primer fichero.

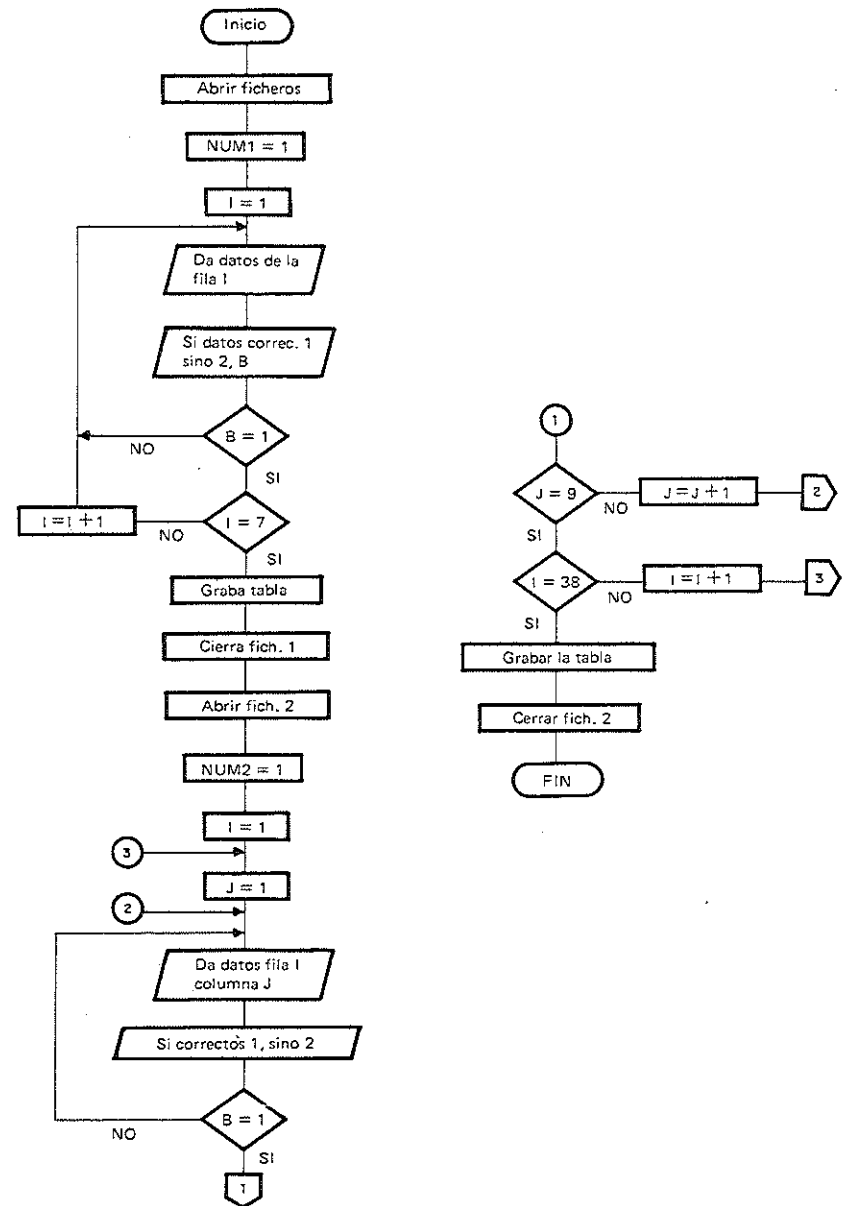
NUM2 = clave física del segundo fichero.

B = variable a la que daremos el valor 1 si la fila introducida ha sido correcta y 2 si ha sido incorrecta.

I = subíndice en la 1.ª tabla, indica en la 2.º el n.º de fila.

J = subíndice que indica en la segunda tabla el n.º de columna.

ORGANIGRAMA



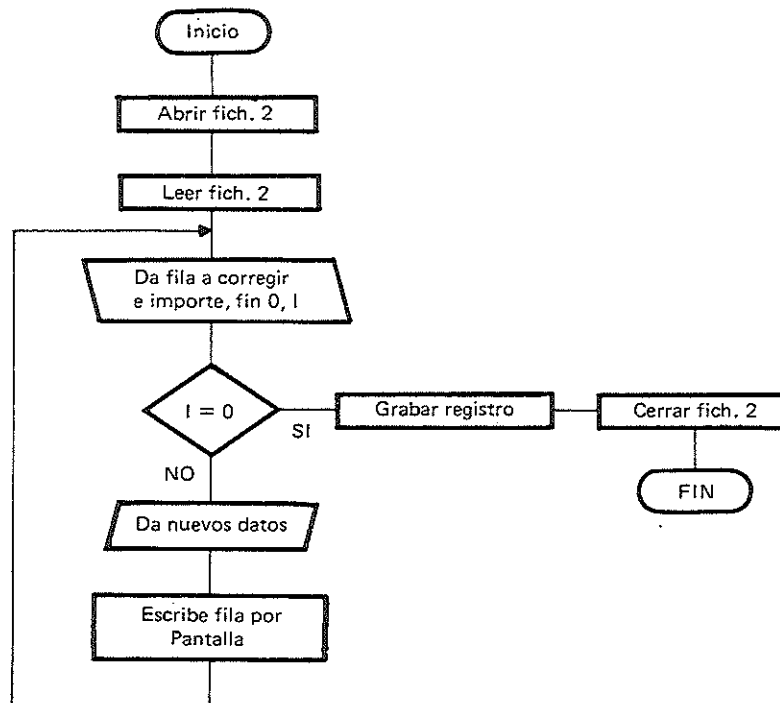
5. Modificaciones de la tabla general.

Nos da opción a modificar cualquiera de las filas de la tabla general para su actualización.

Lista de variables

I = indica n.º fila.

J = indica n.º columna.



6. Proceso

Este programa realiza el cálculo necesario, así como la impresión de la nómina. Utilizaremos en este programa el fichero de empleados, así como los ficheros de tablas.

Al iniciarse el programa se pide la fecha actual. Si el campo clave del fichero de empleados está a cero, aparecerá el mensaje y podremos introducir un nuevo campo clave.

Lista de variables

NUM1 = Clave física del fichero de empleados.

NUM2 = Clave física del fichero de tablas.

NUM3 = Clave física del fichero de tabla general.

I = Indica n.º de fila.

DIAL, MES1, A-01 = Día final de mes, mes y año actual. Serán movidas a las variables del fichero de salida DIL, MELL, A-01 respectivamente y a las variables DI, ME, A-0 9 Período de liquidación.

PROPE = Variable en la que calculamos la prorrata de las pagas extraordinarias. La moveremos a PROREL.

TOTA = Contendrá la remuneración total más la prorrata, la moveremos a TOTAL.

TANTUL = Variable de salida que cargaremos con 4,80 o 1,20 según corresponda, para calcular la aportación del trabajador.

CUA = Variable de entrada que contiene la variable sobre la que vamos a normalizar. La moveremos a CUAL.

D y D1 = Variables intermedias para el proceso de normalización.

CUAL2 = En ella calculamos el 4,80 por cien de la variable.

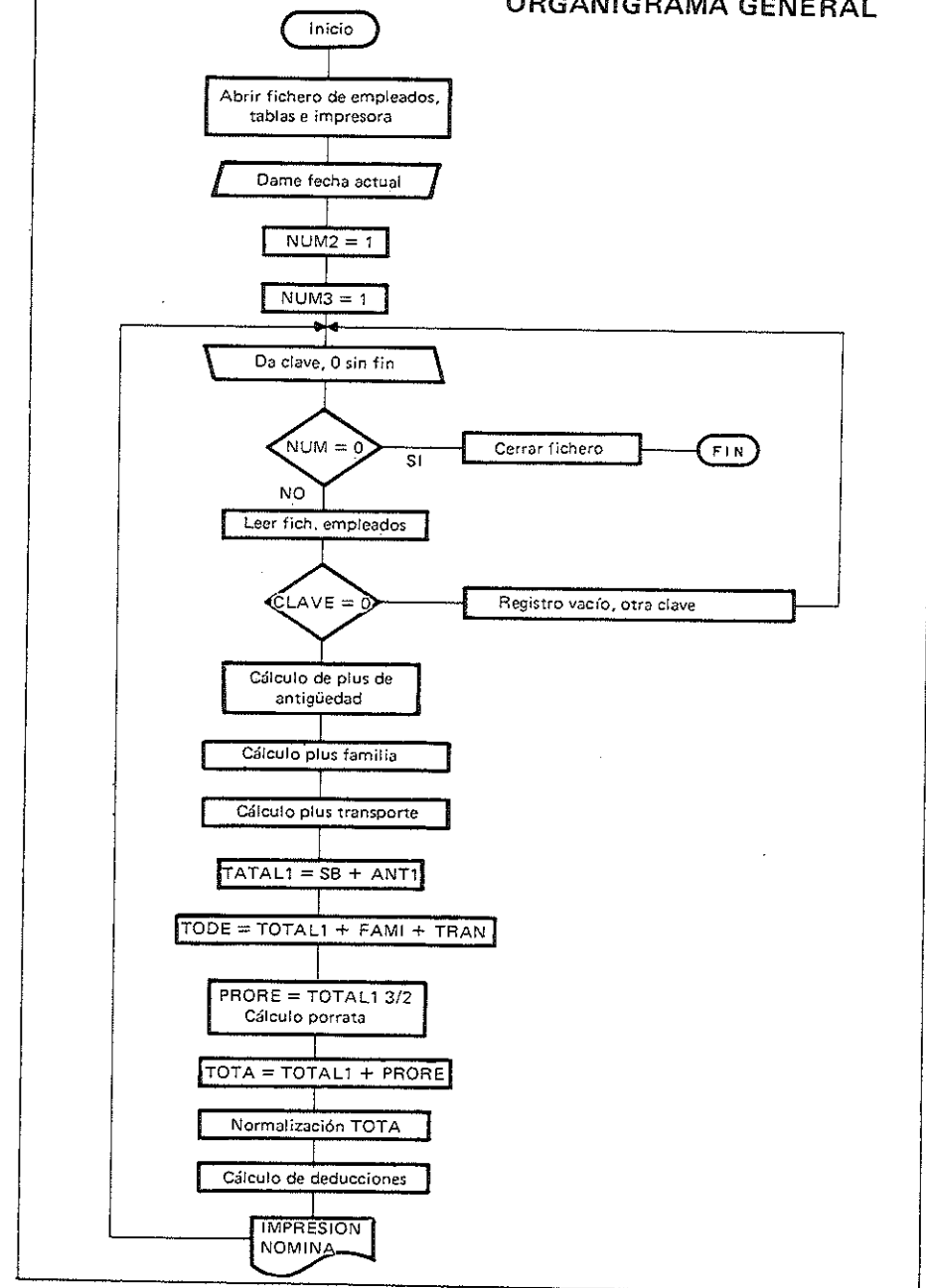
CUA = La moveremos a CUAL1.

CUA3 = En ella calculamos el 1,20 de CUA. La moveremos a CUAL1.

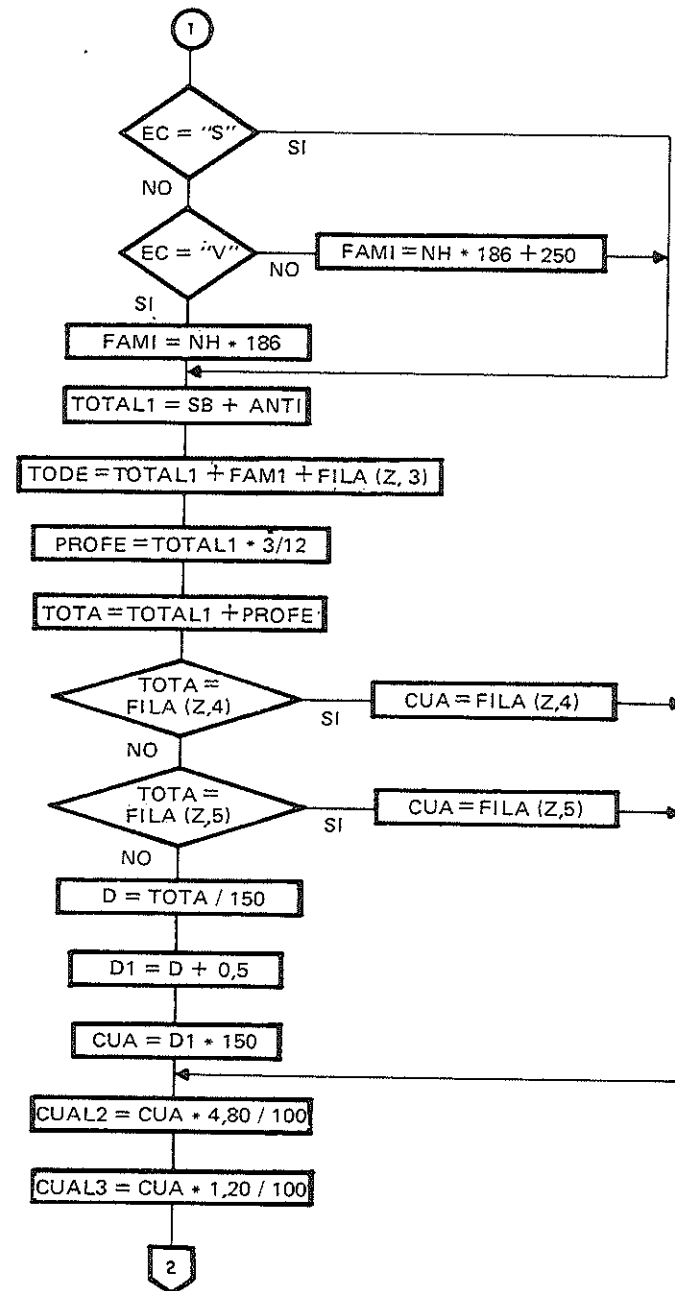
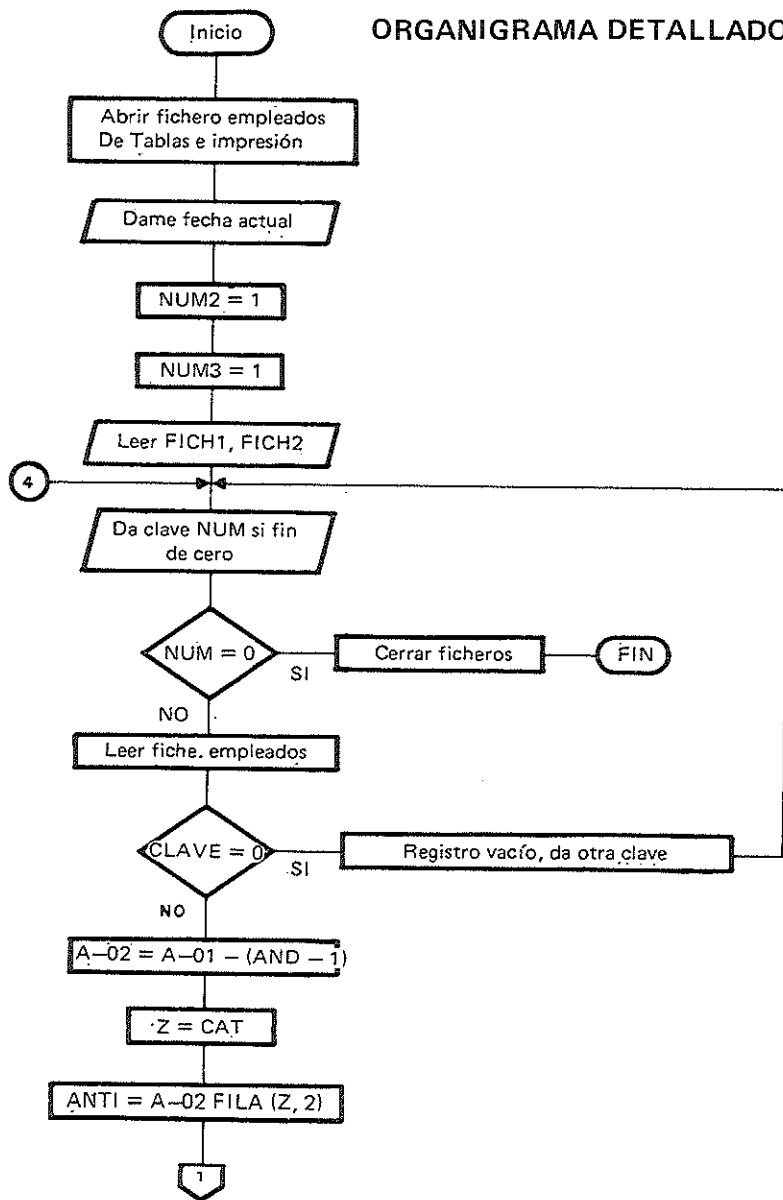
PROFE = En ella guardamos la suma de CUAL2 y CUAL3, movemos a PROFEL.

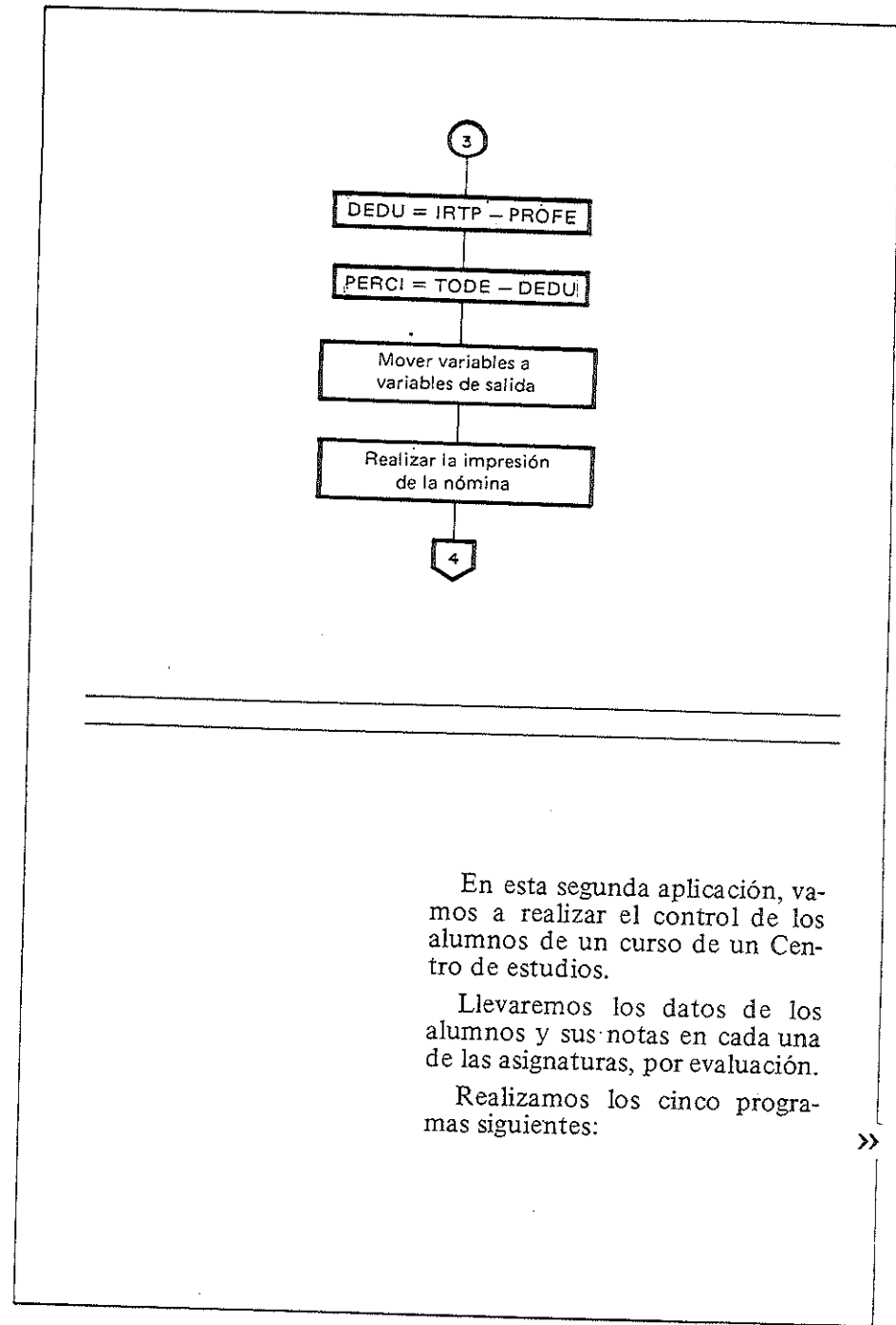
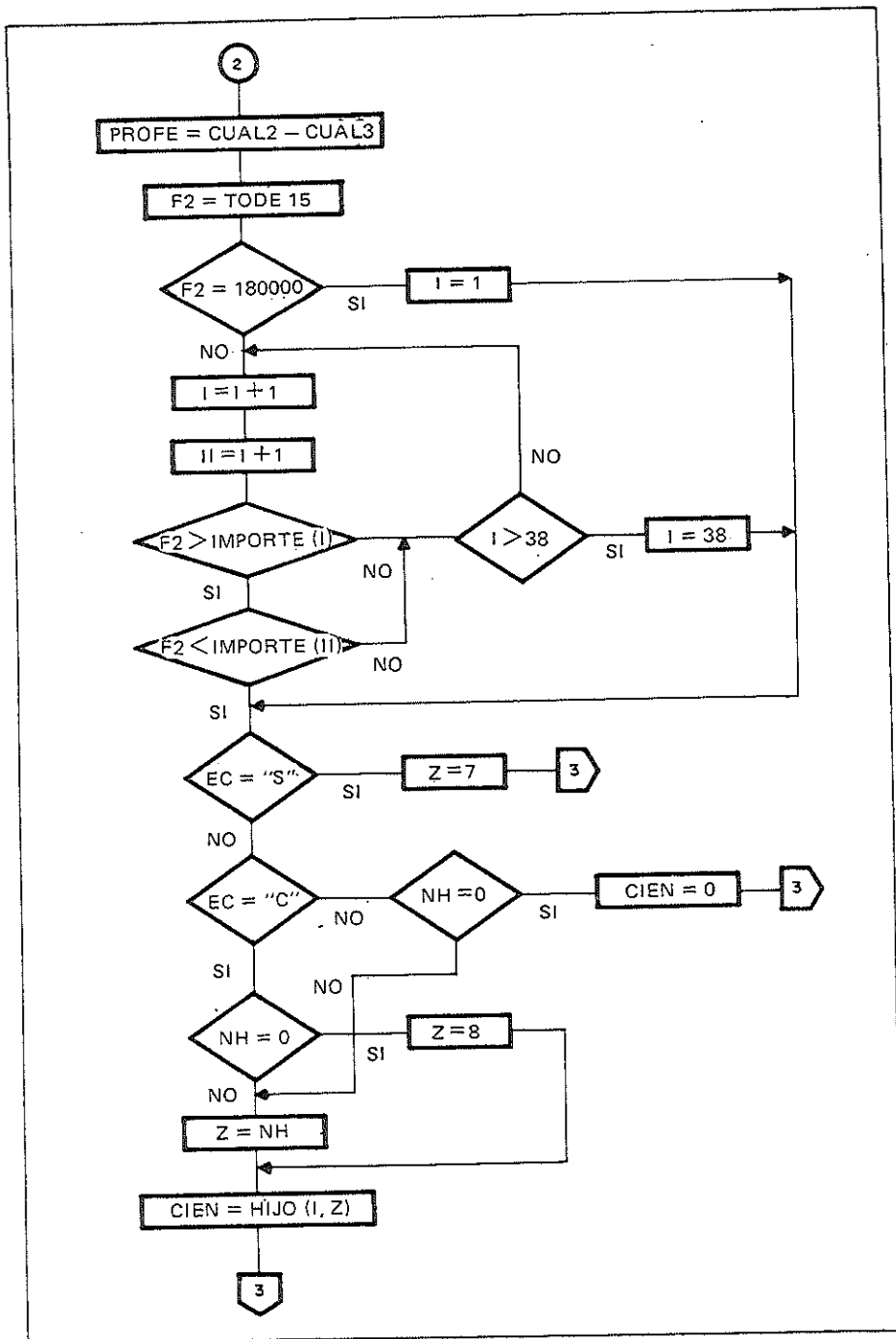
F2 = En ella llevamos el importe de la retribución anual.
 CIEN = La cargamos con el tanto por ciento que va a descontar para hacienda. La movemos a CIENL.
 IRTPL = Variable de salida a la que moveremos la variable de proceso TODOE.
 IRTPF = La cargamos con deducción hacienda. La movemos a IRTPL.
 DEDU = Total a deducir, la movemos DUL.
 PERCI = Se carga con el líquido total a percibir. La movemos a PERCIL.
 EMPRESA = Nombre de la empresa. Movemos a variable de salida EMPRL.
 DCM = Constante que moveremos a PROL. Dirección empresa.
 NISS = Constante que moveremos a EMPRL (Nº S.S. empresa).
 SBL = Contendrá el sueldo base del empleado. Es variable de salida.
 A-O2 = Contendrá el sueldo base del empleado. V. de salida.
 A-o2 = Contendrá los años de antigüedad del empleado. Lo calcularemos restándole el año actual el año de entrada a la empresa.
 ANTI = Variable que cargamos multiplicando el plus de antigüedad, por los años de antigüedad. Lo movemos a la V. de salida ANTIL.

ORGANIGRAMA GENERAL



ORGANIGRAMA DETALLADO





En esta segunda aplicación, vamos a realizar el control de los alumnos de un curso de un Centro de estudios.

Llevaremos los datos de los alumnos y sus notas en cada una de las asignaturas, por evaluación.

Realizamos los cinco programas siguientes: >>

1. Creación del fichero de alumnos

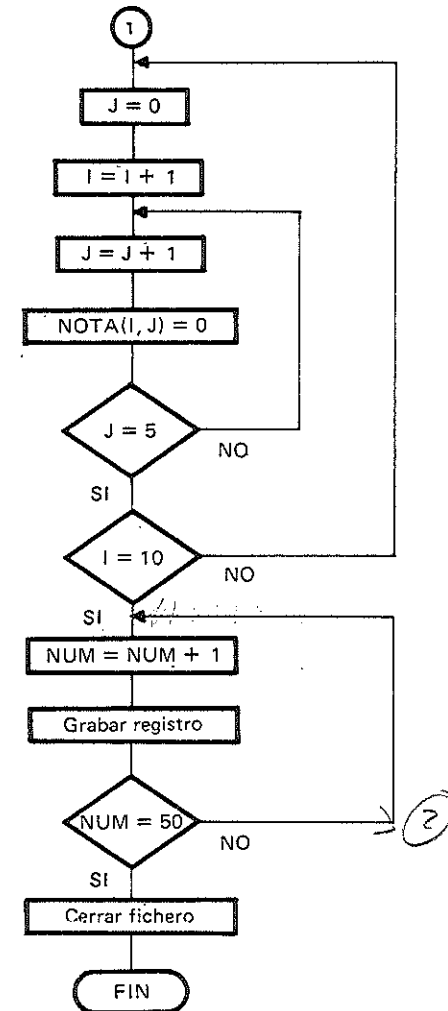
Este fichero lo creamos rellenándolo a ceros y a espacios, para evitar errores a la hora de la grabación. Guardamos espacio para 50 registros.

El formato del registro es:

/NOMBRE / DIRECCION / TELEFONO / NOTAS /

La variable NOTAS, será una tabla de cinco columnas, que corresponden a las asignaturas del curso.

ORGANIGRAMA



2. Adición de registros en el fichero de alumnos

En este programa introducimos los datos del nuevo alumno por pantalla y a continuación los grabamos en el fichero.

Como el fichero es de acceso directo, sabremos que el registro está vacío cuando el campo nombre esté a espacios.

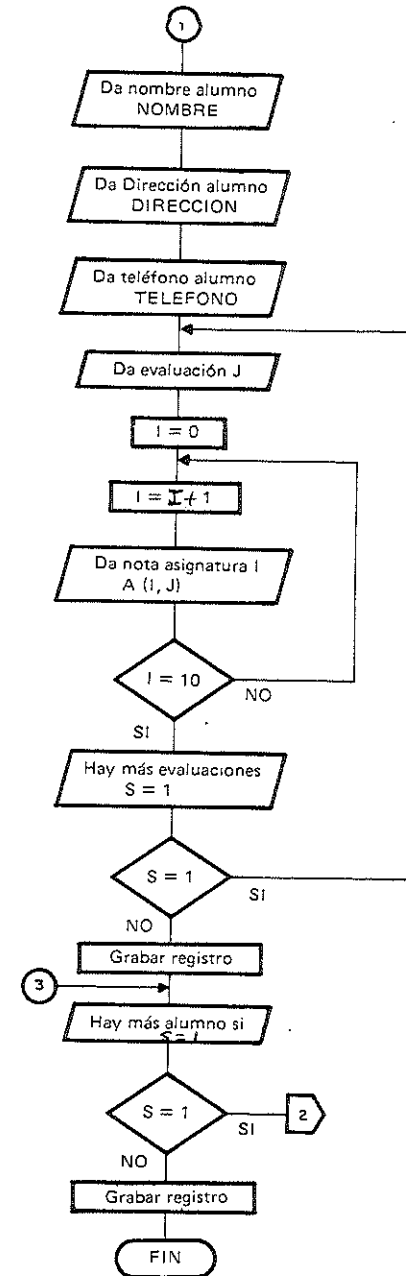
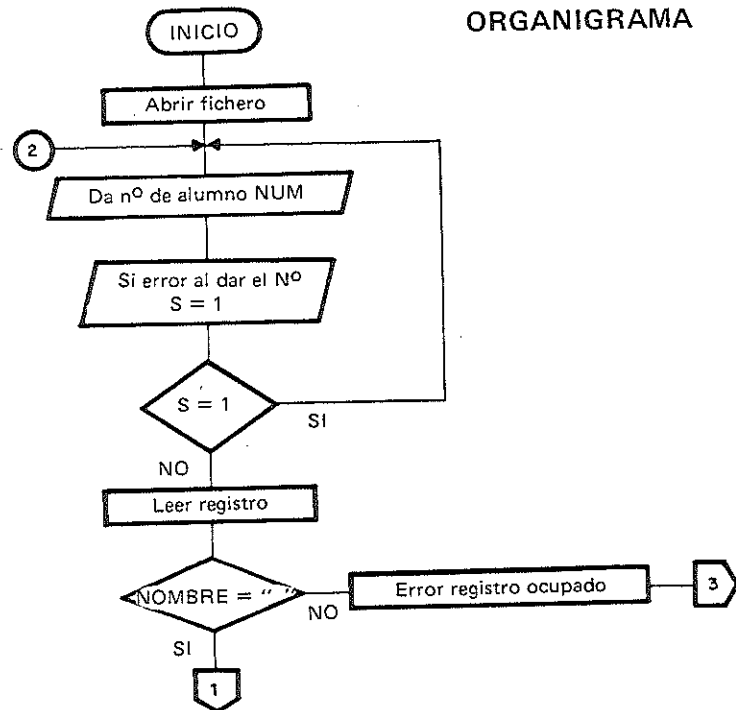
Lista de variables.

S = Esta variable la utilizamos para meter el valor 1 en el caso de que nos hayamos equivocado.

También la utilizaremos para indicar que hemos terminado el proceso de inserción.

I = Esta variable me indicará la fila de la variable NOTAS.

J = Esta variable me indicará la columna de la variable NOTAS.



CONSULTA

3. Eliminación y modificación del fichero de alumnos

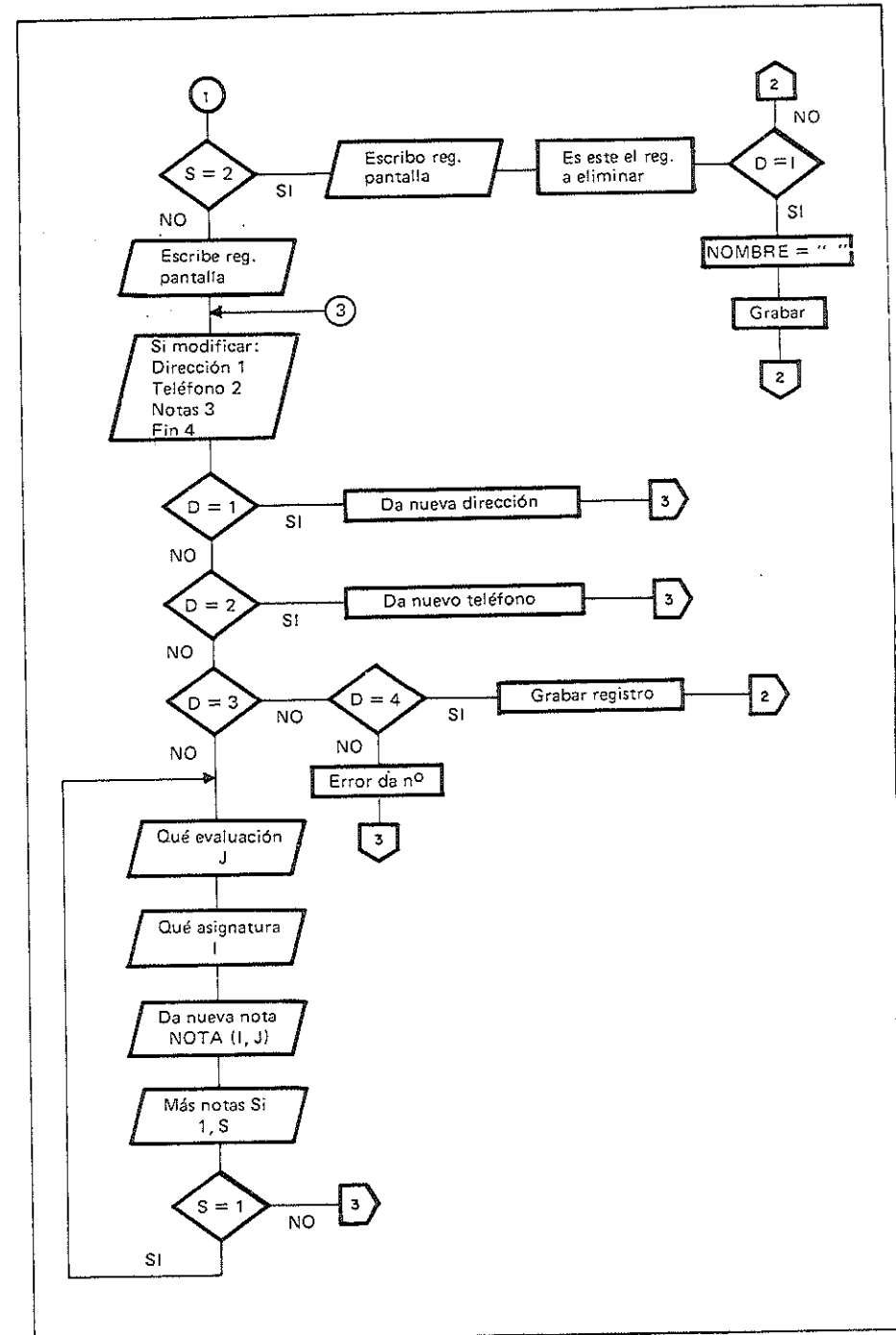
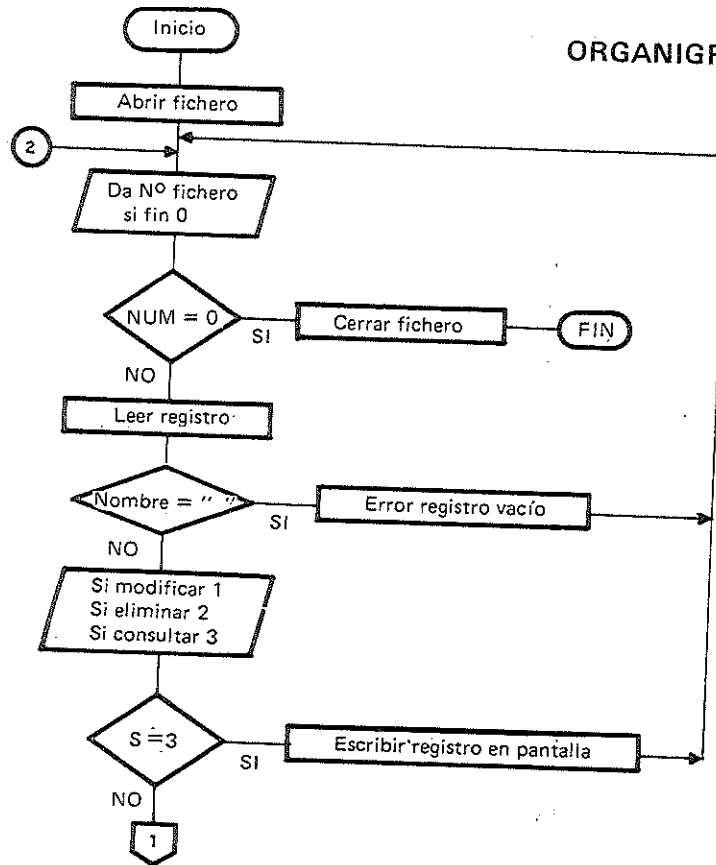
Hemos confeccionado un menú de forma que nos aparecen por pantalla las distintas opciones, y nosotros indicamos con un numero la que nos interesa en ese momento.

Lista de variables

S = esta variable la utilizamos para seleccionar la opción deseada.

D = esta variable la utilizamos para el caso de modificación que nos indique el campo a modificar.

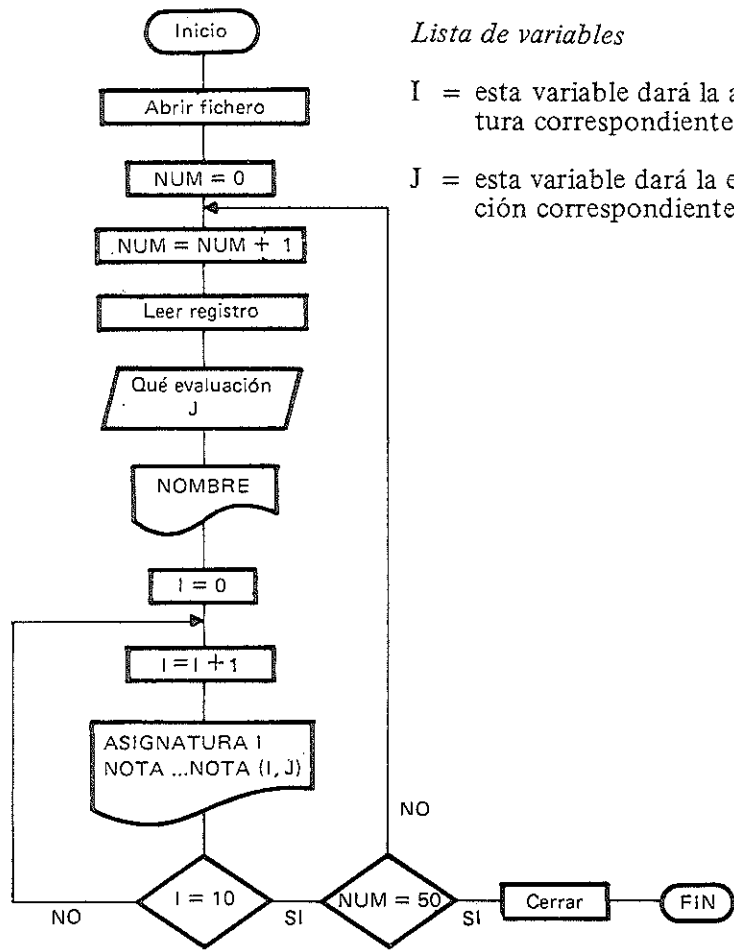
ORGANIGRAMA



4. Listado de notas por alumno

En este programa vamos a realizar el listado de los alumnos de la clase con sus notas en todas las asignaturas, de una determinada evaluación.

ORGANIGRAMA



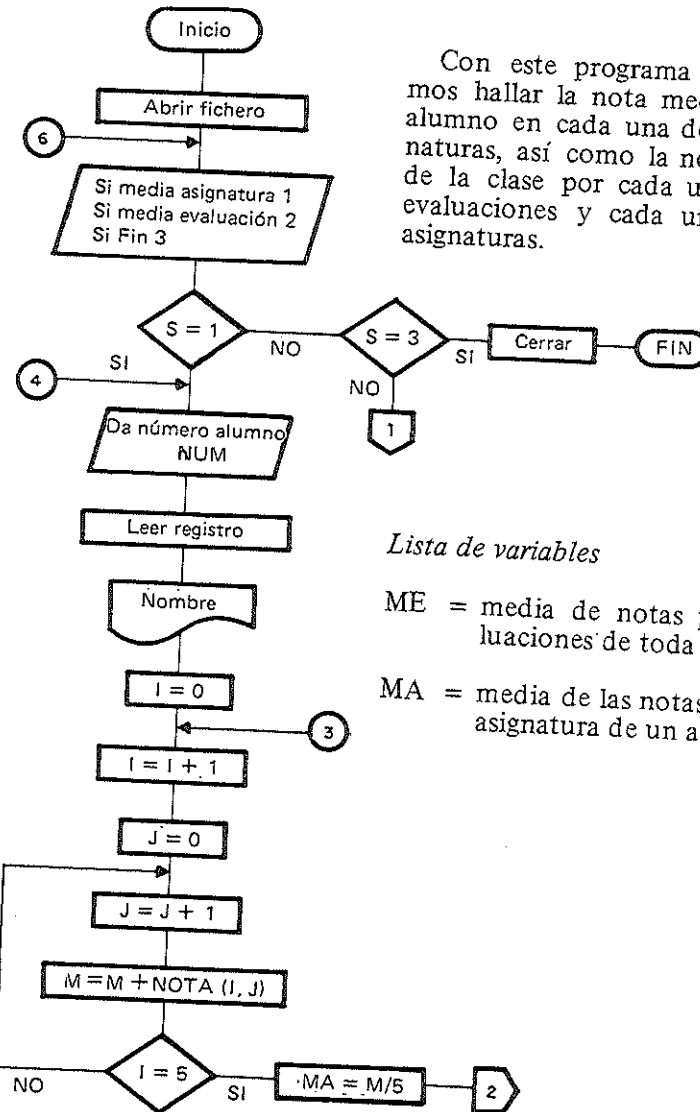
Lista de variables

I = esta variable dará la asignatura correspondiente.

J = esta variable dará la evaluación correspondiente.

5. Media de notas por evaluación y media de notas por asignatura

ORGANIGRAMA

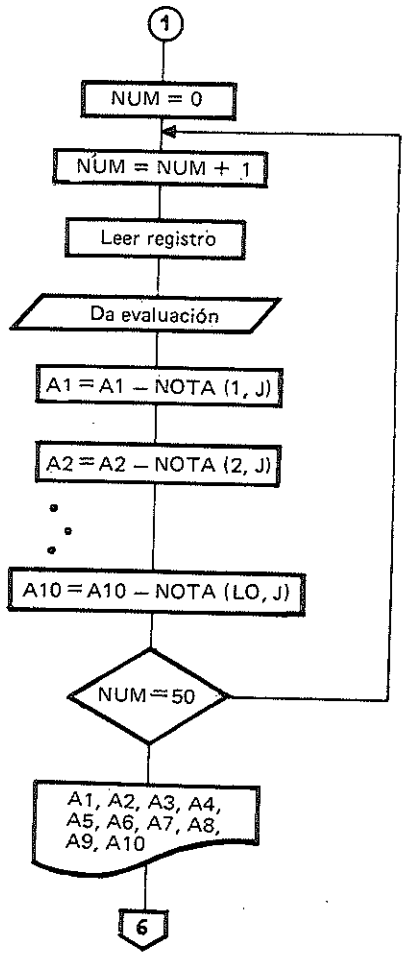
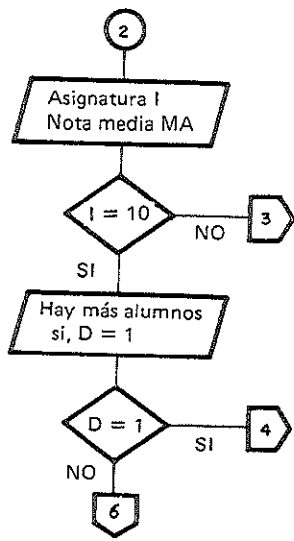


Con este programa pretendemos hallar la nota media de un alumno en cada una de sus asignaturas, así como la nota media de la clase por cada una de las evaluaciones y cada una de las asignaturas.

Lista de variables

ME = media de notas por evaluaciones de toda la clase.

MA = media de las notas de una asignatura de un alumno.



En esta tercera aplicación vamos a realizar los programas necesarios para efectuar Estadísticas de tipo I, II y III.

ESTADISTICA TIPO I

1. *Media aritmética*: Se obtiene dividiendo la suma de cada valor distinto de la variable por el número de observaciones.

$$X = \frac{\sum x_i}{N}$$

2. *Media aritmética ponderada*: La fórmula de ésta es:

$$X_p = \frac{\sum x_i p_i}{\sum p_i}$$

3. *Mediana*: Es el valor que ocupa el lugar central, es decir, al que deja por encima y por debajo el mismo número de observaciones.

4. *Moda*: Es el valor de la variable que se repite más veces.

5. *Desviación típica*:

$$S = \sqrt{\frac{\sum x_i^2}{N} - \left(\frac{\sum x_i}{N}\right)^2}$$

6. *Media geométrica*:

$$MG = \sqrt[N]{X_1 \cdot X_2 \cdot \dots \cdot X_n}$$

7. *Media cuadrática*:

$$MC = \sqrt{\frac{\sum (X_i - \bar{X})^2}{N}}$$

8. *Variancia:*

$$V = S^2$$

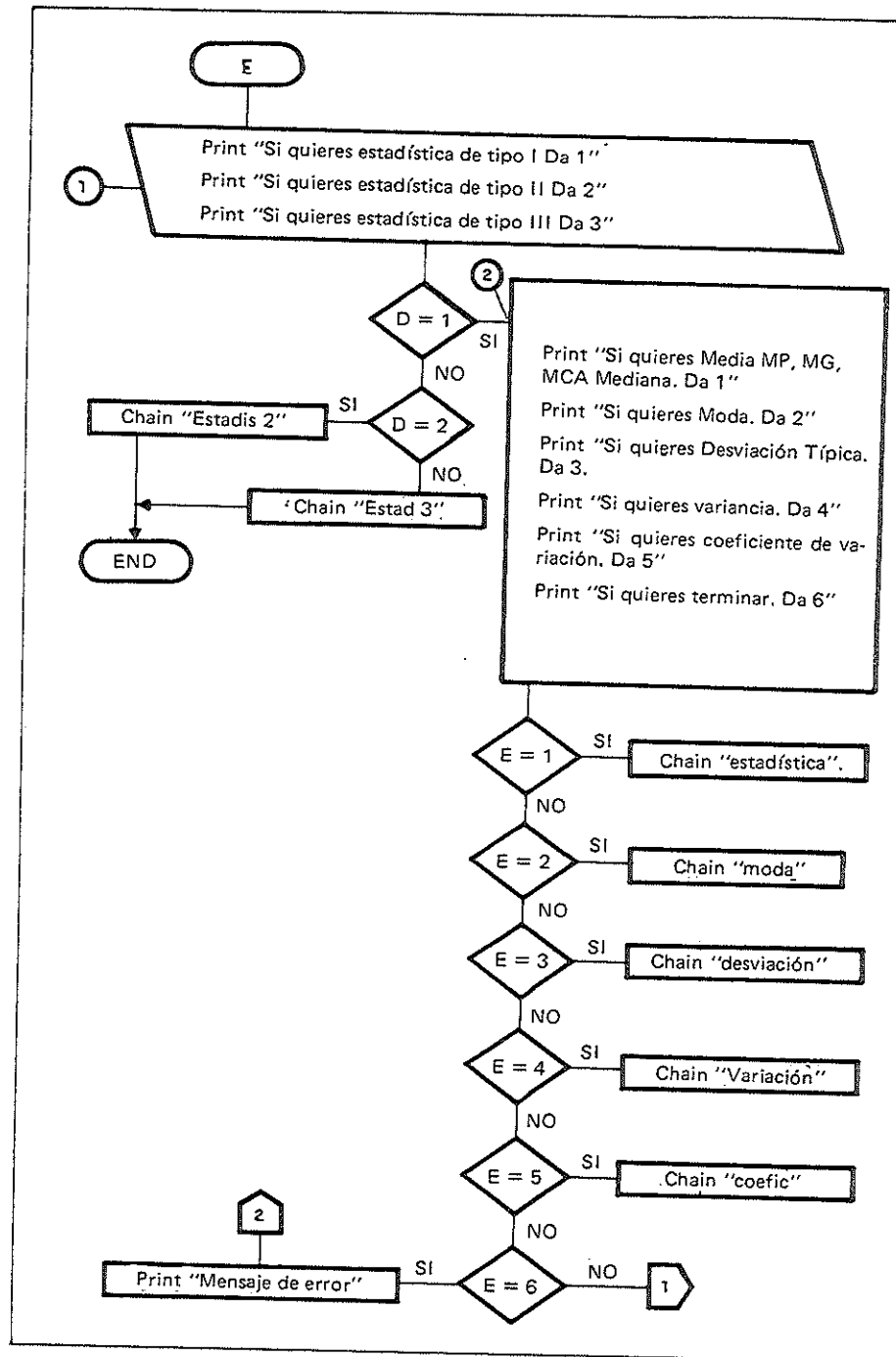
9. *Coficiente de variación:*

$$CV = \frac{S}{X}$$

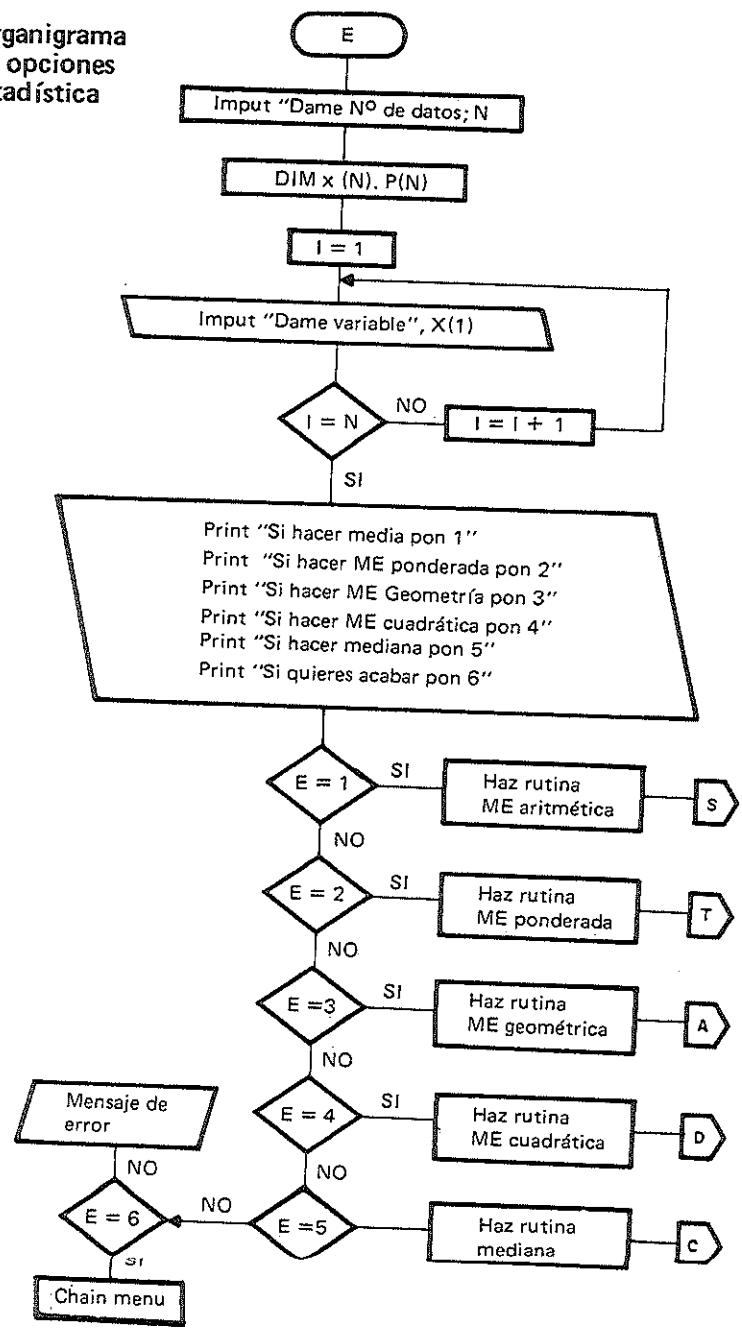
ESTADISTICA TIPO I

Listados de variables

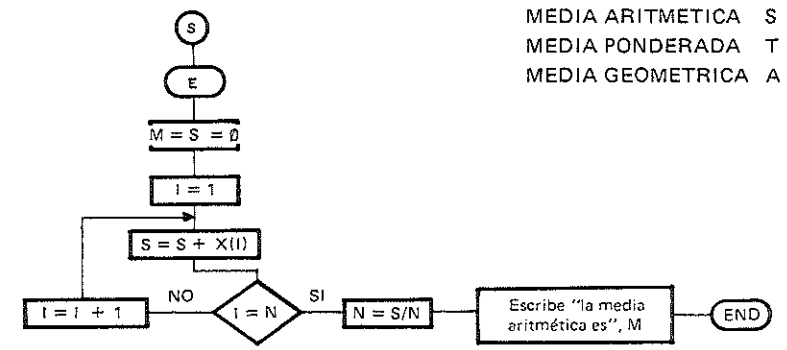
- N = Número de observaciones.
- X = valor de la variable.
- P = valor de la ponderación de la variable.
- M = valor de la media aritmética.
- S = sumatorio de las variables.
- H = sumatorio de las ponderaciones.
- Pl = sumatorio del producto de las ponderaciones por su variable.
- MP = valor de la media ponderada.
- MG = valor de la media geométrica.
- C = valor de las observaciones partido por dos.
- T = Lleva la parte entera de C.
- ME = valor de la mediana.
- W = sumatorio de los cuadrados de las variables.
- D = W dividido por el número de observaciones.
- S = desviación típica.
- V = desviación típica al cuadrado.
- CV = coeficiente de variación.



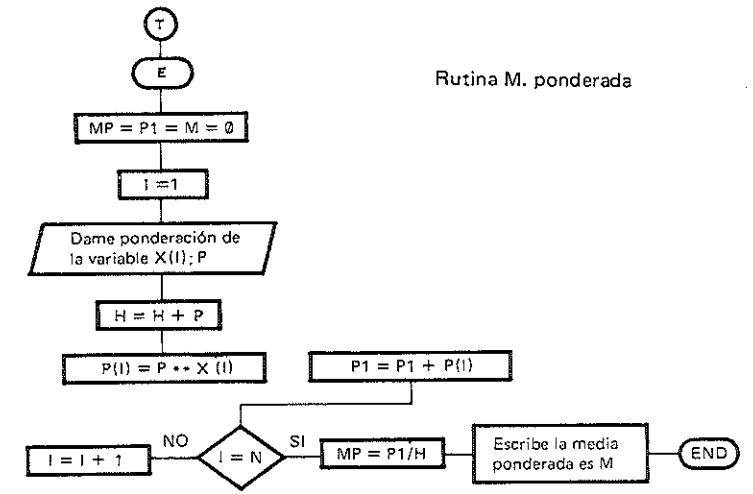
Organigrama de opciones estadística



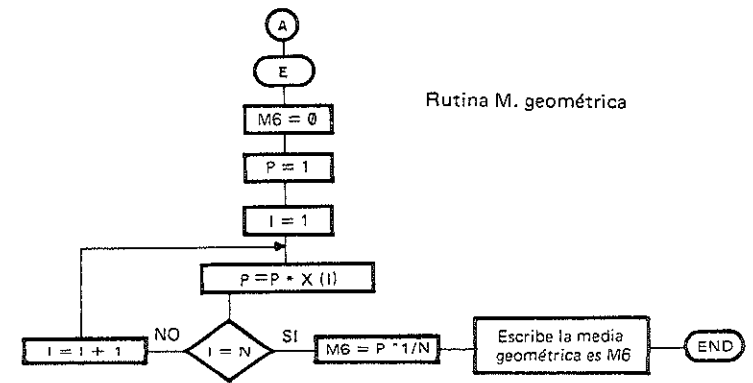
MEDIA ARITMETICA S
 MEDIA PONDERADA T
 MEDIA GEOMETRICA A

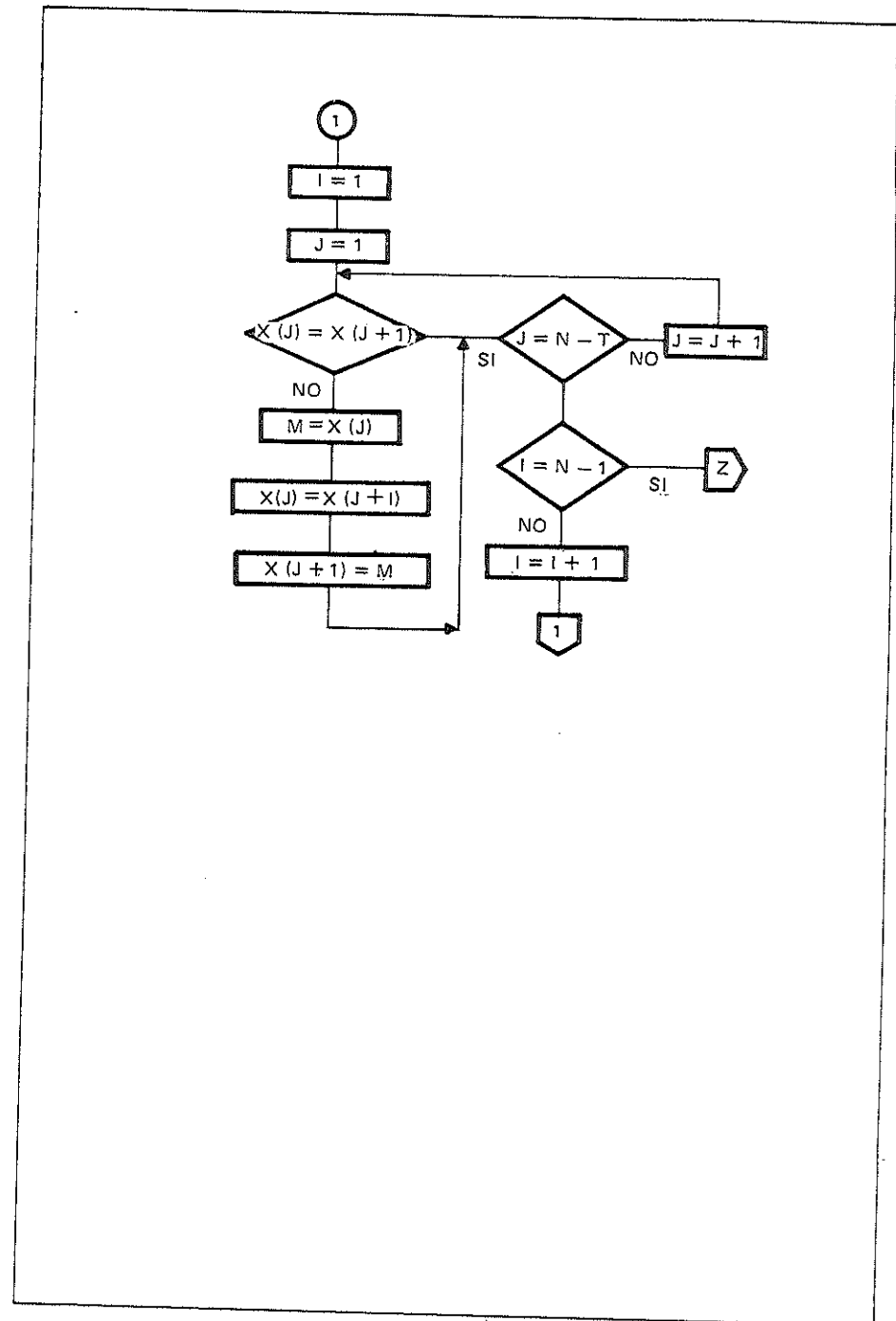
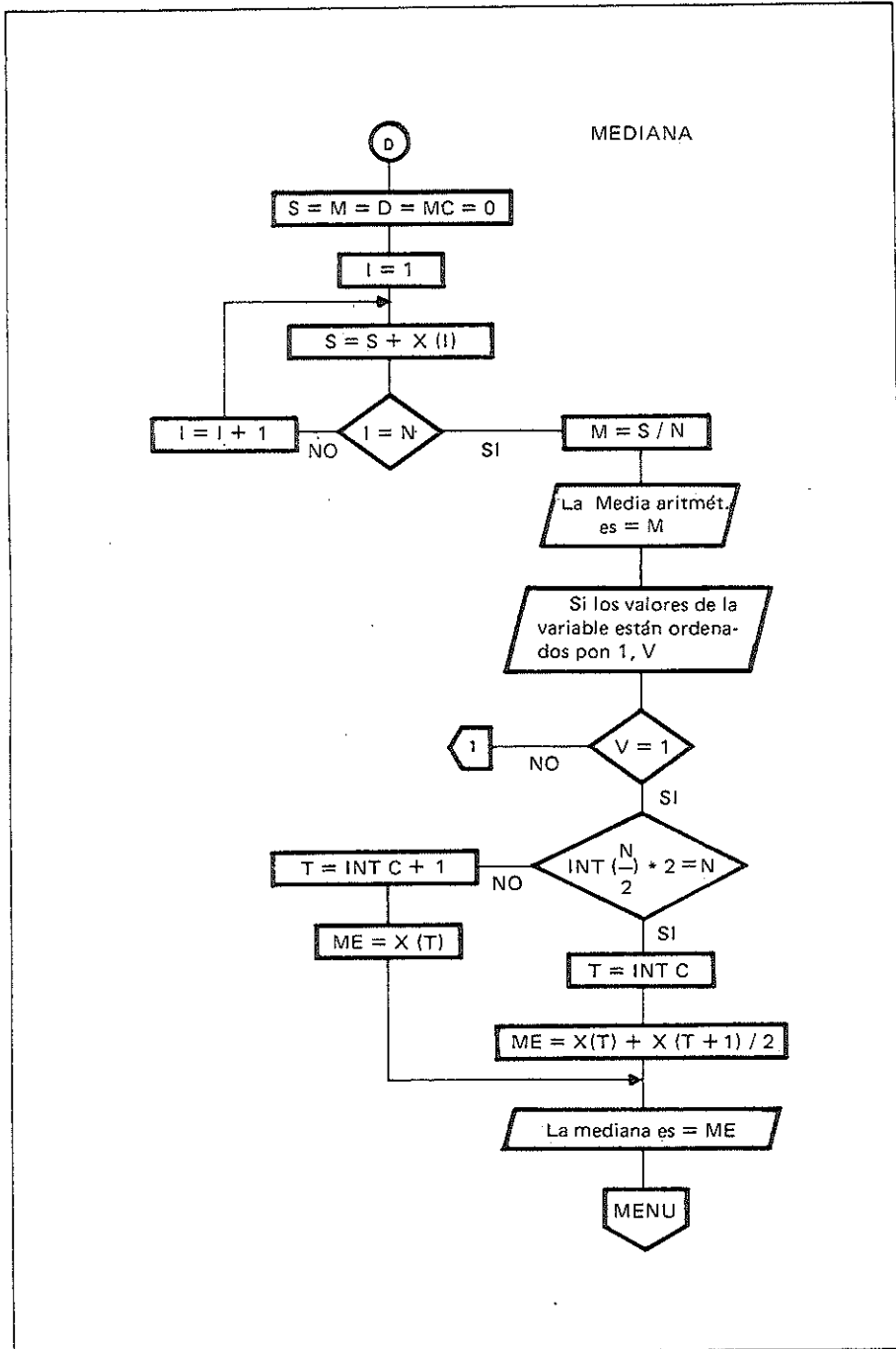


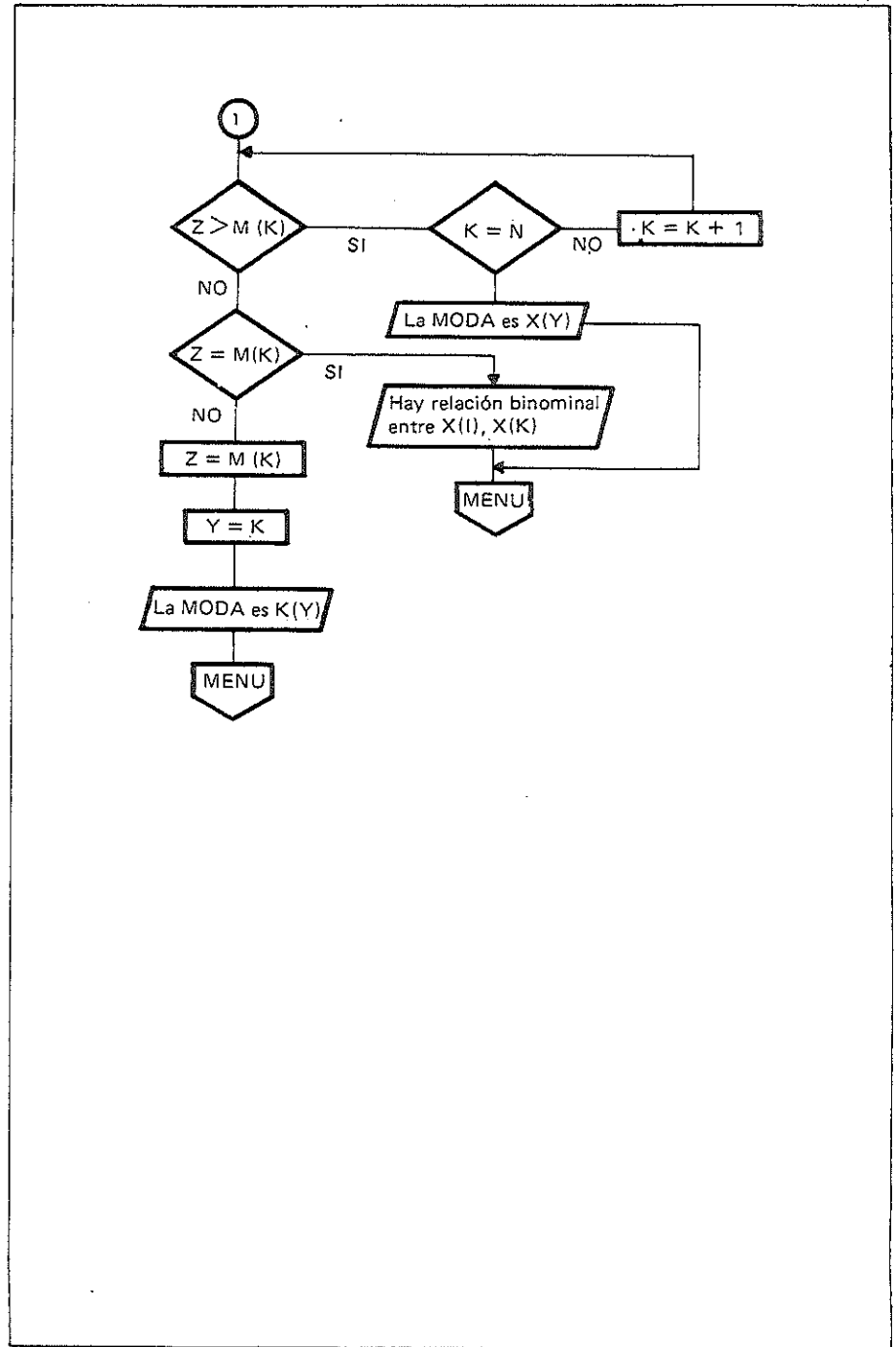
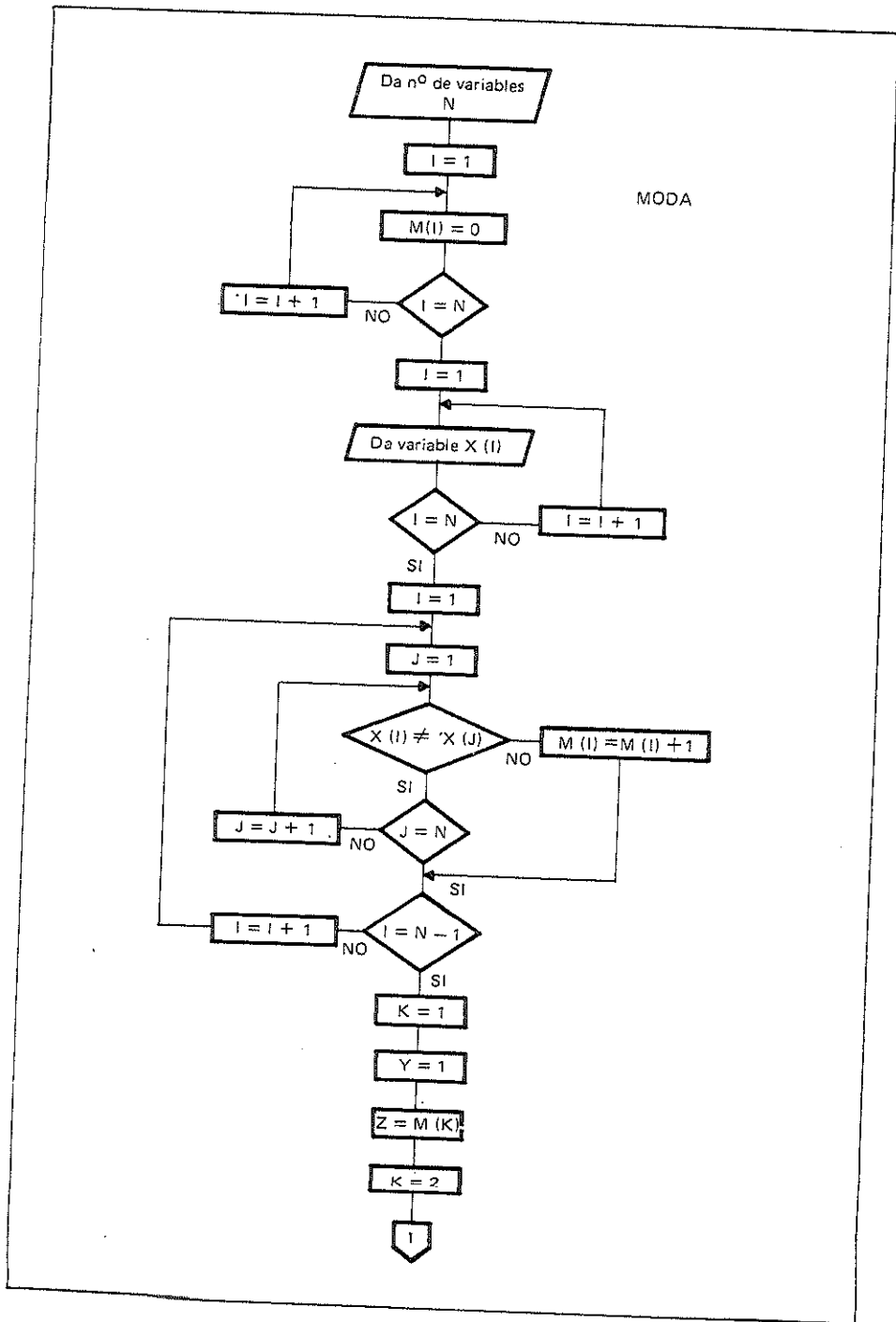
Rutina M. ponderada

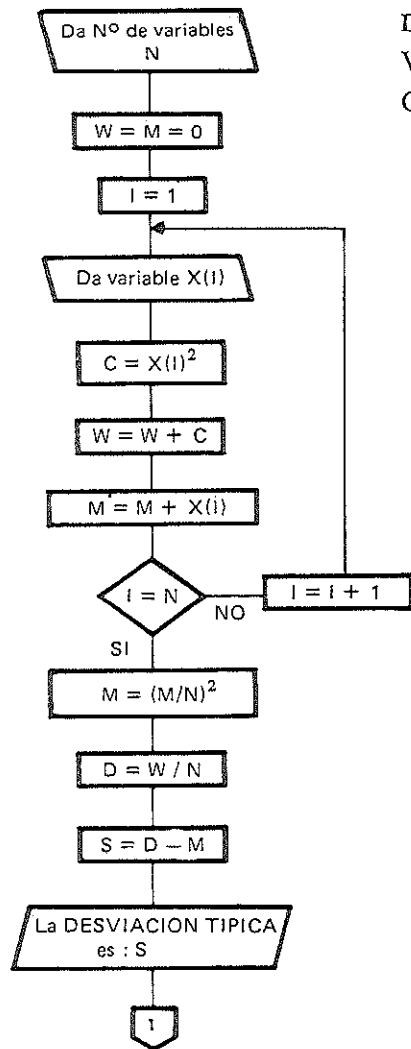


Rutina M. geométrica

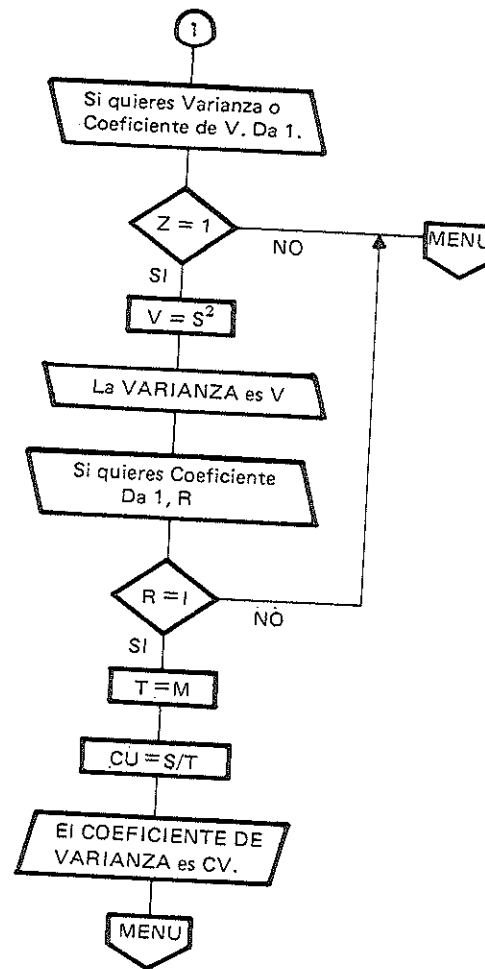








Desviación típica.
 Varianza.
 Coeficiente de varianza.



ESTADISTICA TIPO II

Es el programa "estadis 2" que corresponde a los cálculos de ESTADISTICA II y realiza los promedios correspondientes a éstas, que son:

1. Media
2. Mediana
3. Moda
4. Desviación típica
5. Variancia
6. Coeficiente de variancia

Cuando la *media* es requerida por operador, el programa consulta el número de observaciones, y según esté dimensiona las tablas correspondientes. Después pide los valores de las variables y las frecuencias con que éstas se repiten y efectúa el siguiente cálculo:

$$ME = \frac{\sum V(I) * F(I)}{N}$$

La opción 2 corresponde al cálculo de la *moda* y efectúa en principio lo mismo que la *media*, ya que pide el número de observaciones y con este número pide la variable. El cálculo de la moda consiste, mediante dos bucles, en obtener la mayor frecuencia de la tabla de frecuencias y la posición de ésta. Con la posición nos dirigiremos a la tabla de las variables, con lo que obtendremos el valor moda.

La opción 3 calcula la mediana. Como datos se requieren el número de observaciones y las variables. La mediana será el valor intermedio de la tabla de variables. Si el número de éstas fuese impar, se toma el inmediato superior.

La opción 4 calcula la desviación típica, varianza y coeficiente de variación, una vez proporcionadas las variables y frecuencias de estas, teniendo en cuenta la siguiente fórmula:

$$Z = \sqrt{[\sum V(H)^2 * F(H)] - [\sum F(H) * V(H)]}$$

siendo la Variancia = $Z \uparrow 2$

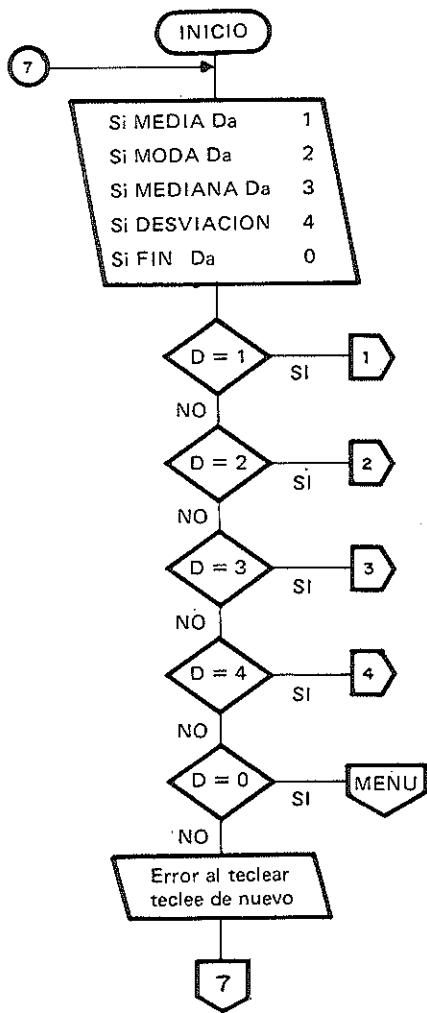
$$T = \frac{\sum V(I) * F(I)}{N}$$

Siendo el Coeficiente de variancia = Z/T .

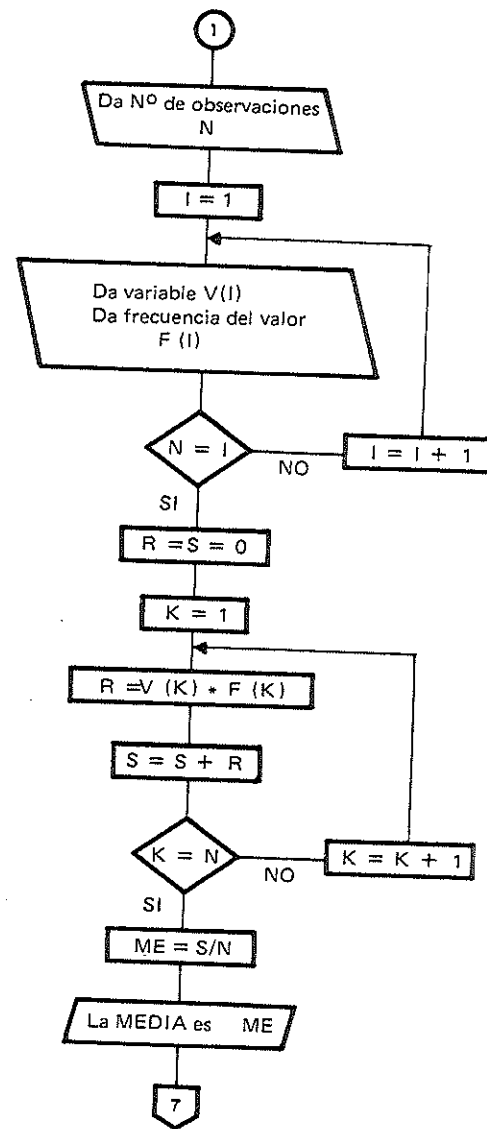
Listado de variables del programa "ESTADIS 2"

- V = variables de cualquier aplicación.
- F = frecuencias de las variables.
- N = número de observaciones.
- CX = valor de la variable al cuadrado y multiplicado por la frecuencia.
- J = Contador para bucles.
- H = contador para bucles.
- L = posición para dirigirse al lugar de la tabla donde debemos buscar la moda.
- S = sumas acumuladas.
- T = media.
- Z = desviación típica.

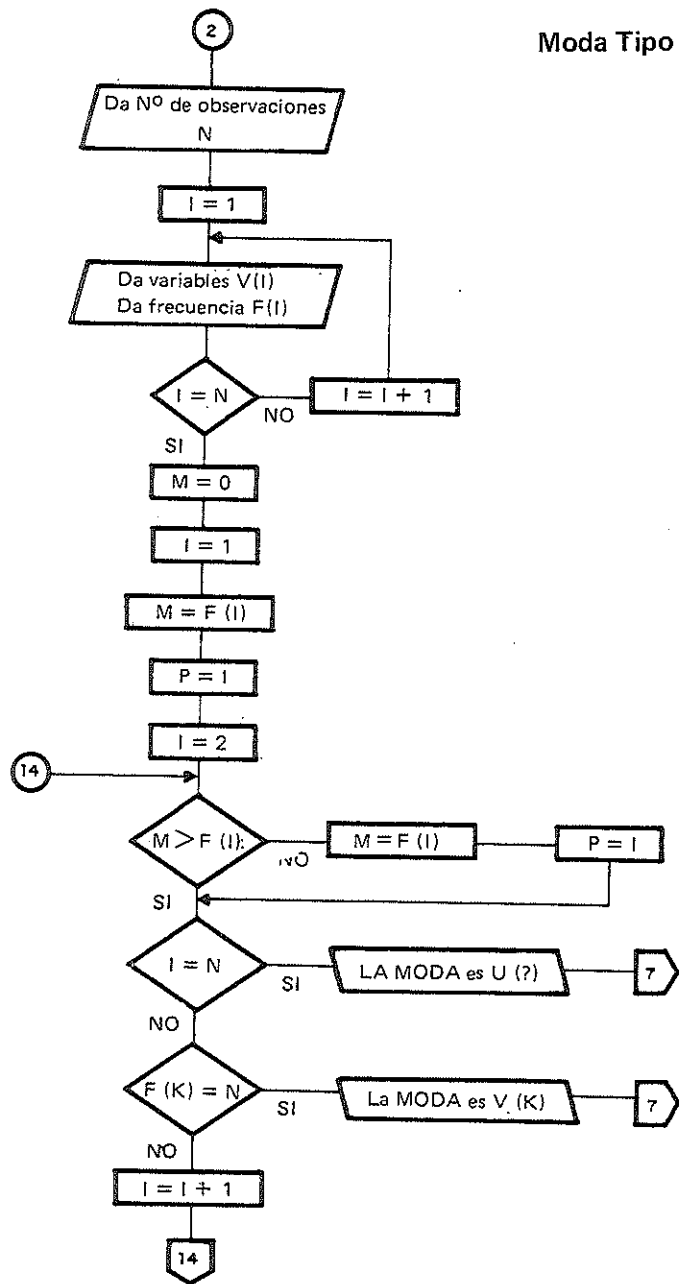
Programa menú para estadísticas de tipo II



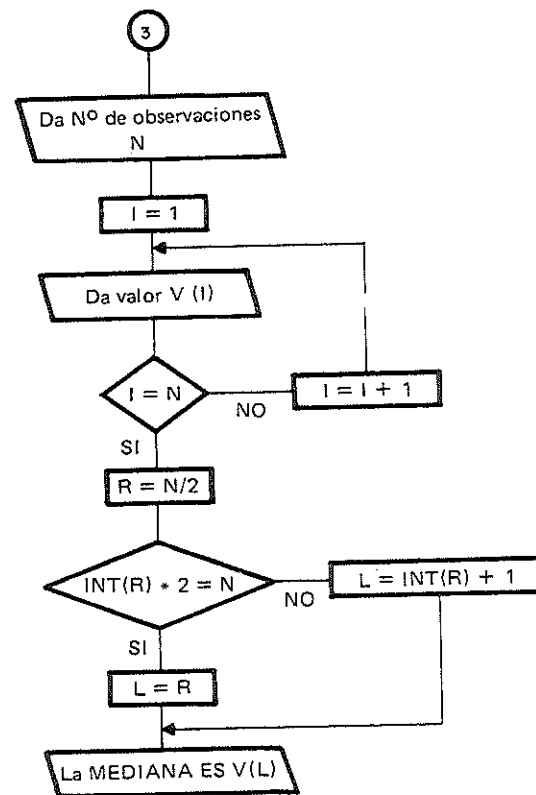
Media Tipo II



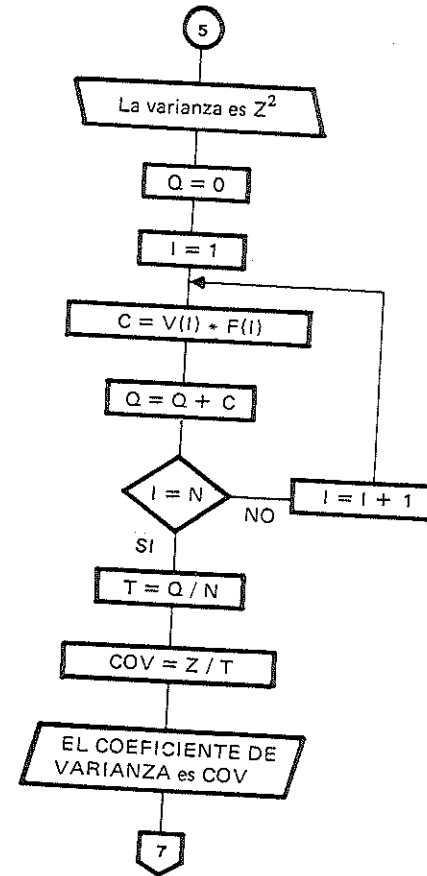
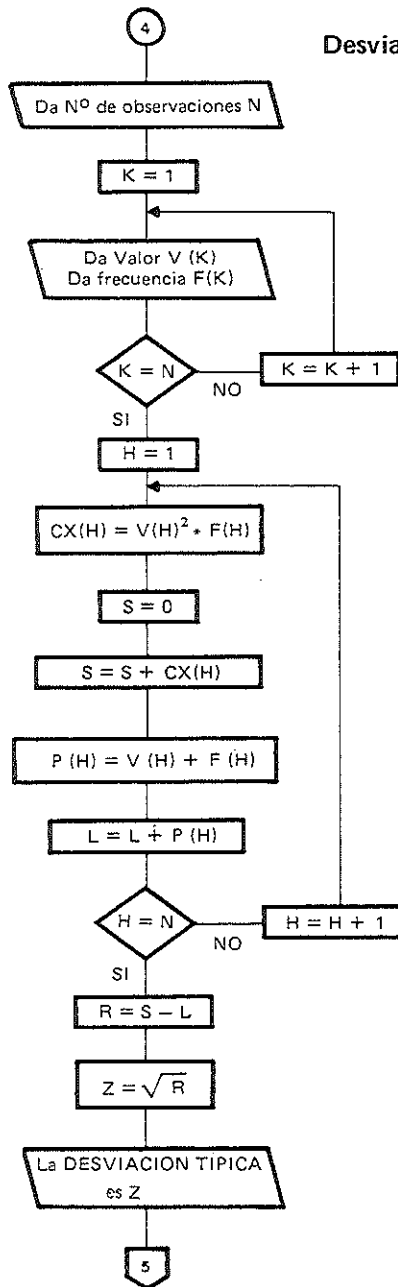
Moda Tipo II



Mediana tipo II



Desviación típica tipo II



ESTADISTICA TIPO III

Este programa calcula los promedios de la estadística de intervalos.

1. La media aritmética cuya fórmula general es:

$$\bar{X} = \frac{\sum Xi \mu_i}{N}$$

X_i es la media aritmética de cada uno de los intervalos, llamadas marca de clase.

$$X_i = \frac{\text{lim } 2 + \text{lim } 1}{2}$$

μ_i es la frecuencia absoluta de cada intervalo.

N es el número total de observaciones, es decir, la suma de todas las frecuencias absolutas.

2. La Mediana es el valor de la variable que divide a la tabla estadística en dos. Su fórmula es:

$$Me = \text{lim } (i - 1) + \frac{\frac{N}{2} - N(i - 1)}{\mu_i} \cdot a_i$$

$\text{Lim } (i - 1)$ es el límite inferior del intervalo correspondiente a la primera frecuencia acumulada, que es mayor que $N/2$.

M es el número de observaciones.

$N(i - 1)$ es la frecuencia acumulada del intervalo.

a_i es la amplitud del intervalo.

μ_i es la frecuencia absoluta de intervalo.

3. La Moda es el valor de la variable de mayor frecuencia. Su cálculo se efectúa por medio de la fórmula:

$$Mo = \text{lim } (i - 1) + \frac{\mu(i + 1)}{\mu(i - 1) + \mu(i + 1)} \cdot a_i$$

$\text{Lim } (i - 1)$ es el límite inferior del intervalo de mayor frecuencia absoluta.

$\mu(i - 1)$ y $\mu(i + 1)$ son las frecuencias anterior y posterior de la línea i .

a_i es la amplitud del intervalo.

Es importante destacar que esta fórmula no podrá ser aplicada si los intervalos no son de la misma amplitud.

4. La desviación típica es una medida de dispersión que nos indica la concentración de la serie de valores respecto a su media aritmética. Su cálculo se efectúa por medio de la fórmula:

$$S = \sqrt{\frac{\sum Xi^2 \cdot \mu_i}{N} - \left(\frac{\sum Xi \mu_i}{N}\right)^2}$$

X_i^2 es la suma de clase de cada intervalo al cuadrado

$\left(\frac{\sum Xi \mu_i}{N}\right)^2$ es la medida aritmética al cuadrado.

5. La Varianza es la desviación típica elevada al cuadrado.

6. Coeficiente de Variación es la desviación típica dividida por la Media Aritmética.

7. Los cuartiles, dividen a la serie de valores en cuatro partes. La fórmula para su cálculo es igual que la de la mediana, a diferencia de que la línea buscada se calcula: $nN/4$ en donde n es igual al cuartil a buscar.

$$Qu = \text{lim } (i - 1) + \frac{\frac{u \cdot N}{4} - N(i - 1)}{\mu_i} \cdot a_i$$

8. Los deciles dividen a la serie de valores en diez partes, por lo que la línea i buscada en $\mu N/10 = i$.

$$Du = \lim (i - 1) - \frac{\frac{u \cdot N}{4} - N(i - 1)}{u_i} ai$$

Los intervalos van en variables alfanuméricas y se deben de meter de la siguiente forma:

$$\text{Lim } (i - 1) - \text{Lim } (i + 1)$$

Listado de variables

- B = número de intervalos.
- T\$ = tabla de intervalos de las variables.
- F = tabla de frecuencias absolutas.
- X = tablas de la marca de clase.
- X2 = tabla de la marca de clase al cuadrado por la frecuencia.
- L = tabla para llevar las amplitudes de los intervalos.
- Z = tabla de las frecuencias acumuladas.
- N1 = longitud del límite inferior.
- N2 = longitud del límite superior.
- M1\$ = límite inferior alfanumérica.
- L1 = límite inferior convertido a numérico.
- L2 = límite superior convertido a numérico.
- M2\$ = límite superior alfanumérico.
- M = sumatorio de $X_i \cdot n_i$; y la frecuencia absoluta mayor.
- D = H = Z = sumatorio de la frecuencia absoluta.
- ME = media aritmética.
- N = la línea de mayor frecuencia.
- MD = mediana.
- L = amplitud de los intervalos.
- MO = moda.
- S1 = x_i al cuadrado.

S2 = marca de clase al cuadrado multiplicado por su frecuencia.

HE = sumatorio de S2 dividido por D.

S = desviación típica.

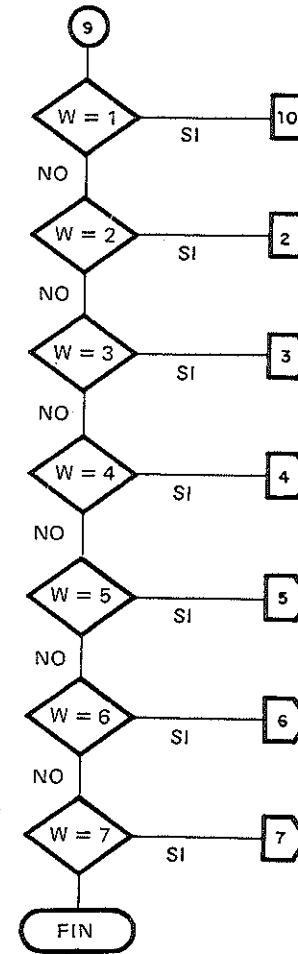
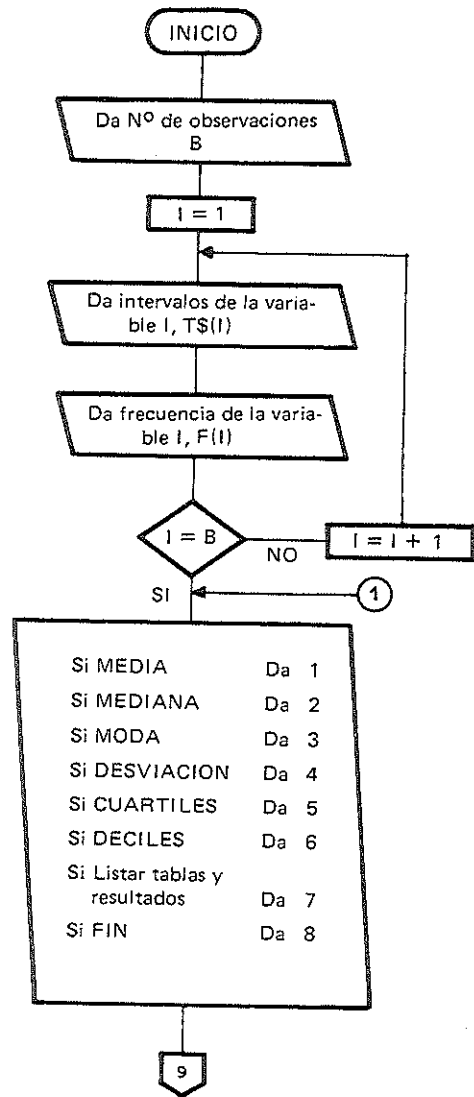
V = Variancia.

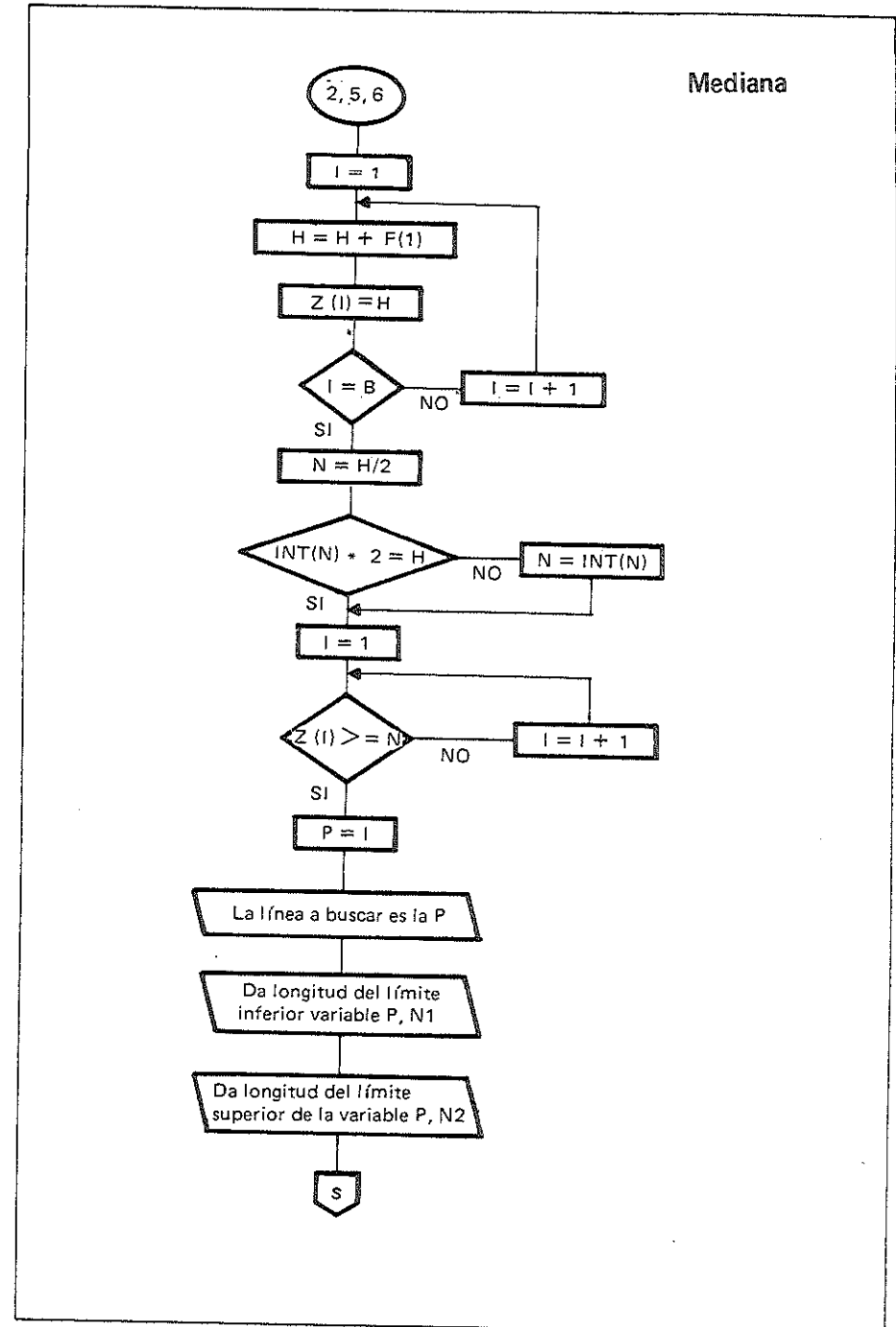
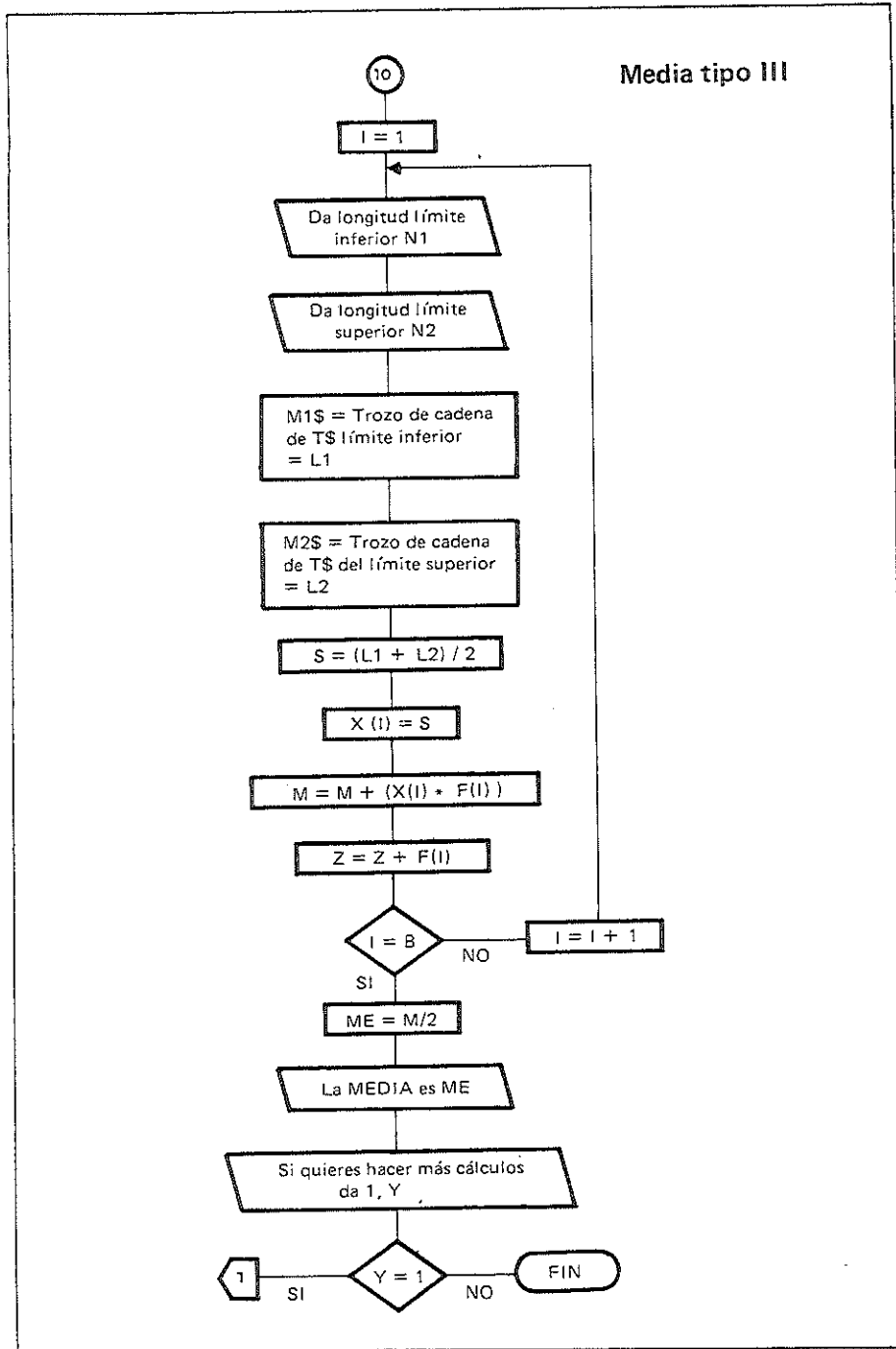
CV = coeficiente de variación.

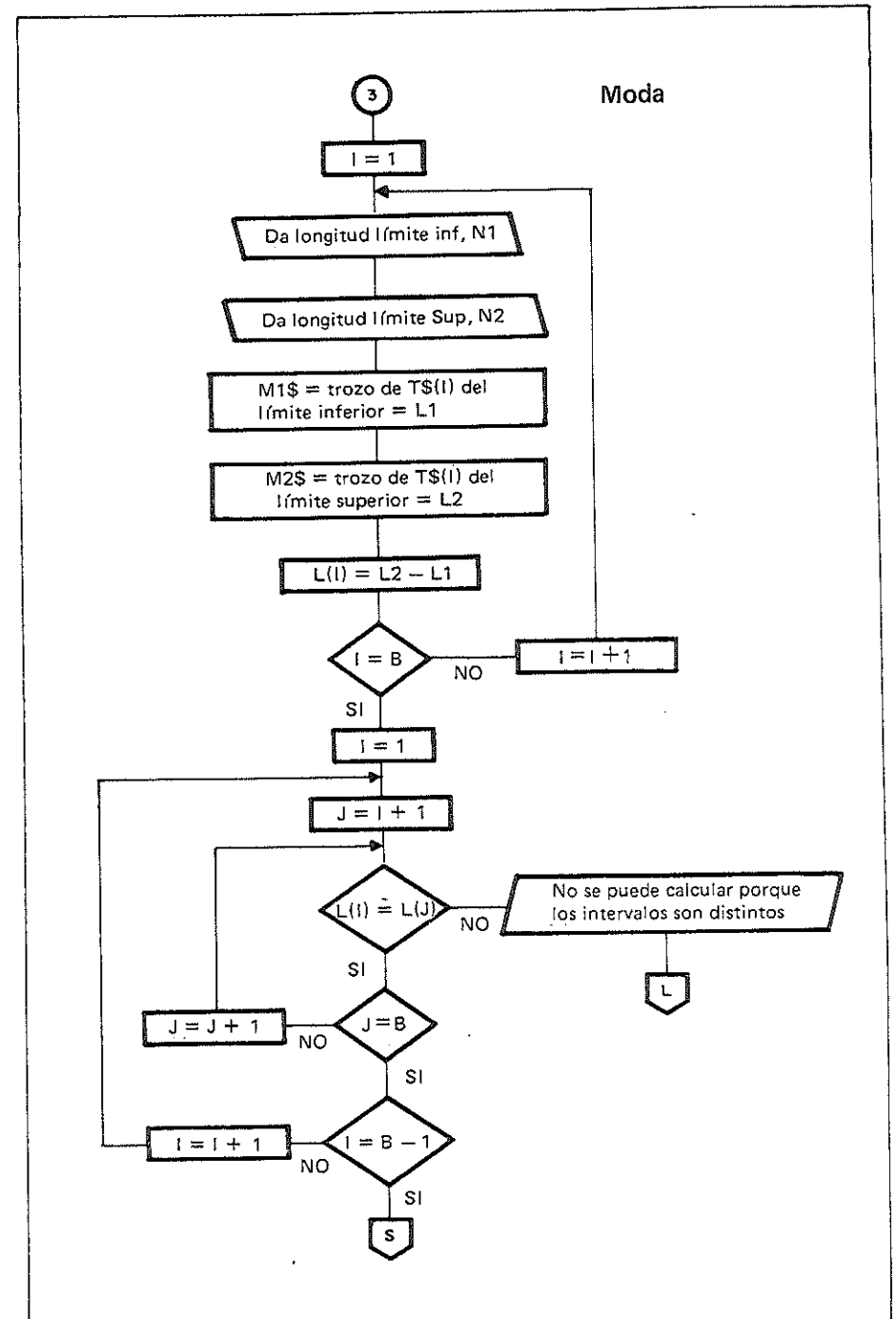
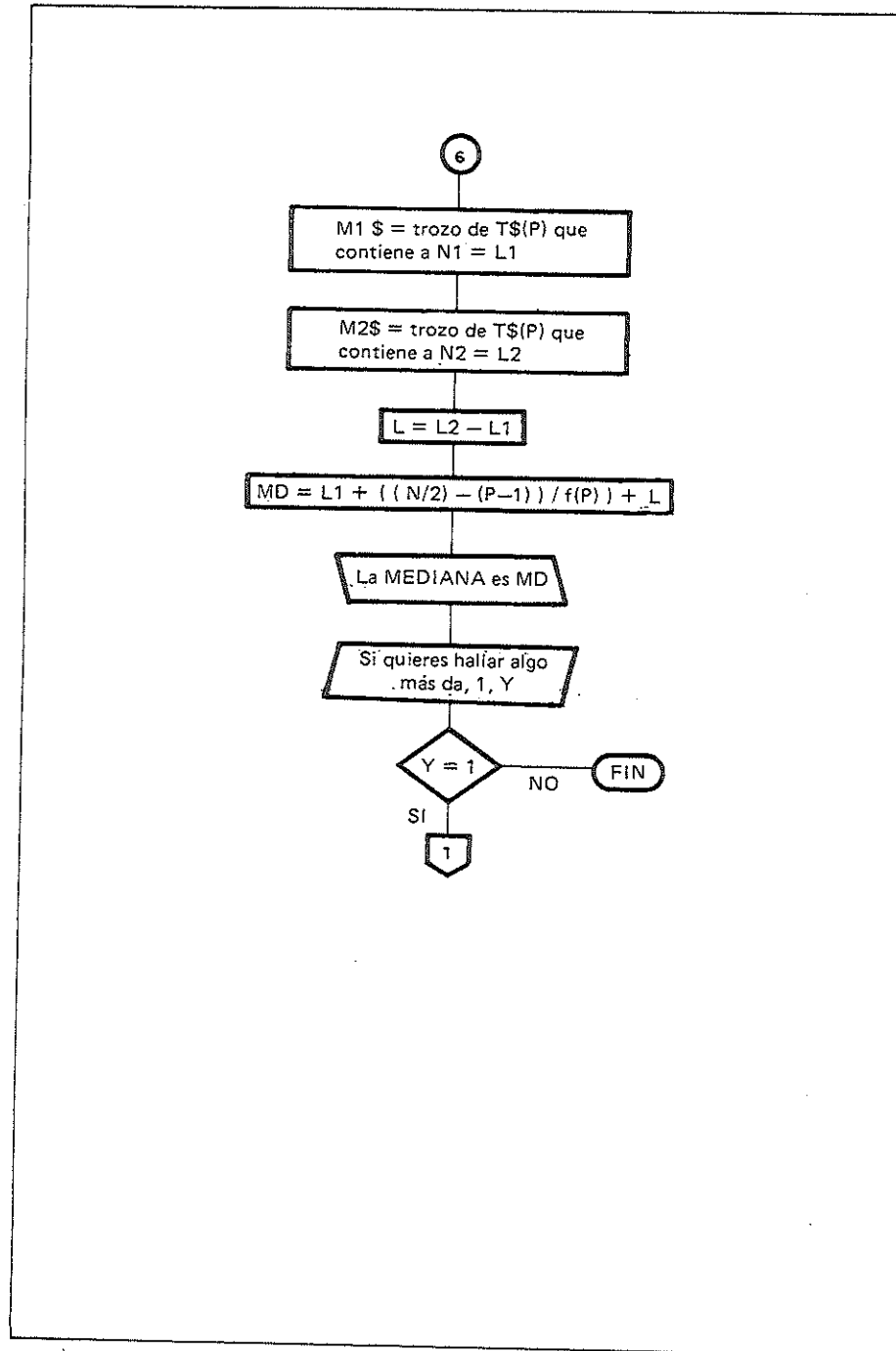
Q = tabla de cuartiles.

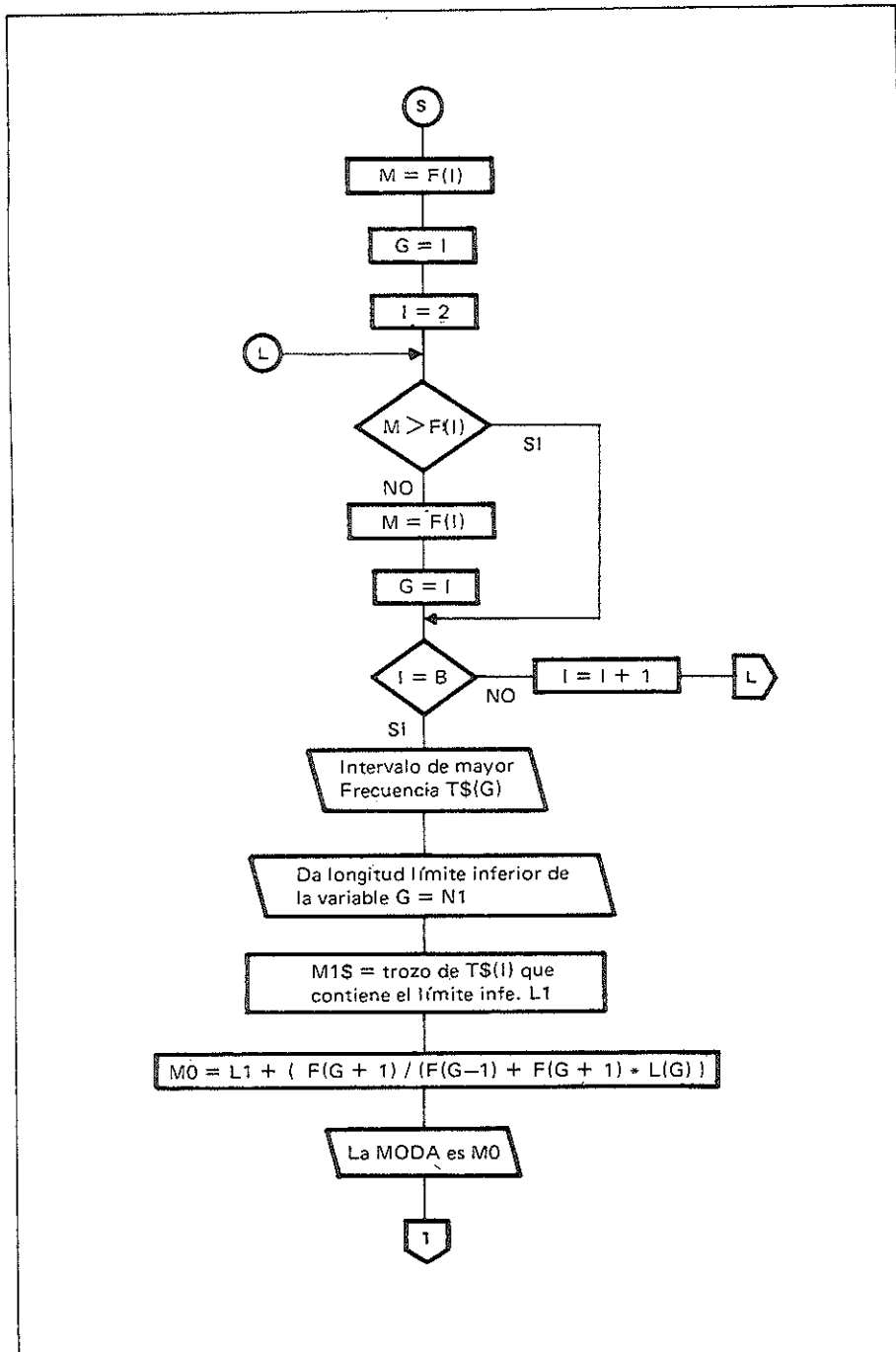
DE = tabla de deciles.

Programa menú para estadísticas de tipo III

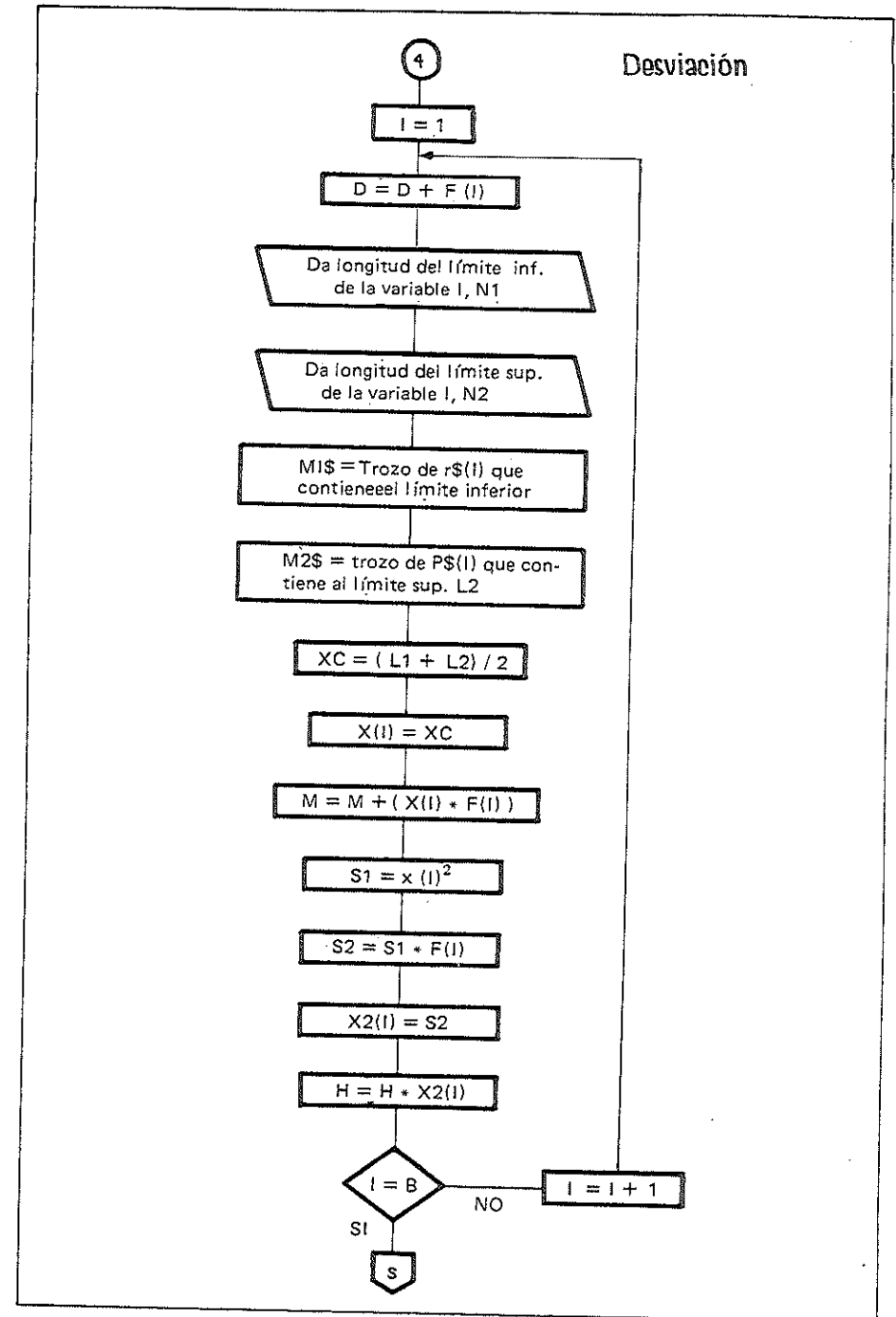


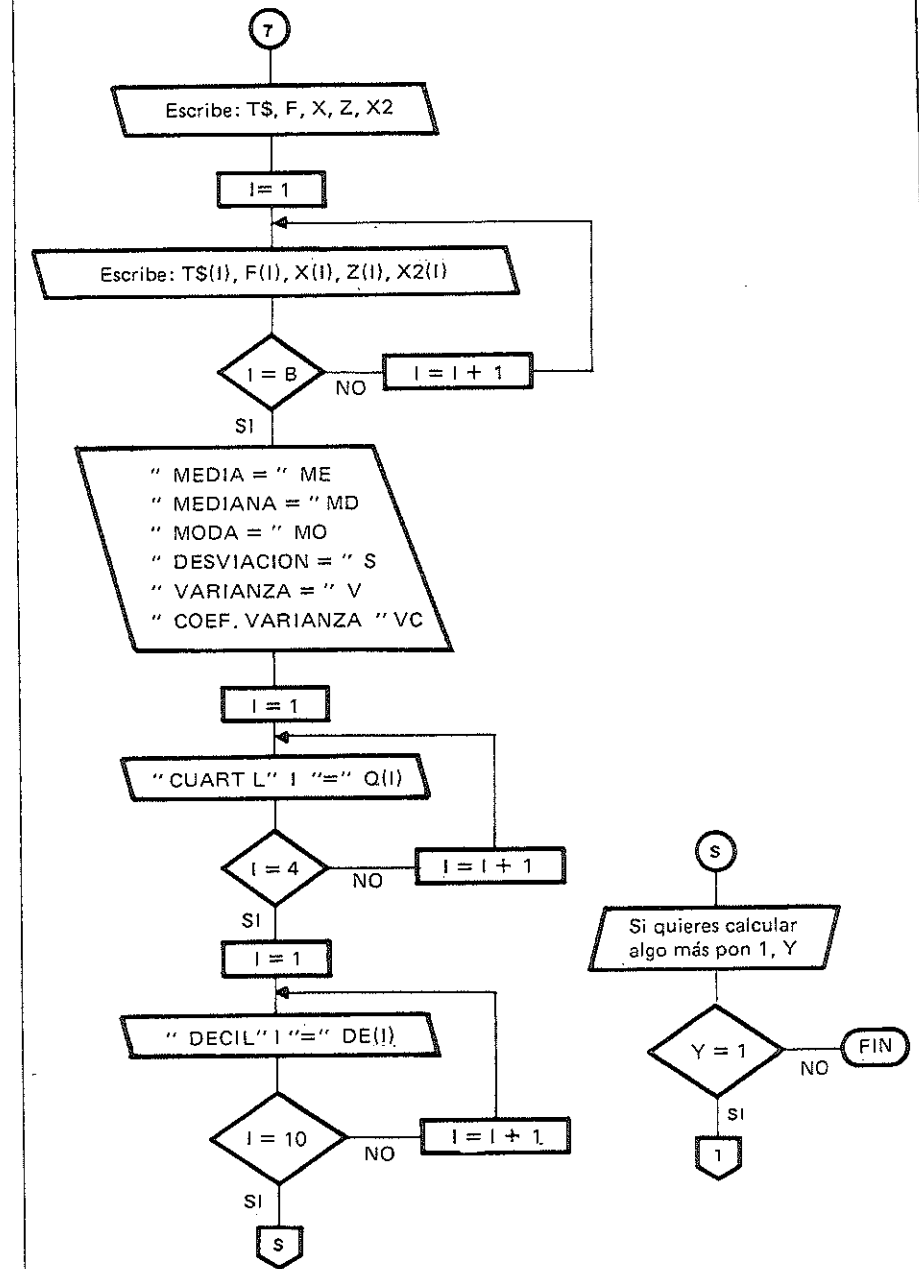
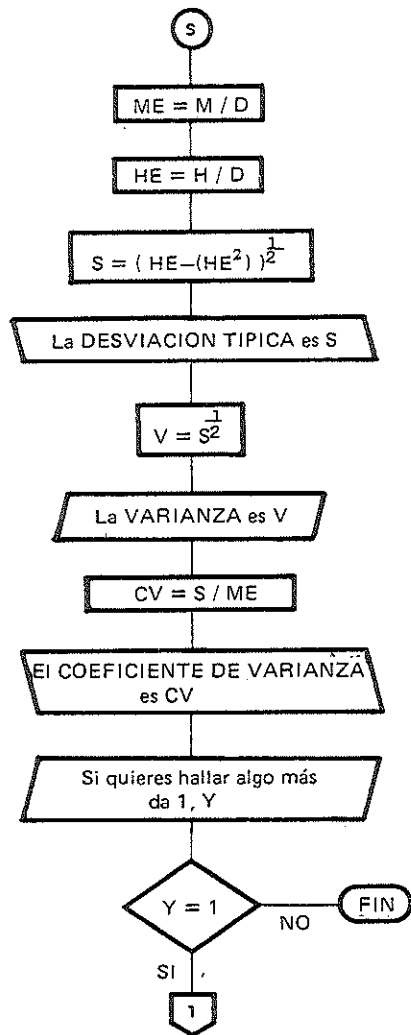






Desviación





5 ORGANIGRAMAS
ESTRUCTURADOS

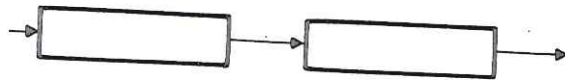
En este último tema vamos a realizar una serie de ejercicios utilizando otro tipo de representación, es decir, empleando organigramas estructurados.

Los Organigramas estructurados pretenden representar el programa de forma que sea fácilmente comprensible por el programador, de depurar, de un tamaño reducido y que se vea de forma modular, es decir, cada una de sus partes por separado.

Los Organigramas estructurados utilizan bloques iniciales básicos, que se representan mediante estructuras que sólo tienen un punto de entrada o comienzo y un punto de salida o final.

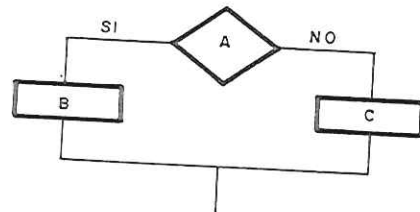
Utilizan tres bloques básicos o estructuras simples, cuya representación gráfica es:

SECUENCIAL



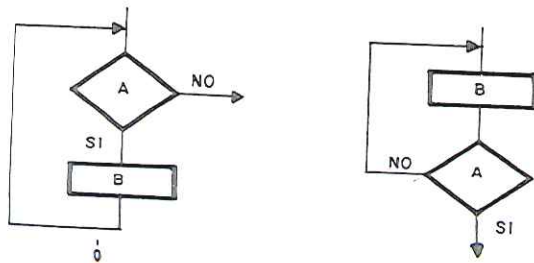
Representa la ejecución en serie de varias instrucciones.

SELECCION



Dependiendo de si se cumple una determinada condición, tomamos dos caminos diferentes.

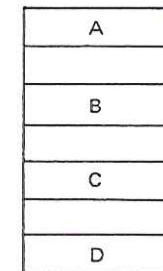
ITERACION



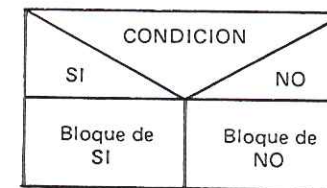
Con esta representación indicamos que queremos que se repitan una serie de instrucciones, un número determinado de veces o hasta que se cumpla una determinada condición.

Estas estructuras las podemos representar sin necesidad de utilizar flechas, pues la secuencia de ejecución va siempre desde arriba hacia abajo.

SECUENCIAL

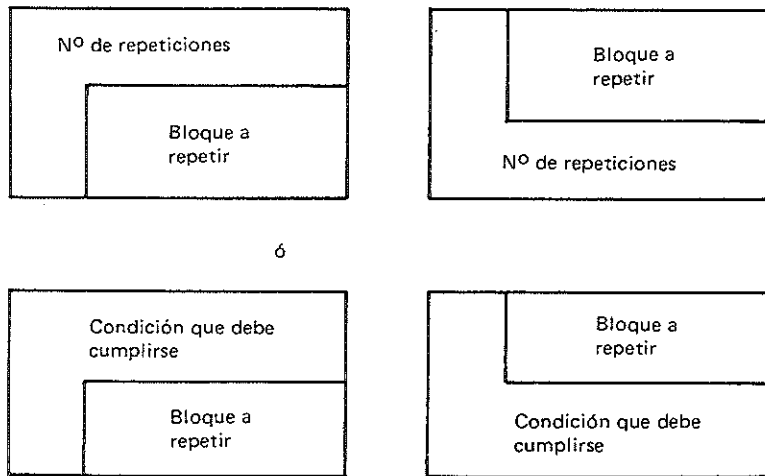


SELECCION



En esta representación, en el bloque de la izquierda representamos el grupo de sentencias a ejecutar si se cumple la condición, y en la parte de la derecha introducimos el grupo de sentencias que debemos ejecutar si no se cumple la ejecución.

ITERACION



6

En la parte superior de este esquema se indica el número de veces que queremos repetir un grupo de sentencias o bien la condición que debe cumplirse para detener este bucle. En la parte de bloque a repetir, se introducirán las sentencias que queremos ejecutar varias veces.

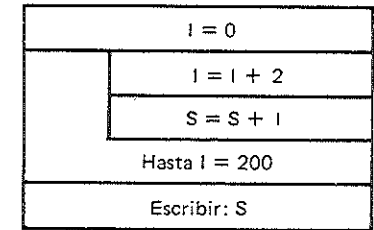
A continuación vamos a ver una serie de Organigramas estructurados. Algunos de los ejemplos expuestos están también resueltos por medio de organigramas tradicionales, con el fin de que se puedan comparar las dos estructuras.

77 Hallar la suma de los diez primeros números pares.

Lista de variables

I = Con esta variable vamos generando los números pares.

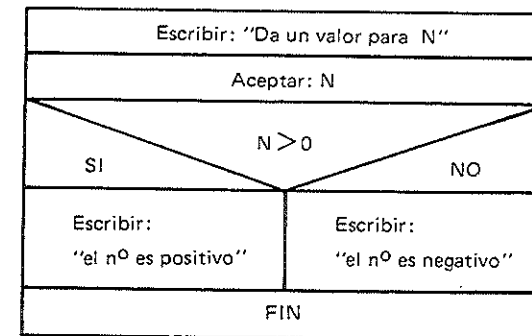
S = en esta variable realizamos la suma acumulada.



78 Dado un número, contestar si es positivo o negativo.

Lista de variables

N = Número a investigar.



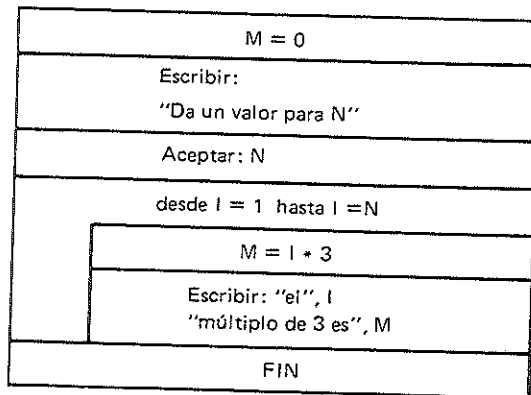
79 Hallar los N primeros múltiplos de 3.

Lista de variables

N = n.º total de múltiplos que queremos hallar.

M = en esta variable vamos almacenando el múltiplo hallado.

I = con esta variable contamos los múltiplos.



80 Dados varios números por pantalla, contestar cuál es par y cuál es impar, contando los pares y los impares. Preveer un final.

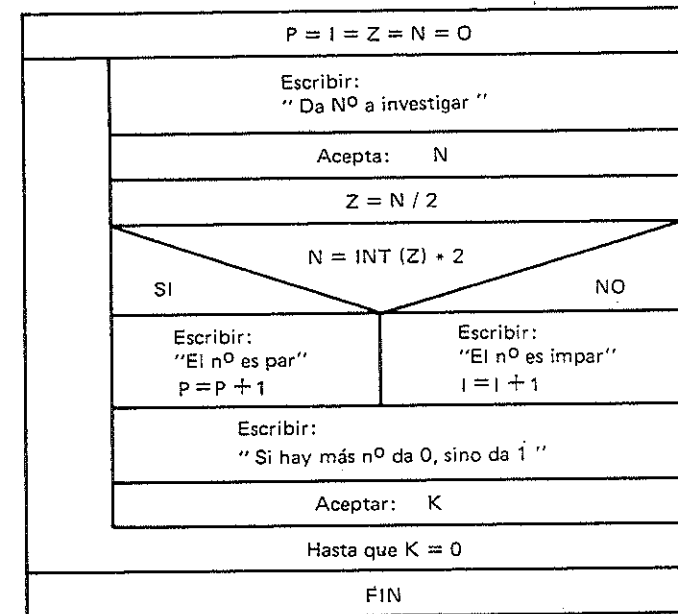
Lista de variables

N = n.º total de números a investigar.

Z = en esta variable vamos a guardar el resultado de dividir el n.º a investigar por dos, para ver si es par.

P = total de n.º pares.

I = total de n.º impares.



81) Hallar el factorial de N números.

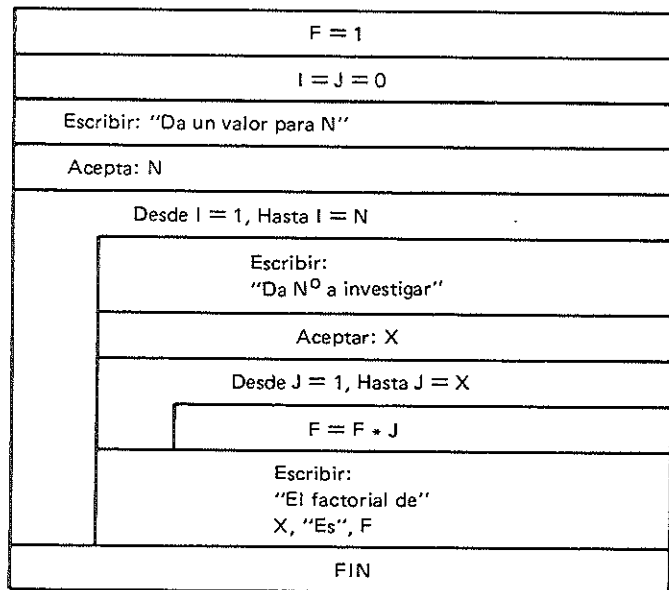
Lista de variables

I = cuenta los números a investigar.

J = genera el n.º por el cual queremos ir multiplicando.

N = n.º total de números que queremos hallar el factorial.

F = almacena el factorial de cada uno de los n.º.



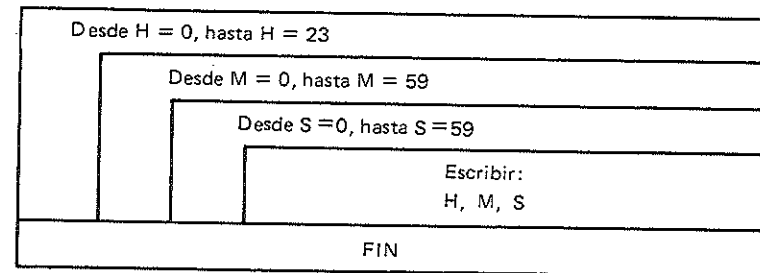
82) Hacer que el ordenador funcione como un reloj.

Lista de variables

H = con ella generamos las horas. Varía desde 1 hasta 23.

M = con ella generamos los minutos. Varía desde 1 hasta 59.

S = con ella generamos los segundos. Varía desde 1 hasta 59.

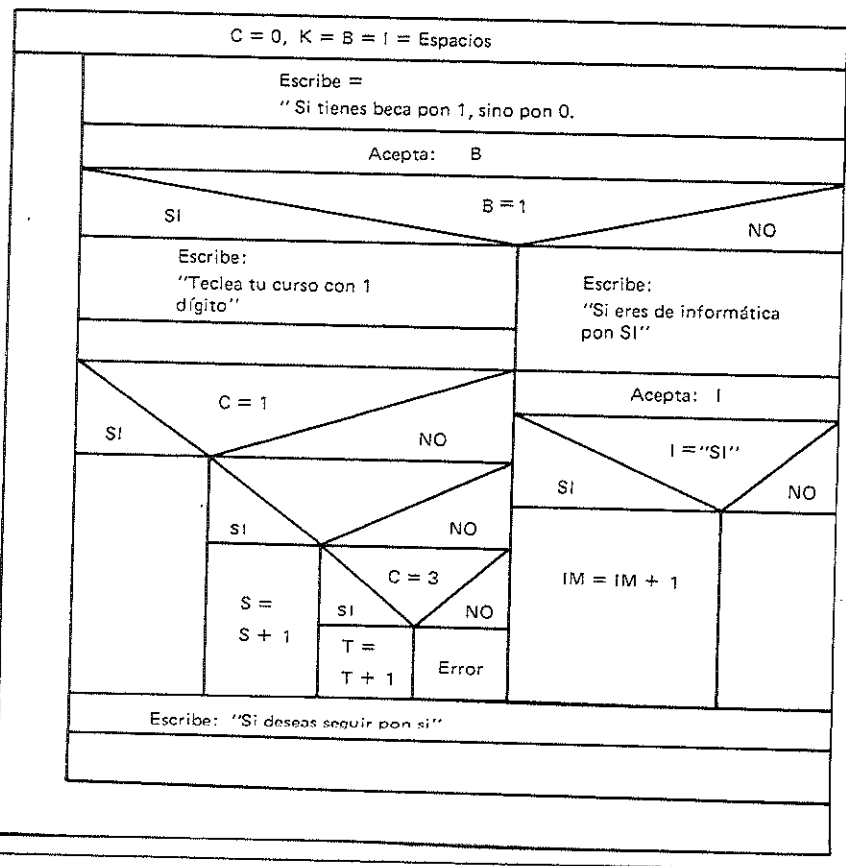


83) Vamos a suponer que tenemos un ordenador en la puerta de un Instituto de Formación Profesional, y que queremos hacer una encuesta entre los alumnos del centro.

Se desea saber cuántos alumnos tienen beca y cuántos no. Dentro de los que tienen beca, cuántos son de 1.º, cuántos de 2.º y cuántos de 3.º, y dentro de los que no la tienen, cuántos son de Informática.

Lista de variables

- P, S, T = sirven para contar, los alumnos de 1.º, 2.º y 3.º.
- B = en ella guardamos "SI", si el alumno tiene beca.
- C = en ella guardamos el curso.
- I = en ella guardamos "SI", si el alumno es de Informática.



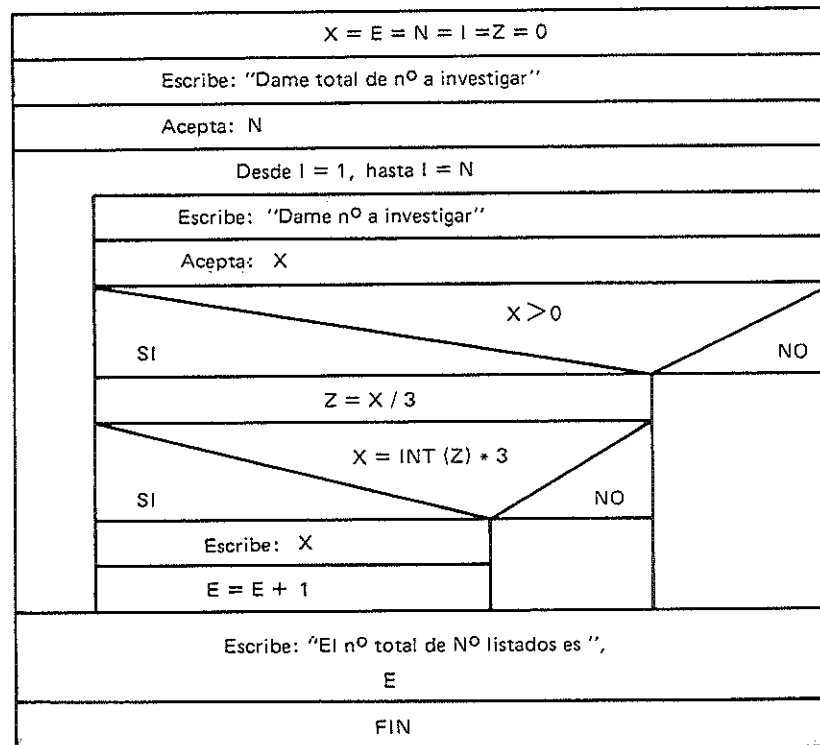
84) Dados N números enteros, obtener un listado de todos los positivos múltiplos de 3, dando al final del listado la cantidad de números que han sido listados.

Lista de variables

N = cantidad de n.º a investigar.

X = en esta variable vamos guardando los n.º a investigar.

Z = en esta variable vamos guardando el resultado de dividir el n.º por tres para ver si es múltiplo.



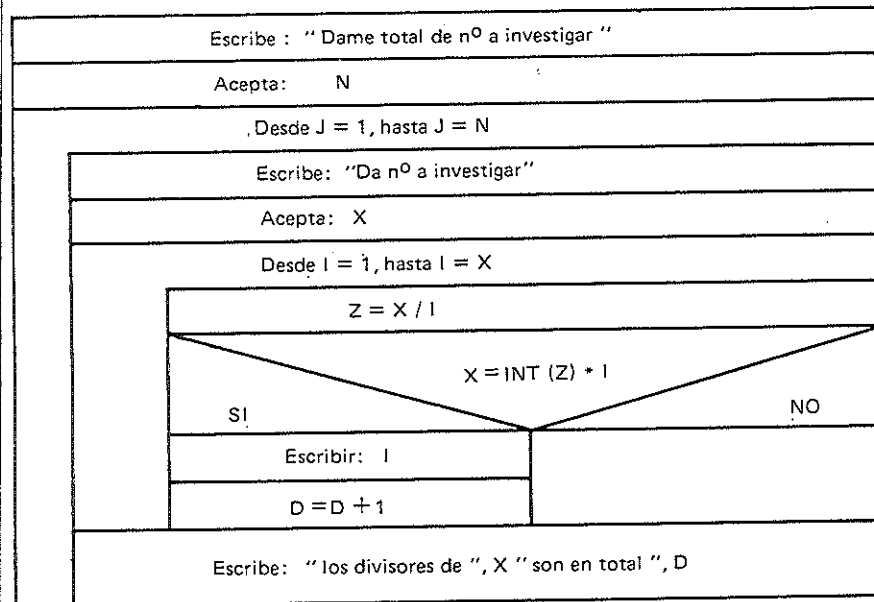
85) Calcular todos los divisores de N números y contarlos.

Lista de variables

N = en ella guardamos el total de n.º a investigar.

X = en ella guardamos cada uno de los n.º que tenemos que investigar.

Z = en ella vamos guardando el resultado de ir dividiendo, el n.º X, por todos los n.º desde 1 hasta X.



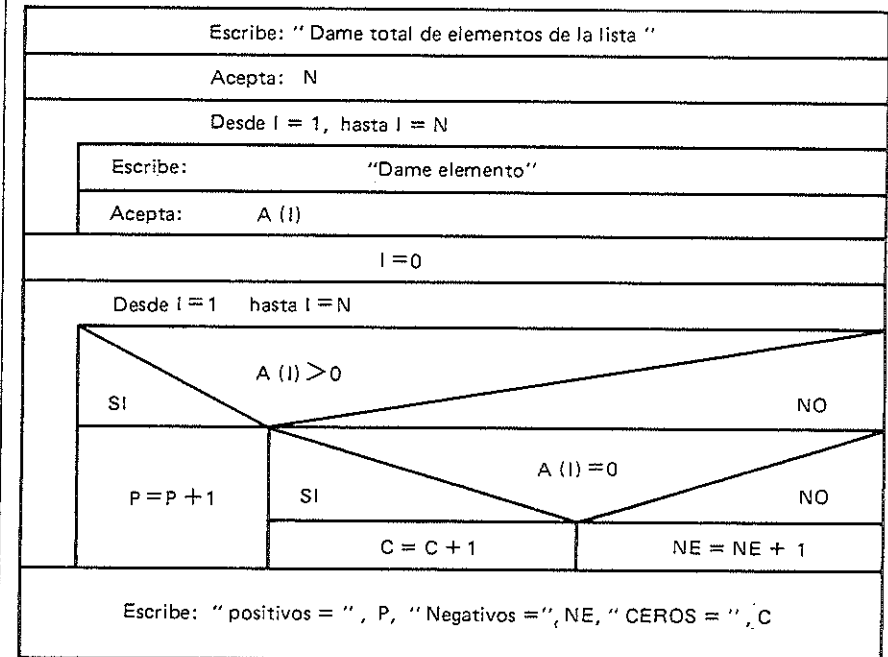
86) Crear una lista en memoria para N elementos y a continuación contar los positivos, los negativos y los ceros.

Lista de variables

N = en ella guardamos el total de elementos de la tabla.

I = la utilizamos como índice de la tabla.

P, C, NE = con ellas vamos contando los positivos, los ceros y los negativos.



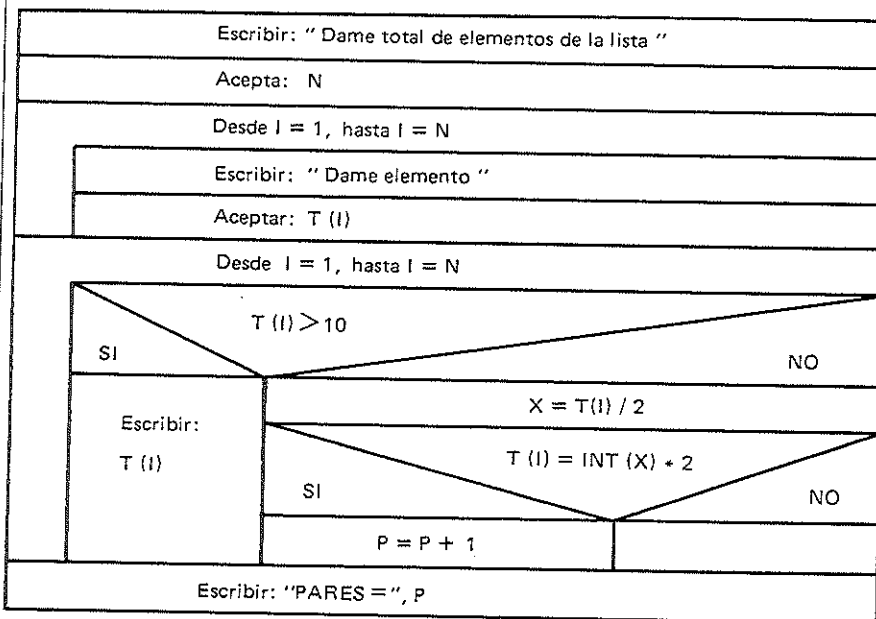
87) Crear una lista en memoria para N elementos, y a continuación contar los que sean pares e imprimir los mayores de 10.

Lista de variables

I = la utilizamos como índice de la tabla

P = con ella vamos contando los pares.

X = la utilizamos para ir guardando el resultado de dividir el n.º por 2 para ver si es par.



88) Crear una lista en memoria para 100 elementos. A continuación investigarla metiendo en distintas listas los pares y los impares y contando los ceros. Por último, escribir las dos listas obtenidas y el total de ceros.

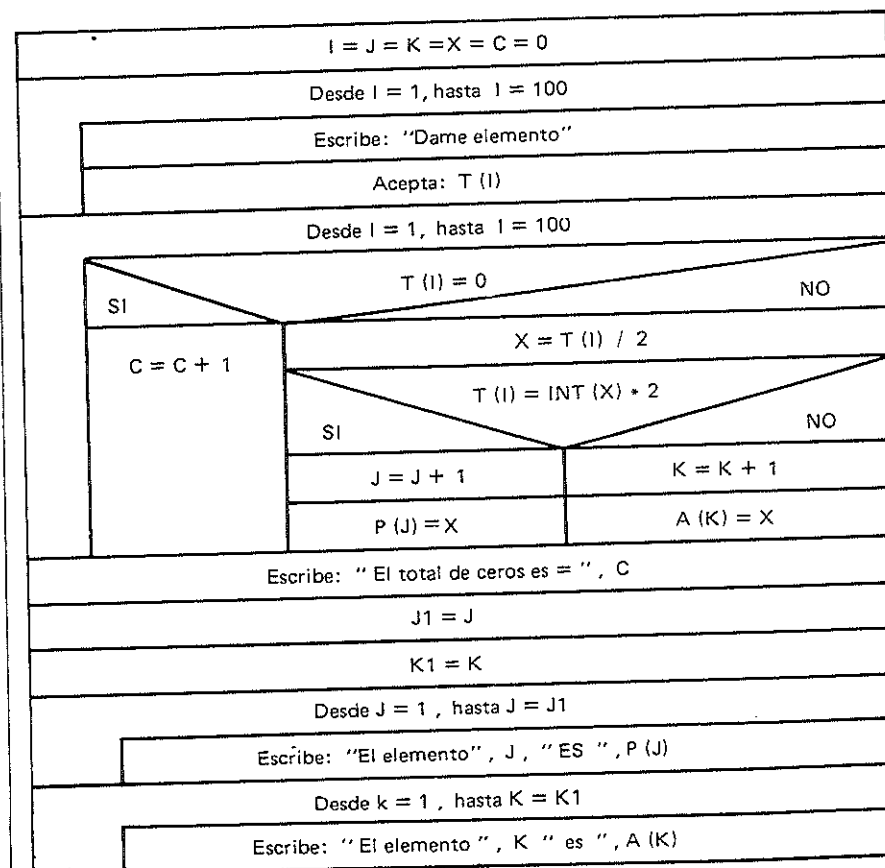
Lista de variables

T = en esta variable meteremos la lista inicial.

P = con este nombre llamaremos a la lista de los pares.

Ai = con este nombre llamaremos a la lista de los impares

C = con esta variable contaremos el total de ceros



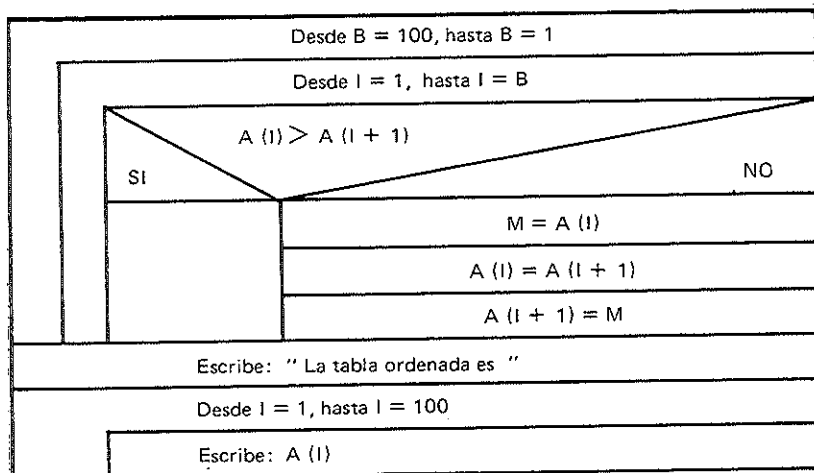
- 89 Dada una tabla en memoria de 100 elementos, ordenarla de mayor a menor y a continuación escribir la tabla ordenada.

Lista de variables

A = este es el nombre que le damos a la tabla primitiva.

I = esta variable la utilizamos como índice de la lista.

B = esta variable la vamos a utilizar como tope hasta donde llevamos ordenada la tabla.



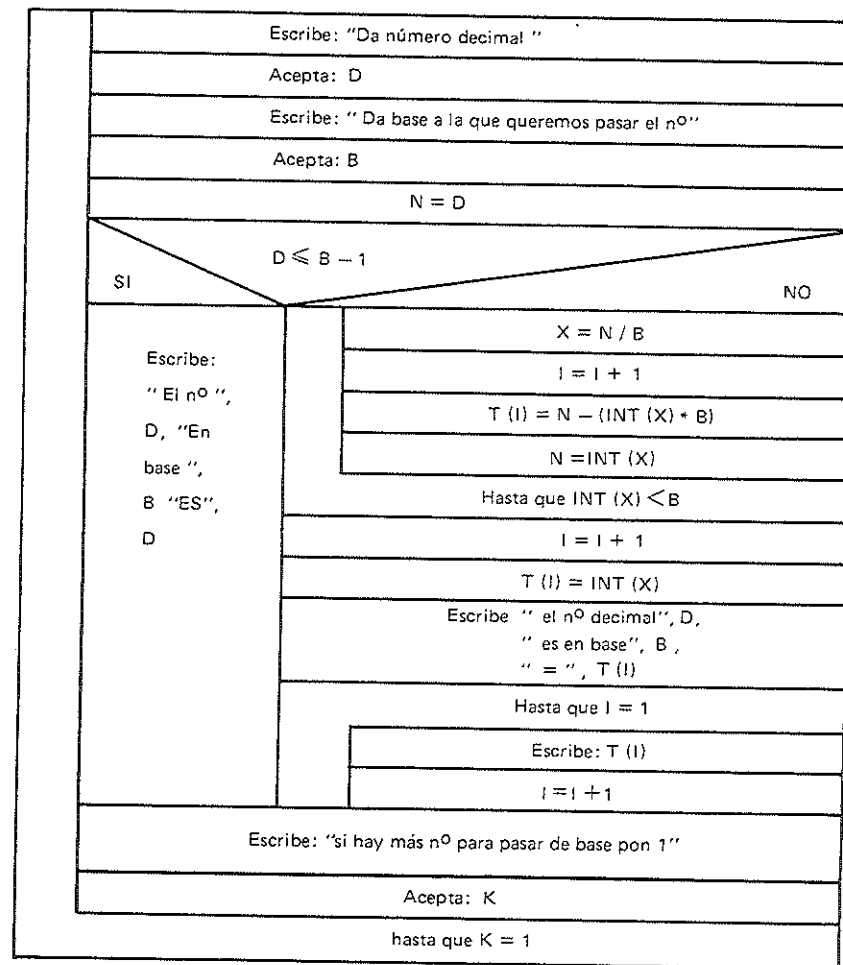
- 90 Hacer un programa que pase un número decimal a cualquier otra base. Preveer un final.

Lista de variables

D = en esta variable guardamos el n.º decimal.

B = en esta variable guardamos la base a la que queremos pasar el n.º decimal.

T = esta variable es una tabla en la cual vamos almacenando los resultados de dividir el n.º decimal por la base.

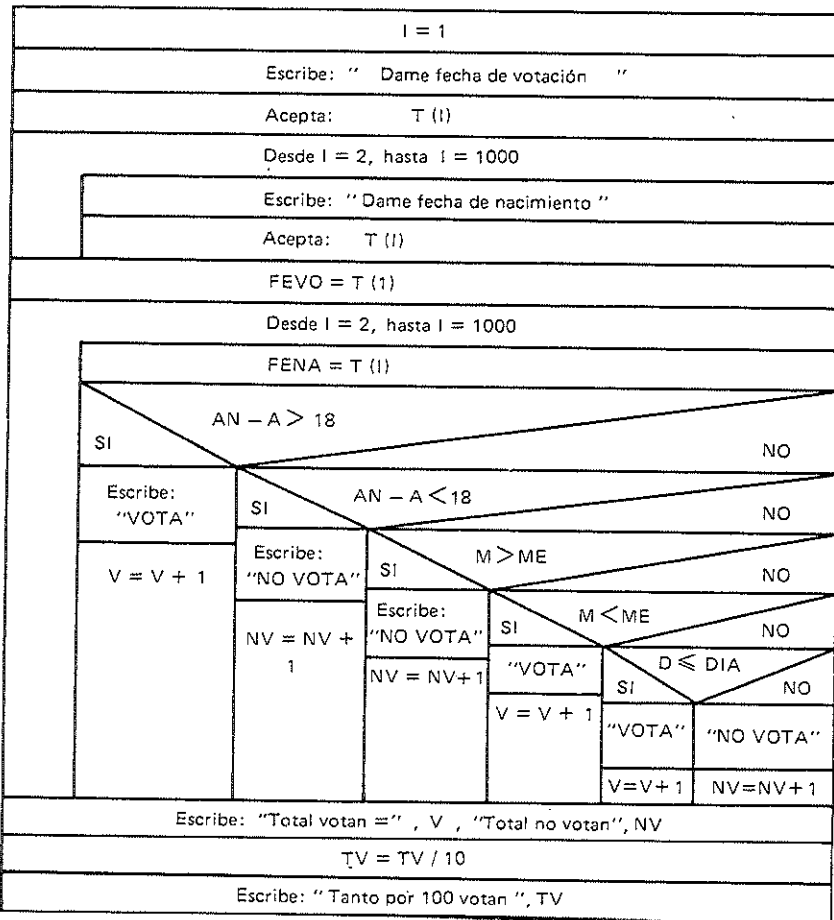


- 91 Crear una tabla en memoria para guardar 1001 elementos, de forma que en el primero metamos la fecha de una determinada votación y en los restantes metamos las fechas de nacimiento de los votantes. A continuación queremos investigar dicha tabla, para saber cuántos tienen edad suficiente para votar y cuántos no, así como el tanto por ciento.

Lista de variables

FEVO = esta variable se divide en tres subcampos A; M; D.

FENA = se divide en 3 subcampos AN, ME, DIA.

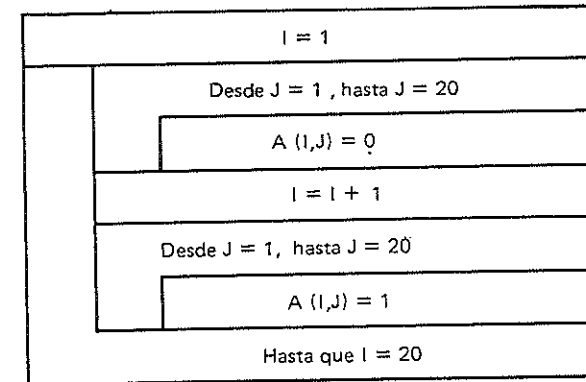


- 92 Generar una matriz de 20 filas y 20 columnas, de forma que las filas pares se rellenen a unos y las impares a ceros.

Lista de variables

I = esta variable la utilizo para indicar la fila en que me encuentro.

J = con esta variable indico la columna en que me encuentro.

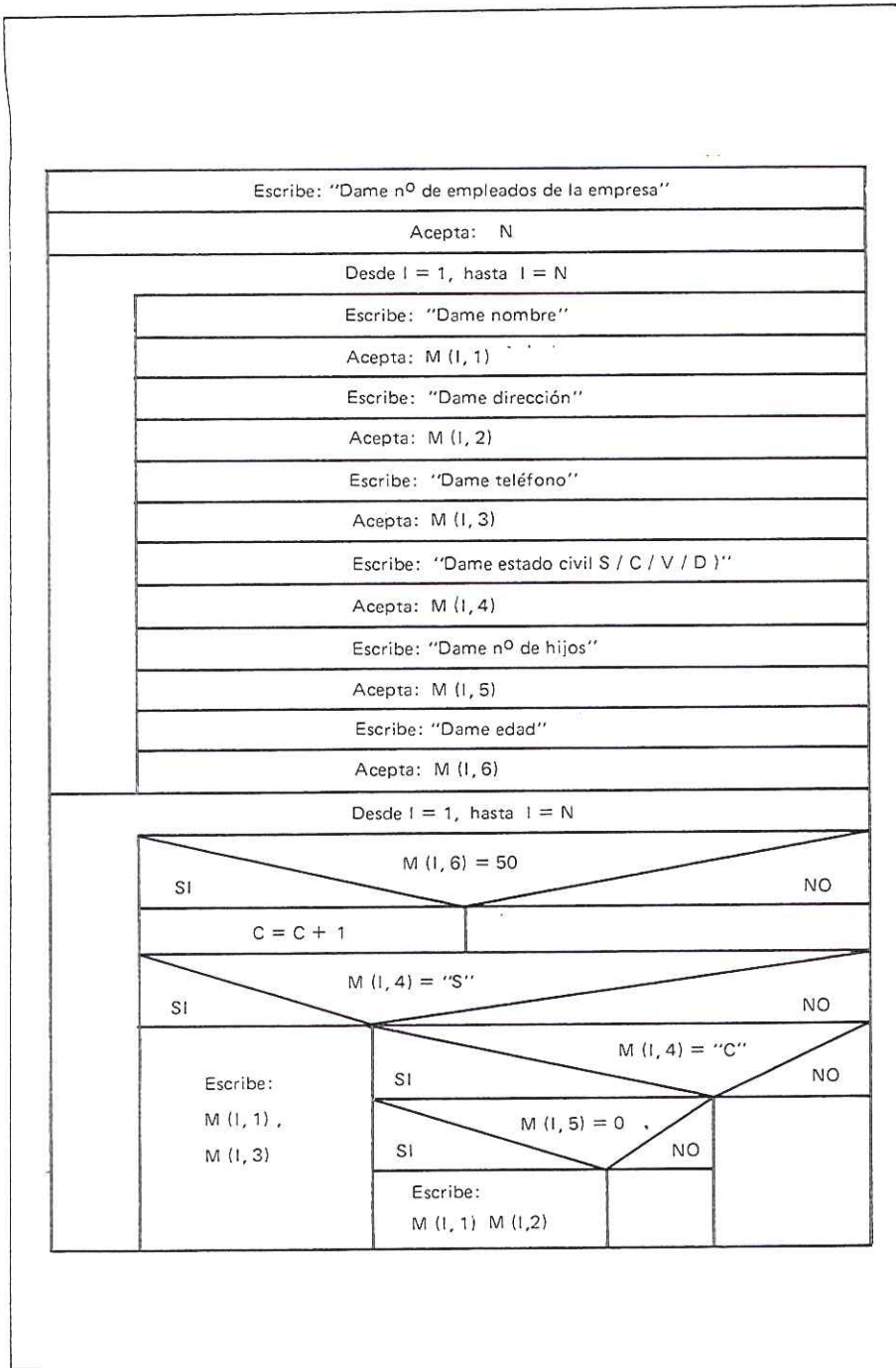


- 93 Crear una matriz con los datos de los empleados de una empresa. A continuación queremos investigarla contando todos aquellos empleados que tengan más de 50 años, y los empleados que sean solteros. Dentro de los empleados casados, queremos saber el nombre y la dirección de todos aquellos que no tengan hijos.

Lista de variables

M = será una matriz de tantas columnas como datos del empleado tengamos. Por ejemplo: nombre, dirección, teléfono, estado civil, n.º de hijos, edad, etc.

C = con ella vamos contando los empleados de más de 50 años.

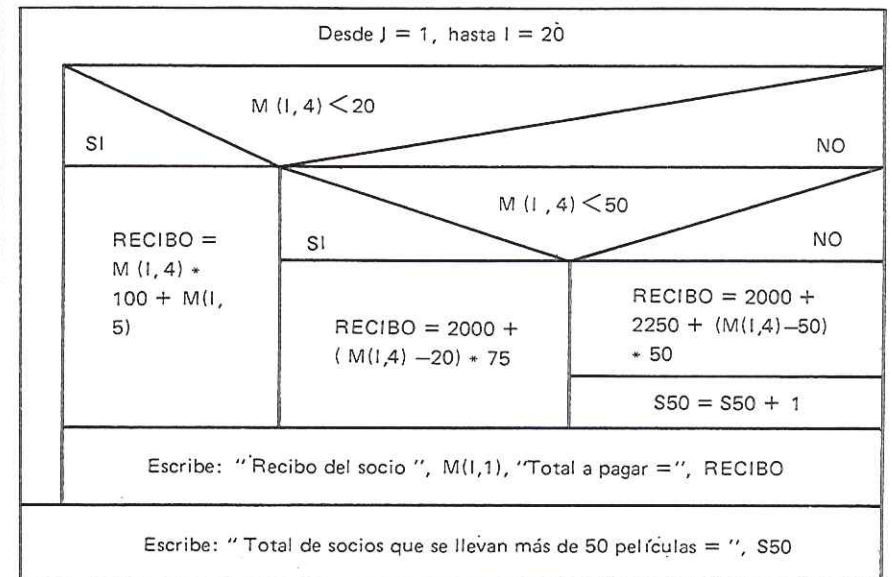


94 Tenemos una matriz en memoria interna con los datos de los abonados de un vídeo club: nombre, dirección, teléfono, n.º de películas llevadas, cuota mínima, etc.

Queremos realizar el recibo de los socios de forma que todos los socios deben pagar su cuota mínima y además por cada película que se hayan llevado en dicho mes deben pagar 100 ptas si son menos de 20 películas, 75 ptas por cada película que pase de 20 y no llegue a 50, y 50 ptas, por las que pasen de 50. Además, queremos contar todos los socios que se llevan más de 50 películas.

Lista de variables

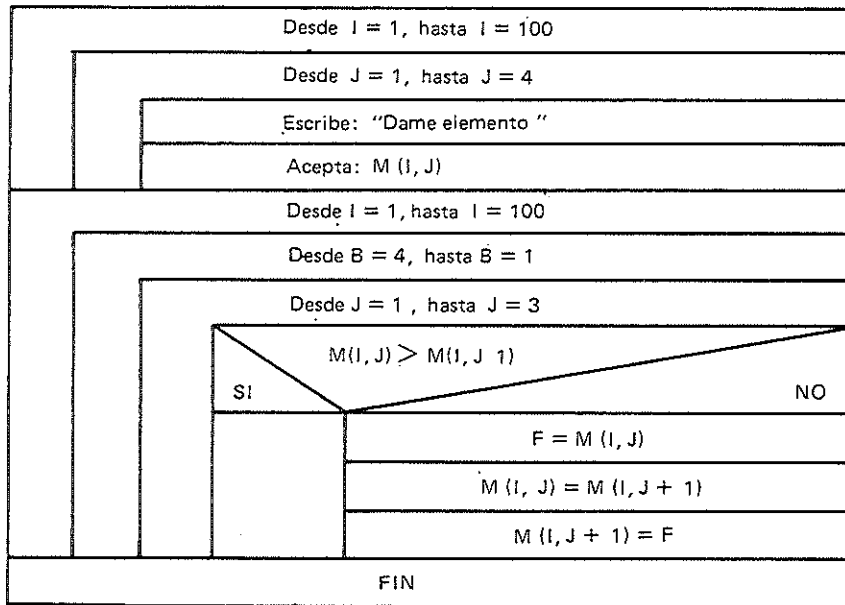
RECIBO = en ella guardamos el total a pagar por cada uno.
S50 = en ella contamos los socios que se llevan más de 50 películas.



- 95) Crear una matriz de 100 filas y 4 columnas, cada una de las cuales es numérica y de tres dígitos. Queremos ordenar cada una de sus filas de menor a mayor.

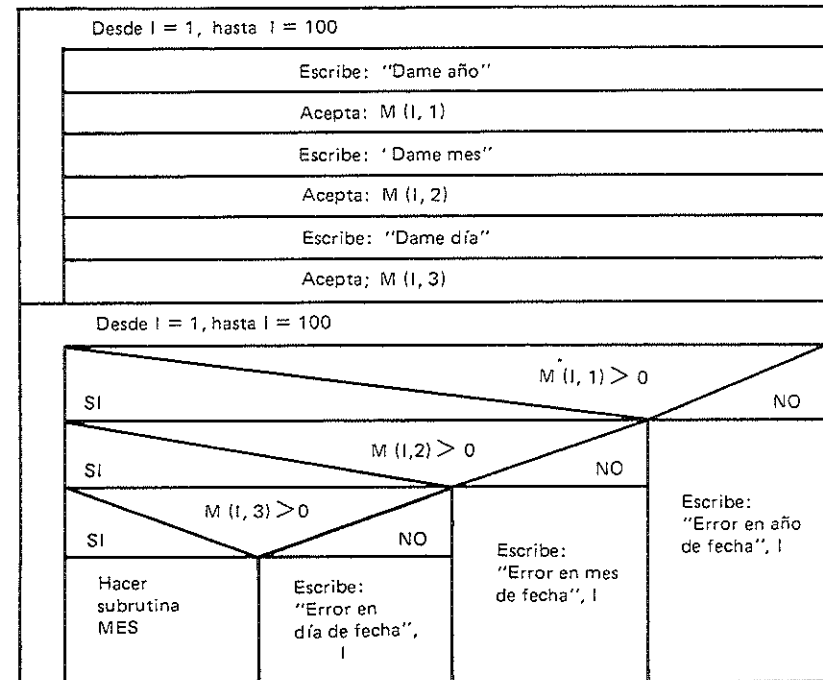
Lista de variables

B = Con ella vamos indicando por qué elemento de la matriz vamos ordenando.

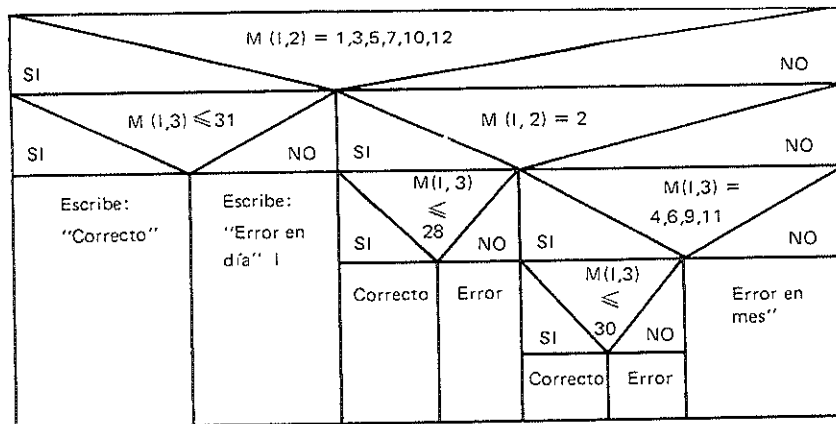


- 96) Crear una matriz de 100 filas, con 100 fechas y tres columnas, en cada una de las cuales vendrá: en la 1.^a columna el año, en la 2.^a el mes y en la 3.^a el día. A continuación queremos investigar comprobando que todas las fechas son correctas, es decir:

- Que ninguno de los tres números sea negativo.
- Que el día sea correcto según el mes
- Que el mes sea correcto

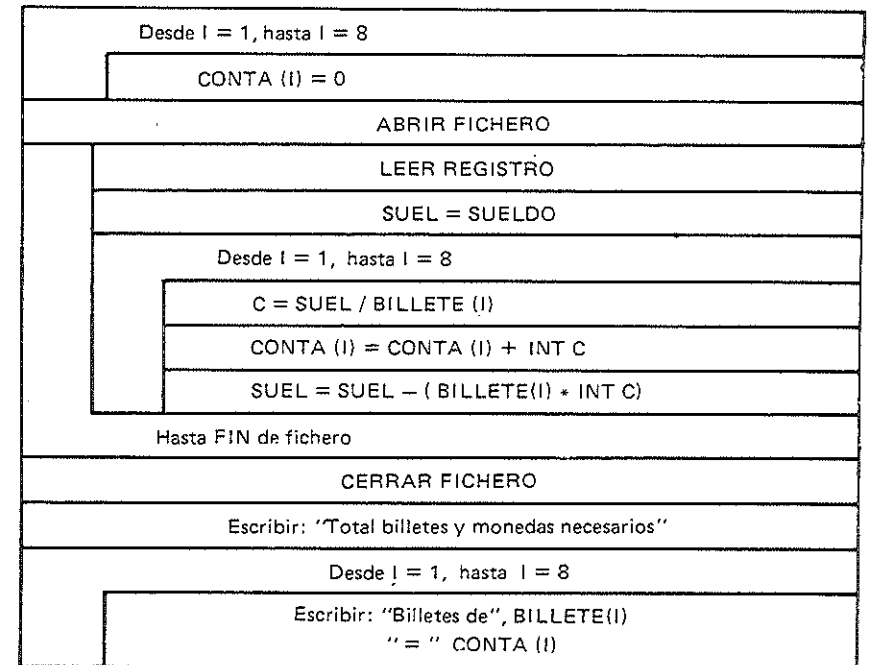


SUBROUTINA MES



97 Tenemos una tabla en memoria con los distintos tipos de billetes: 5000, 1000, 500, 100, 50, 25, 5, 1. Tenemos también otra tabla la cual vamos a utilizar como contador para los distintos tipos de billetes (CONTA). Por último tenemos un fichero de empleados con organización secuencial, en el cual existe un campo denominado sueldo, en el que viene el sueldo a percibir por cada empleado.

Queremos obtener el n.º de billetes de 5000, 1000, 500, 100, 50 etc., necesarios para pagar a los 100 empleados de la empresa.



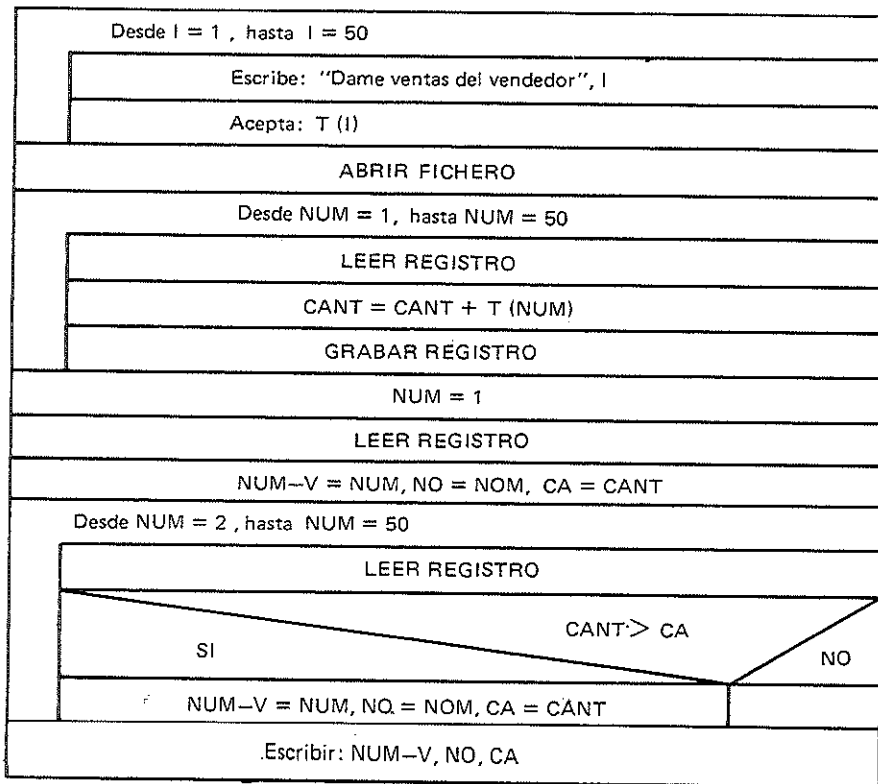
- 98) Tenemos un fichero con organización relativa, con los datos de los 50 vendedores de una empresa: NUM, NOM, CANT.

Queremos crear una tabla en la cual vamos metiendo las nuevas ventas de cada vendedor. El lugar que ocupa dentro de la tabla corresponde con el n.º de vendedor.

A continuación queremos actualizar el fichero de vendedores con la tabla, así como sacar el vendedor con las ventas más altas, sus ventas y su número.

Lista de variables

I = La utilizamos como índice de la tabla en su creación; después utilizamos como índice el campo NUM, que me indica el n.º de vendedor deseado.



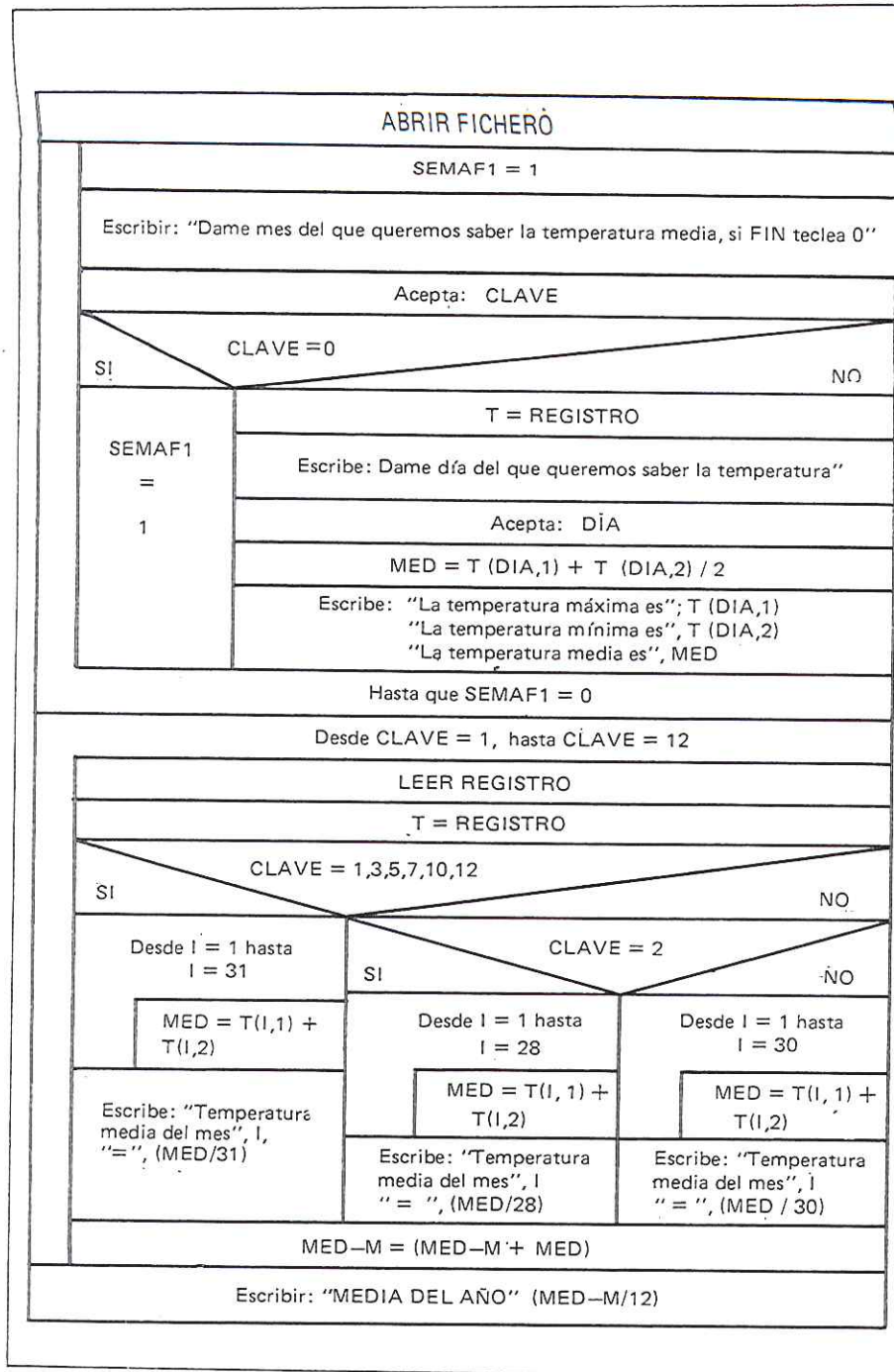
- 99) Tenemos un fichero con organización relativa, con las temperaturas habidas en una ciudad durante todo un año. Es decir, el fichero consta de 12 registros que corresponden a los 12 meses. Cada registro consta de 31 campos que corresponden a los días del mes, y a su vez cada campo se subdivide en dos subcampos en los cuales viene las temperaturas máxima y mínima de ese día.

1. Dar la posibilidad de, dado un mes y un día de dicho mes, que nos conteste con la temperatura máxima, la temperatura mínima y la temperatura media de dicho día.
2. Queremos saber la temperatura media de todos los meses del año.
3. Queremos saber la temperatura media del año.

Lista de variables

T = esta variable será una matriz en memoria central, de 31 filas y 2 columnas, que corresponderá con los días de un mes. En ella vamos a volcar el registro leído indicado por el mes, para podernos situar en el día que deseemos.

CLAVE = Esta variable será el campo por el cual accedemos al fichero, variará desde 1 hasta 12, pues corresponderá con los meses del año.



100 Dado un fichero con organización relativa, cuyo campo clave es N-ART, con los datos de los artículos de una empresa N-ART, FABRICANTE, PROVEEDOR, P.V.P., P-COSTO, EXISTENCIAS, ST-MINIMO.

Queremos:

1. Sacar por impresora, el fabricante y el proveedor de todos aquellos artículos cuyas existencias sean menores de 10.
2. Sacar por pantalla el P-Costo y el P.V.P. de todos aquellos artículos cuyo fabricante sea "GRANADA, S. A."
3. Dado un número de artículo, dar la posibilidad de cambiar cualquier dato de ese artículo.

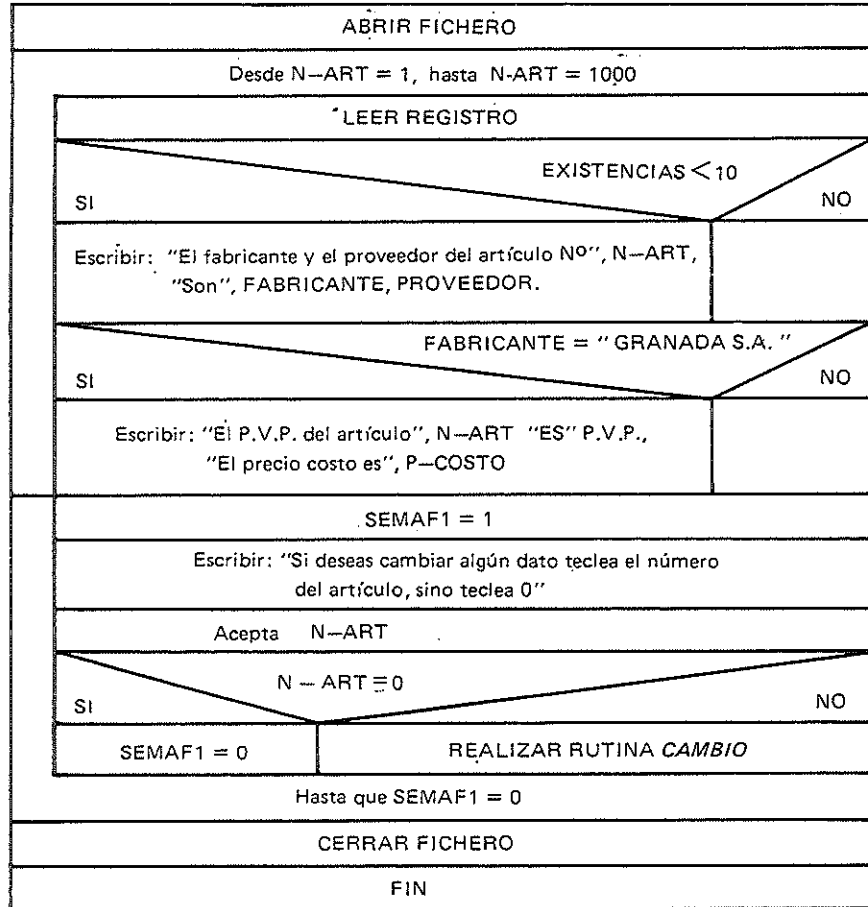
Lista de variables

SEMAF1 y SEMAF2 = con estas variables vamos a dar la posibilidad de tomar varios caminos posibles, según el valor que tengan.

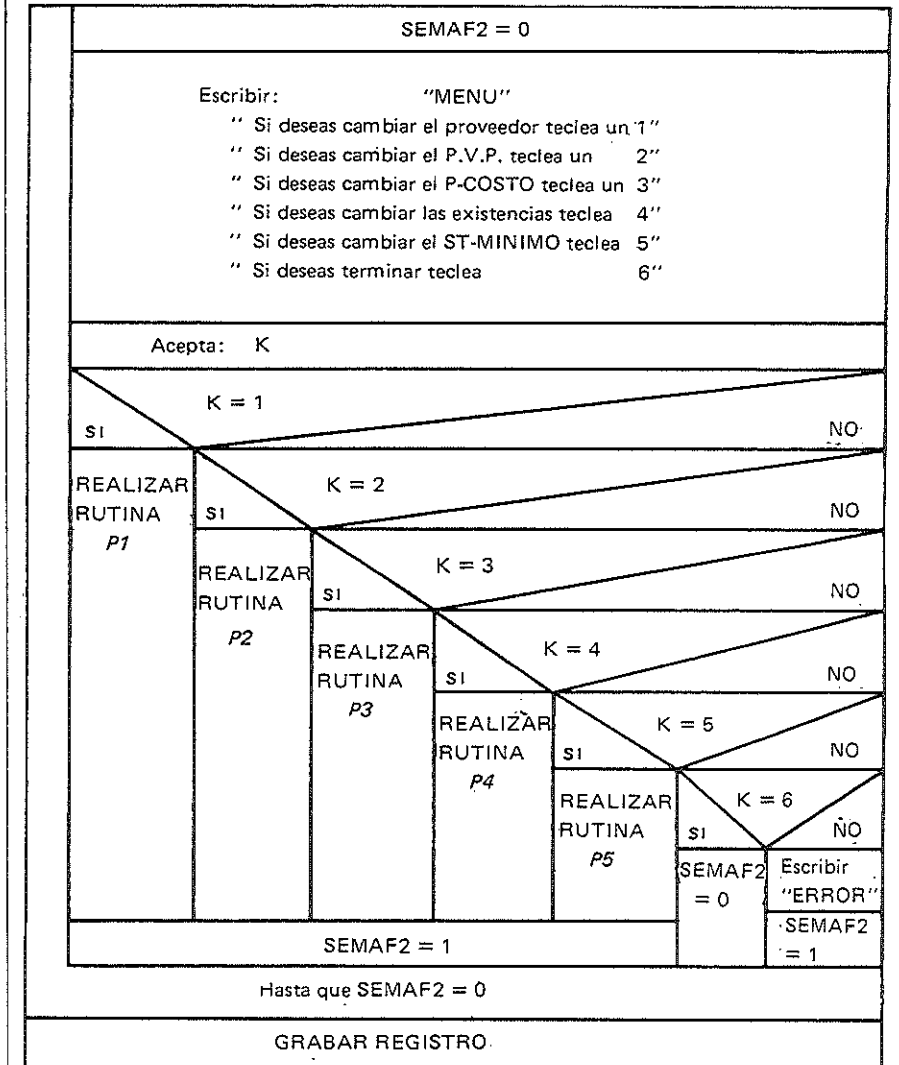
K = en esta variable vamos a introducir el valor escogido dentro de un menú.

UNID= Con esta variable vamos a modificar las existencias de los artículos.

PROGRAMA PRINCIPAL



RUTINA CAMBIO



Otros libros de INFORMATICA publicados por



RUTINA P1

Escribir: " Dame nuevo proveedor "

Acepta: PROVEEDOR

RUTINA P2

Escribir: " Dame nuevo P.V.P. "

Acepta: P.V.P.

RUTINA P3

Escribir: "Dame nuevo P-COSTO "

Acepta: P-COSTO

RUTINA P4

Escribir: " Cuantas unidades se llevan "

Acepta: UNID

EXISTENCIAS = EXISTENCIAS - UNID

RUTINA P5

Escribir: " Dame nuevo STOCK MINIMO "

Acepta: ST-MINIMO

ABRAMSON.— Teoría de la información y codificación. (6ª edición). 216 páginas. 82 ilustraciones.

ANGELL.— Gráficos con computador. 176 páginas. 58 ilustraciones.

ANGULO y NO.— Control de procesos industriales por computador. 176 páginas. 87 ilustraciones.

BELLIDO y SANCHEZ.— Basic para estudiantes. (2ª edición). 158 páginas. 20 figuras.

BELLIDO y SANCHEZ.— Basic para maestros. (3ª edición). 184 páginas. 20 figuras.

COHEN.— Estructura, lógica y diseño de programas. 340 páginas. 198 ilustraciones.

CROCUS.— Sistemas de explotación de computadores. 424 páginas. 89 ilustraciones.

CUÑAT.— Pascal para estudiantes. 304 páginas. 80 figuras.

CHECROUN.— Basic. Programación de microordenadores. (5ª edición). 112 páginas. 9 figuras.

DAX.— CP/M. Guía de utilización. 144 páginas. 13 figuras.

DAX.— Lenguaje C. 204 páginas.

DELANNOY.— Ficheros en Basic. (4ª edición). 158 páginas.

FLORES.— Estructuración y procesos de datos. 478 páginas. 302 ilustraciones.

GALAN PASCUAL.— Pascal y turbopascal. (en prensa).

GALAN PASCUAL.— Programación con el lenguaje Cobol. (6ª edición). 328 páginas. 85 ilustraciones.

GARCIA MERAYO.— Programación en Fortran 77. 368 páginas. 45 figuras.

GAUTHIER.— Diseño de programas para sistemas. (5ª edición). 290 páginas. 122 ilustraciones.

GRIGORIEFF.— dBASE II - dBASE III. Guía de uso. 192 páginas.

GRIGORIEFF.— Informática para actividades profesionales. 134 páginas.

HARTMAN.— Manual de los sistemas de información. T.1. (7ª edición). 348 páginas. 58 figuras.

HARTMAN.— Manual de los sistemas de información. T.2. (7ª edición). 384 páginas. 88 figuras.

LANE.— Telemática y comunicaciones en la empresa. 128 páginas. 27 figuras.

LARRECHE.— Basic. Introducción a la programación. (5ª edición). 132 páginas.

LEWIS.— Estructuras de datos. Programación y aplicaciones. 404 páginas. 149 ilustraciones.

LUCAS.— Sistema operativo UNIX. 212 páginas. 31 figuras.

LUCAS.— Sistemas de información. Análisis. Diseño. Puesta a punto. 360 páginas. 59 ilustraciones.

Mc NITT.— Simulación con ordenador. 350 páginas.

PAZOS.— Inteligencia Artificial. Programación heurística. 496 páginas. 149 ilustraciones.

POLITIS y VANRYB.— Sistemas operativos PC-DOS MS-DOS para IBM/PC y compatibles. 220 páginas.

POPKIN.— Introducción al proceso de datos. 552 páginas. 214 figuras.

QUANEUX.— Tratamiento de textos con Basic. 160 páginas. 8 figuras.

QUEINNEC.— Programación en Lisp. 202 páginas.

RODRIGUEZ PRIETO.— Protección de la información. Diseño de criptosistemas informáticos. 256 páginas. 59 figuras.

ROSSI.— Basic. Curso acelerado. (5ª edición). 224 páginas. 35 figuras.

SANCHIS y GALAN.— Compiladores. Teoría y construcción. 624 páginas. 246 figuras.

SANCHIS y MORALES.— Programación con el lenguaje Pascal. (7ª edición). 368 páginas. 72 figuras.

SARAM, DE.— Programación en micro-prolog. Un lenguaje de la 5ª generación. 184 páginas. 27 ilustraciones.

SCHMIDT.— Introducción a los ordenadores y al proceso de datos. (5ª edición). 452 páginas. 105 figuras.

THOMAS y DOUGLAS.— Auditoría informática. 214 páginas. 16 figuras.

YOUNG.— Lenguajes en tiempo real.