

3-3 Algunos ejemplos de gráficos

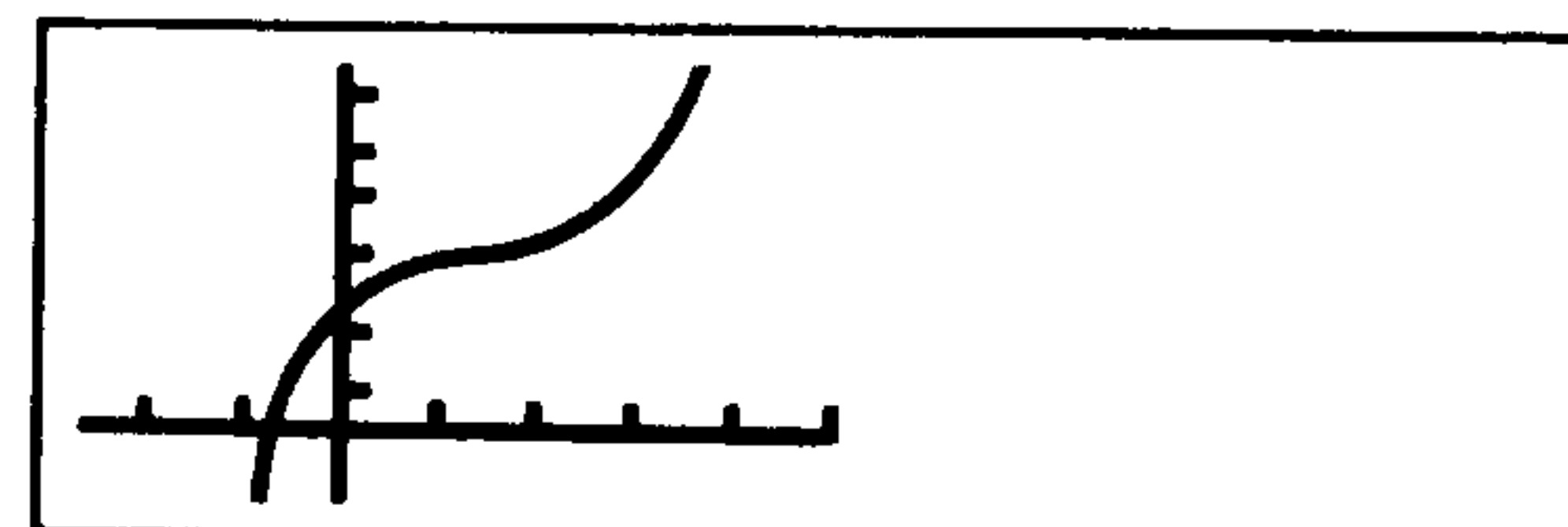
Se dan los ejemplos siguientes para mostrar algunas de las maneras en que las funciones de graficación pueden ser usadas efectivamente.

Ejemplo 1 Para graficar la función $y = x^3 - 9x^2 + 27x + 50$

Utilice los parámetros de gama siguientes.

Xmin : -5 Ymin : -30
Xmax : 10 Ymax : 150
Xscl : 2 Yscl : 20

SHIFT CIs EXE
Graph ALPHA X x^3 3 -
9 ALPHA X x^2 +
2 7 ALPHA X + 5 0 EXE

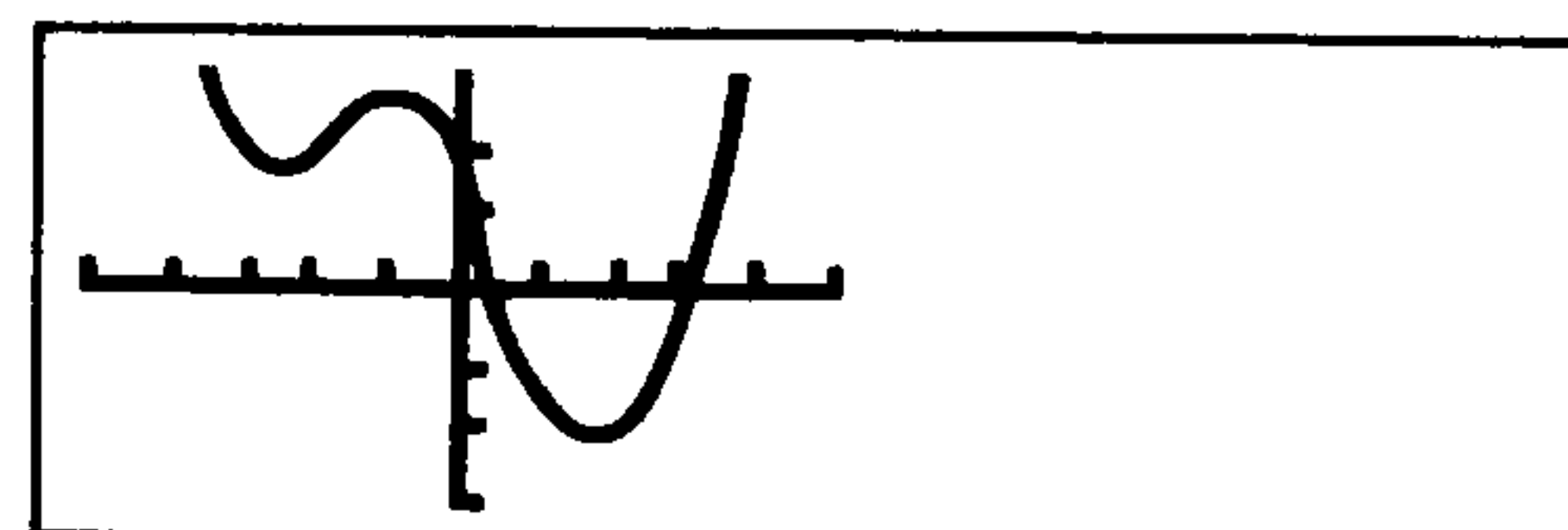


Ejemplo 2 Para graficar la función $y = x^4 + 4x^3 - 36x^2 - 160x + 300$ y determinar sus máximos y mínimos.

Utilice los parámetros de gama siguientes.

Xmin : -10 Ymin : -600
Xmax : 10 Ymax : 600
Xscl : 2 Yscl : 200

SHIFT CIs EXE
Graph ALPHA X x^4 4 +
4 ALPHA X x^3 3 -
3 6 ALPHA X x^2 - 1 6 0
ALPHA X + 3 0 0 EXE



3-4 Gráficos de estadísticas con una variable

- Los gráficos de estadísticas con una variable se trazan en el modo SD2 (SHIFT MODE X). "SD2" aparece en la presentación.
- Los gráficos de estadísticas con una sola variable pueden ser de barras o curvas de distribución normal.
- En el modo SD2 pueden trazarse, además, gráficos de funciones, de tal modo que puedan sobreponerse gráficos con valores teóricos y gráficos con valores reales.
- *Abs y $\sqrt[3]{}$ no pueden usarse en el modo SD2.
- El número máximo de datos es igual al número de memorias. Las memorias pueden ampliarse a un máximo de 19. Si se especifica un número mayor que 19, la unidad automáticamente ajusta el número de memorias a 19.
- En los gráficos trazados, la coordenada x corresponde al alcance de los datos, mientras que la coordenada y al número (frecuencia) de cada dato.
- La tecla DT se utiliza para la entrada de los datos.
- La tecla CL se utiliza para corregir los datos.

■ Trazado de gráficos para estadísticas con una sola variable

• Procedimiento

- ① Especificación del modo SD2 (SHIFT MODE X).
- ② Establecimiento de los límites (Range).
- ③ Ampliación de las memorias en base al número de barras (MODE n EXE).
- ④ Borrado de las memorias para cálculos estadísticos (SHIFT Scl EXE).
- ⑤ Entrada de los datos (dato DT).
- ⑥ Trazado del gráfico.

- Gráfico de barras Graph EXE
- Curva de distribución normal Graph SHIFT Line 1 EXE

*El método para la entrada de los datos (paso 5) es el mismo que se utiliza para los cálculos de la desviación estándar (ver la página 67).

Ejemplo Trazado de un gráfico ordinal (por rangos) en base a los siguientes datos.

Número de rango	1	2	3	4	5	6	7	8	9	10	11
Rango	0	10	20	30	40	50	60	70	80	90	100
Frecuencia	1	3	2	2	3	5	6	8	15	9	2

Realice los preparativos para el gráfico por medio del siguiente procedimiento:

- ① Especificación del modo SD2 (SHIFT MODE X).
- ② Establecimiento de los límites. Si bien el valor máximo que ha de trazarse sobre el eje x es igual a 100, el valor máximo X_{\max} se establece en 110 (por regla general, el valor mínimo debe ser igual o mayor que el límite inferior y el valor máximo menor que el límite superior; es por ello que aquí el eje x tiene una escala de 0 hasta 110).

Como la frecuencia máxima es igual a 15, se especifica 20 para Y_{\max} del eje y .

$X_{\min} : 0$ $Y_{\min} : 0$
 $X_{\max} : 110$ $Y_{\max} : 20$
 $X_{\text{scl}} : 10$ $Y_{\text{scl}} : 2$

- ③ Como el número de barras es igual a 11, (0 ~ 9, 10 ~ 19, 20 ~ 29 100 ~ 109), agregue 11 memorias.

MODE 1 1 EXE

SD2
M-37 S-312

- ④ Borrado de la memoria para cálculos estadísticos.

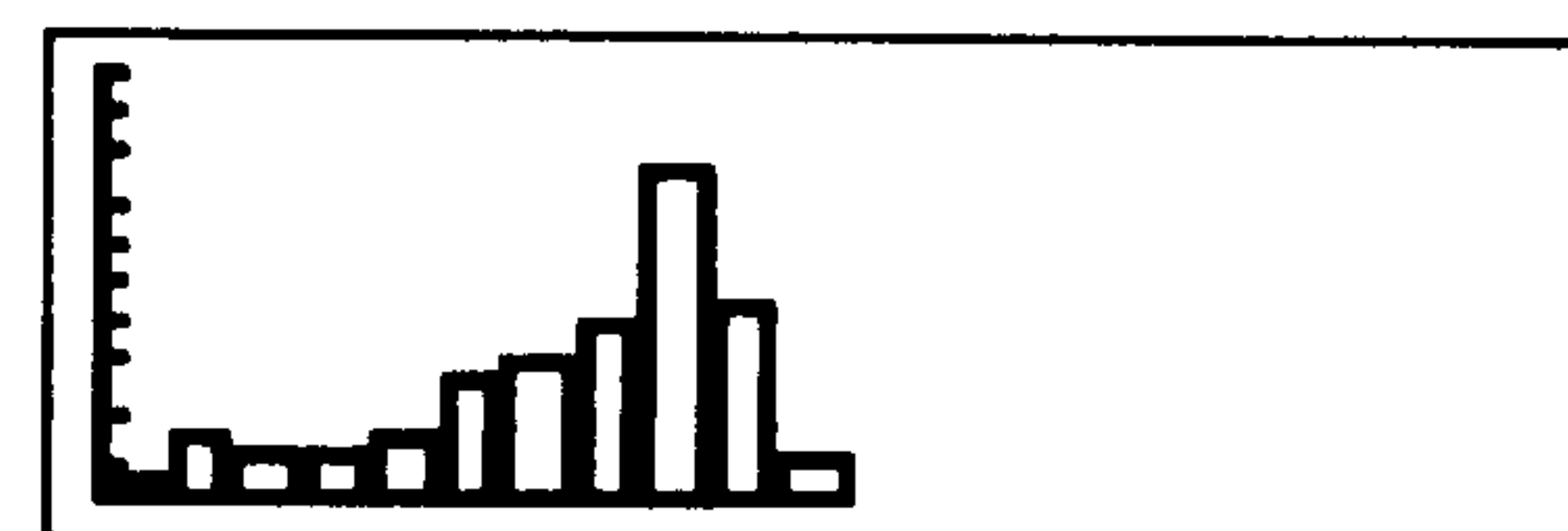
SHIFT SCI EXE

- ⑤ Entrada de los datos.

0 DT 10 DT DT DT 20 DT DT 30 DT DT 40 DT DT DT
 50 SHIFT ; 5 DT 60 SHIFT ; 6 DT 70 SHIFT ; 8 DT
 80 SHIFT ; 15 DT 90 SHIFT ; 9 DT 100 DT DT

- ⑥ Primero, se traza un gráfico de barras.

Graph EXE

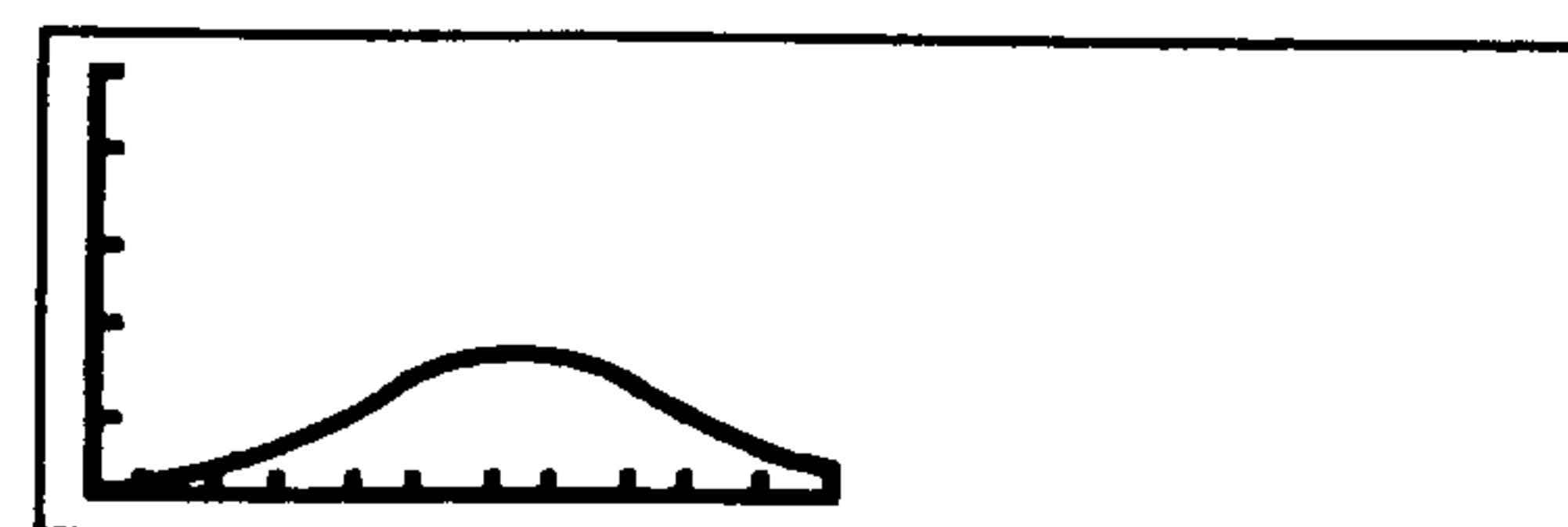


- Luego, se traza una curva de distribución normal. Como el eje y es relativamente pequeño comparado con un gráfico de barras, no pueden usarse los mismos límites. Cambie, por lo tanto, los mismos a los valores a continuación.

$X_{\min} : 0$ $Y_{\min} : 0$
 $X_{\max} : 110$ $Y_{\max} : 0.05$
 $X_{\text{scl}} : 10$ $Y_{\text{scl}} : 0.01$

Graph SHIFT Line 1 EXE

La entrada del número 1 hace que se trace una curva de distribución normal.



- Asegúrese de agregar la cantidad de memorias que sean necesarias (igual al número de barras). Cuando la cantidad de memorias no es suficiente, aparece un error Mem.
- El cambio del número de memorias durante la entrada de los datos hace que se modifique también el número de divisiones de datos y que el gráfico trazado no sea el apropiado.
- En caso de entrarse un valor que exceda los límites establecidos, el mismo sólo se asigna a la memoria para cálculos estadísticos, pero no a la memoria para gráficos.
- En caso de entrarse más datos que el límite establecido para el eje y , el gráfico de barras se traza hasta el límite superior de la pantalla (los puntos que quedan fuera del límite no pueden unirse).
- La fórmula utilizada para las curvas de distribución normal es:

$$y = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

*La designación del teclado de σ es $x\sigma n$. m es \bar{x} .

- Para los límites inferior y superior, X_{\min} debe ser menor que X_{\max} .
- Después de trazar un gráfico de barras, en la pantalla aparece la indicación "done" (terminado).

3-5 Gráficos de estadísticas con dos variables

- Los gráficos con pares de variables se trazan en el modo LR2 (SHIFT MODE LR2). "LR2" aparece en la presentación.
- Los gráficos de pares de variables pueden trazarse como líneas de regresión.
- En el modo LR2 se pueden trazar, además, gráficos de funciones comunes, lo cual permite sobreponer gráficos con datos teóricos, de distribución de datos y de líneas de regresión.
- Después de entrados los datos en el modo LR2, los puntos se visualizan inmediatamente en la pantalla y los datos se almacenan en la memoria para cálculos estadísticos.
- En caso de entrarse un valor que exceda los límites, el mismo se almacena en la memoria para cálculos estadísticos, pero no se visualiza en la pantalla.
- Los datos se entran usando la tecla DT conforme al siguiente formato: dato x SHIFT , dato y SHIFT ; frecuencia DT .
- La tecla CL se utiliza para corregir los datos después de entrados (en este caso, los puntos ya visualizados en la pantalla no se borran aun si el dato que le corresponde se corrige por medio de la tecla CL).
- Una vez borrada la pantalla (SHIFT CIS EXE), los puntos previamente visualizados no pueden recuperarse.

■ Trazado de gráficos para estadísticas con pares de variables

• Procedimiento

- ① Especificación del modo LR2 (SHIFT MODE LR2).
- ② Establecimiento de los límites (Range).
- ③ Borrado de la memoria para cálculos estadísticos (SHIFT SCI EXE).
- ④ Entrada de los datos (dato x SHIFT , dato y SHIFT ; frecuencia DT).
- ⑤ Trazado del gráfico (Graph SHIFT Line 1 EXE).

*El método para la entrada de los datos (paso 4) es el mismo que se utiliza para los cálculos de regresión (página 69).

Ejemplo Obtención de la regresión lineal y trazado de la línea de regresión para los datos siguientes.

x_i	y_i
-9	-2
-5	-1
-3	2
1	3
4	5
7	8

- ① Especificación del modo LR2 (SHIFT MODE LR2).
- ② Establecimiento de los límites que se observan en la tabla.

Xmin : -10 Ymin : -5
 Xmax : 10 Ymax : 15
 Xscl : 2 Yscl : 5

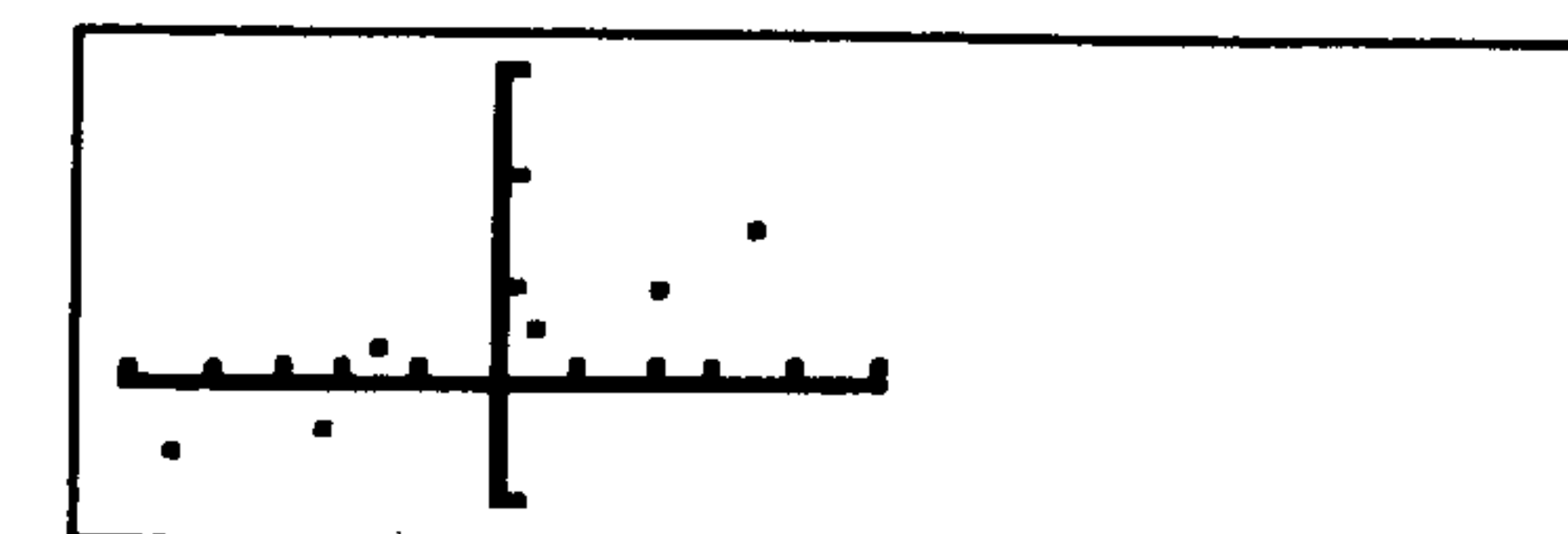
*Por regla general para los valores límites del eje x , los valores de x son:
 $-10 \leq x < 10$.

- ③ Borrado de las memorias para cálculos estadísticos.

SHIFT SCI EXE

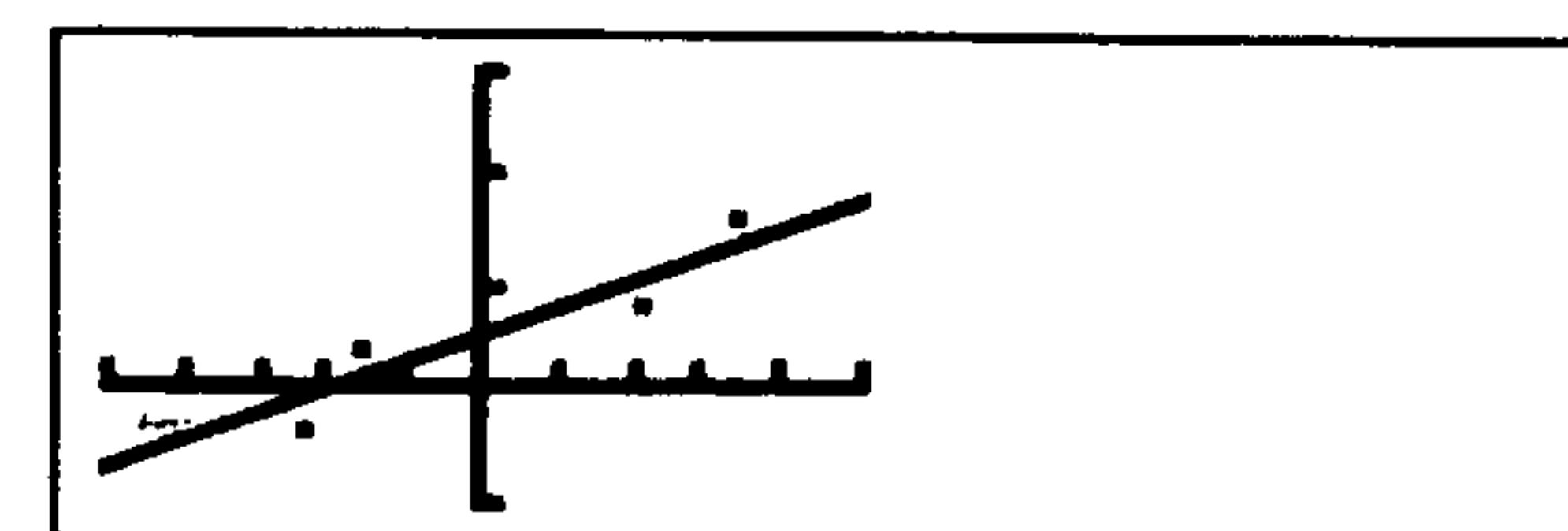
- ④ Entrada de los datos.

SHIFT (-) 9 SHIFT , SHIFT (-) 2 DT
 SHIFT (-) 5 SHIFT , SHIFT (-) 1 DT
 SHIFT (-) 3 SHIFT , 2 DT
 1 SHIFT , 3 DT
 4 SHIFT , 5 DT
 7 SHIFT , 8 DT



- ⑤ Trazado del gráfico.

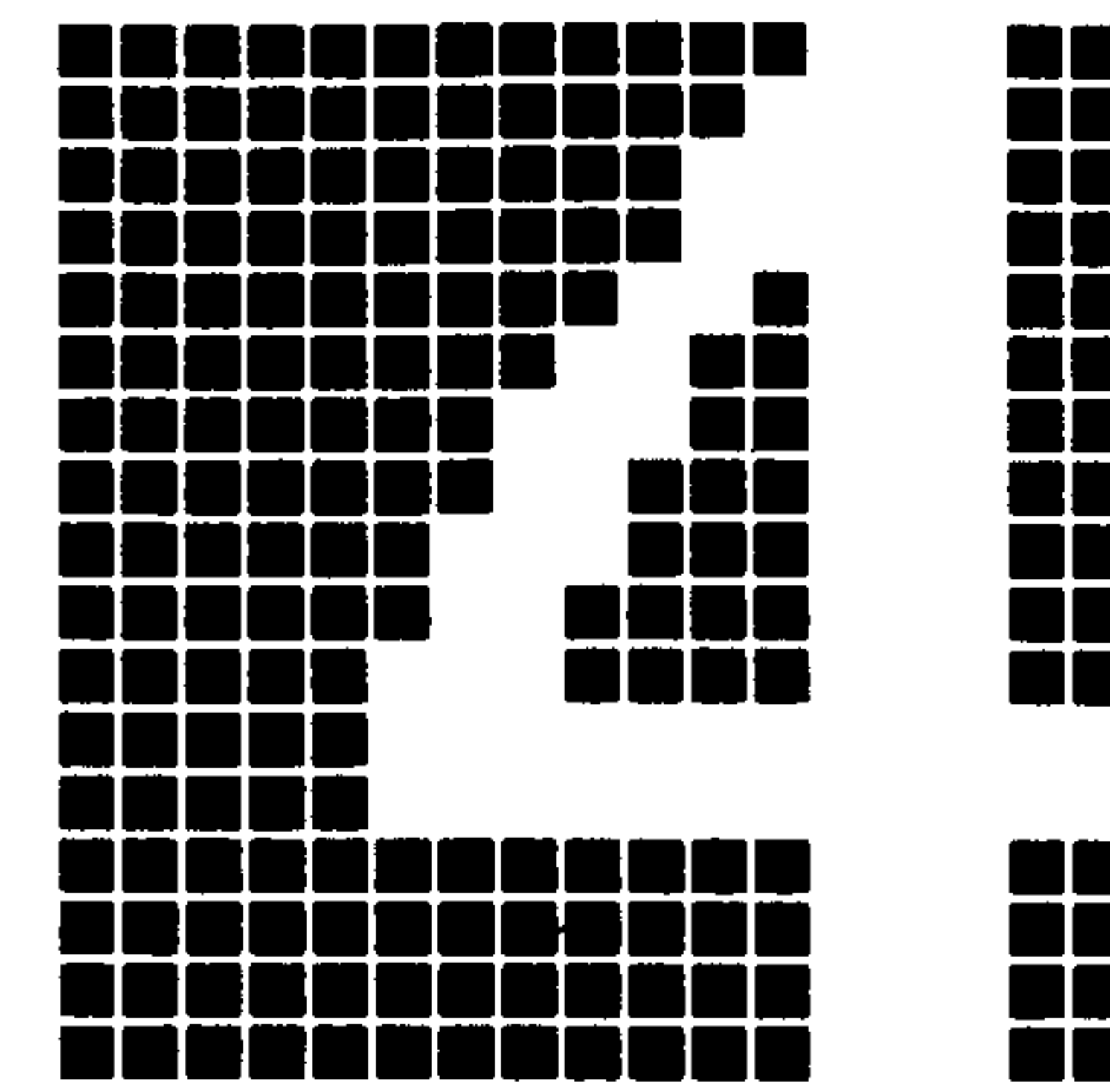
Graph SHIFT Line 1 EXE



*En caso de entrarse un dato que exceda los límites establecidos, el punto que le corresponde no se visualiza en la pantalla.

*En caso de no entrarse dato alguno y de proseguirse con la siguiente operación (Graph SHIFT Line 1 EXE), aparece un error Ma.

*Cuando se establecen los límites, Xmin debe ser menor que Xmax.



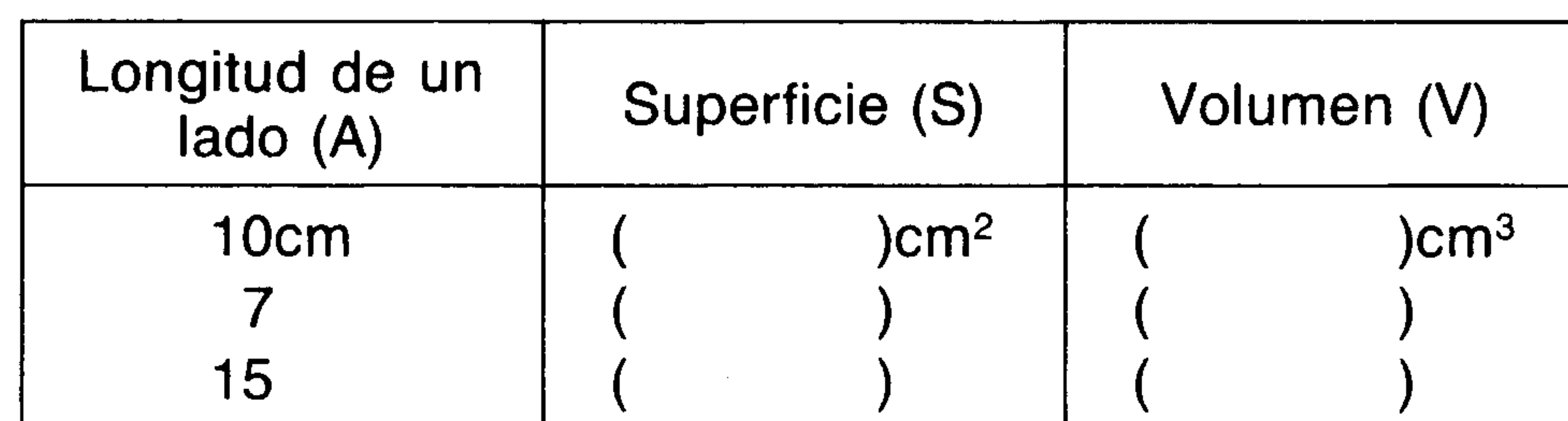
Cálculos por programa

-
- 4-1 ¿Qué es un programa?
 - 4-2 Verificación y compaginación de programas
(corrección, inserción y supresión)
 - 4-3 Depuración de programas (corrección de errores)
 - 4-4 Cuenta del número de pasos de programa
 - 4-5 Areas de programa y modos de cálculo
 - 4-6 Borrado de programas
 - 4-7 Convenientes mandos de programa
 - 4-8 Memorias matriciales
 - 4-9 Visualización de caracteres alfanuméricos y símbolos
 - 4-10 Uso de la función para gráficos en programas



Esta unidad permite la utilización de programas para simplificar la ejecución de cálculos recurrentes. Gracias a ello, se pueden ejecutar fórmulas en forma consecutiva, del mismo modo que se realiza con las sentencias múltiples en los cálculos manuales. De aquí en más veremos las aplicaciones de los programas con la ayuda de ejemplos ilustrativos.

Averigue la superficie y el volumen de una octaedro regular dada la longitud de un lado.



***Llene los paréntesis**

Para una superficie S , el volumen V y el lado A , S y V para un octaedro regular se definen del siguiente modo:

$$S = 2\sqrt{3}A^2 \qquad V = \frac{\sqrt{2}}{3}A^3$$

Se denomina programación al proceso de creación de un programa en base a una fórmula de cálculo. Este programa lo desarrollaremos en base a las fórmulas dadas arriba. La base de la programación son los cálculos manuales, por lo que se recomienda familiarizarse con estos últimos antes de comenzar con la programación.

Superficie (S): 2 ☒ 3 Valor numérico A ☒

Volumen (V): $\sqrt{} 2 \div 3 \times$ Valor numérico A $x^y 3$ **EXE**





En el ejemplo citado, se utiliza dos veces el mismo valor numérico A; sería, entonces, muy conveniente almacenar este valor en la memoria A antes de usarlo en el cálculo.

Valor numérico A

2 \times $\sqrt{}$ 3 \times ALPHA A x^2 EXE S

☐ $\sqrt{}$
☐ 2
 ☐ \div
☐ 3
 ☐ \times
☐ ALPHA
 ☐ A
 ☐ x^y
☐ 3
 ☐ EXE
 V

Con esta unidad, las operaciones que se realizan en los cálculos manuales pueden usarse como se encuentran en un programa. Una vez iniciada la ejecución del programa, la misma continuará en orden sin detenerse. Es por ello que se necesitan mandos que requieran la entrada de los datos y para visualizar los resultados en la pantalla. El mando para solicitar la entrada de información es “?”, mientras que el utilizado para visualizar los resultados es “▲”.

La incorporación de un mando “?” en un programa hará que su ejecución se interrumpa momentáneamente y que aparezca el signo “?” en la pantalla mientras la unidad espera la entrada de la información necesaria. Este mando no puede usarse independientemente, y se combina siempre con  con el formato “   nombre de memoria”. Por ejemplo, para almacenar un valor numérico en la memoria A:

? → A

Quando en la pantalla aparece el signo “?”, se pueden entrar mandos para cálculos y números hasta 127 pasos.

El mando “▲”, por su lado, hace que el programa se interrumpa momentáneamente y que se visualicen el resultado de la última fórmula o caracteres y símbolos alfanuméricos (ver la página 135), según sea necesario. Como los resultados finales se visualizan automáticamente, este mando puede omitirse al final de todo programa. Sin embargo, si se especifica el modo BASE-N para la conversión de números de diferente raíz durante un programa, no omita el final “▲”.

Hagamos uso de estos dos mandos para el ejemplo citado:

Entrada en la memoria A Visualización de S

☐ $\sqrt{}$ 2 ☐ \div 3 ☐ \times ☐ ALPHA ☐ A ☐ x^y 3
 ↑ Se omite “ \blacktriangle ”

Ya, con esto, completamos el programa.

③ Almacenamiento de programas

El almacenamiento de programas se lleva a cabo en el modo WRT, el cual se especifica presionando **MODE** **2**.

MODE **2**

Número remanente de pasos

WRT

400

P 0123456789

Al presionarse **MODE** **2**, el sistema cambia al modo WRT. En este modo, se visualiza en el extremo superior derecho el número remanente de pasos (ver la página 116). Este número disminuye gradualmente siempre que se almacena un programa o si se agregan memorias adicionales. Sin ningún programa almacenado y cuando el número de memorias es igual a 26 (número original), el número de pasos debe ser de 400.

Los números de mayor tamaño que aparecen debajo indican las áreas de programa (ver la página 117). Si la letra “P” aparece seguida por los números 0 hasta el 9, significa que no hay ningún programa almacenado en las áreas P0 hasta P9. El cero parpadeante indica que el área de programa en curso es P0. Las áreas de programa que contienen ya algún programa aparecen indicadas por el símbolo “—” en lugar del número que le correspondería.

WRT

226

P 01_34_6789

Almacenemos, entonces, el programa anterior en el área de programa P0 (indicada por el cero parpadeante):

EXE (Comienzo del almacenamiento)

WRT

Número de pasos usados para el área de programa P0.

ALPHA **?** **→** **ALPHA** **A** **:** **2** **×** **✓** **3**
× **ALPHA** **A** **x²**

WRT

011

?→A:2x√3xA²

ALPHA **▲**

WRT

012

→A:2x√3xA²

✓ **2** **÷** **3** **×** **ALPHA** **A** **x^y** **3**

WRT

020

2√2÷3xAx^y3

El programa se almacena una vez completado este procedimiento.
*Una vez almacenado el programa, presione **MODE** **1** para volver al modo RUN.

④ Ejecución del programa

Los programas se ejecutan en el modo RUN (**MODE** **1**). La tecla **Prog** especifica el área de programa que desea ejecutarse.

Para ejecutar P0: **Prog** **0** **EXE**

Para ejecutar P3: **Prog** **3** **EXE**

Para ejecutar P8: **Prog** **8** **EXE**

Ejecutemos, entonces, el programa almacenado. La superficie (S) y el volumen (V) del octaedro regular se calculan en nuestro ejemplo del siguiente modo:

Longitud de un lado (A)	Superficie (S)	Volumen (V)
10cm	(346,4101615)cm²	(471,4045208)cm³
7	(169,7409791)	(161,6917506)
15	(779,4228634)	(1590,990258)

MODE **1**

Prog **0** **EXE**

?

1 **0** **EXE**
(Valor de A)

Disp

346.4101615

Indica una respuesta visualizada por ▲.
(S cuando A = 10)

EXE

471.4045208

(V cuando A = 10)

Prog **0** **EXE**

?

7 **EXE**
(Valor de A)

Disp

169.7409791

(S cuando A = 7)

EXE

161.6917506

(V cuando A = 7)

Prog 0 EXE

?

1 5 EXE

(Valor de A)

779.4228634

Disp

(S cuando A = 15)

EXE

1590.990258

(V cuando A = 15)

*Los cálculos por programa se llevan a cabo automáticamente por cada pulsación de EXE luego de la entrada de cada dato o después de haberse obtenido el resultado.

*Inmediatamente después que se ejecuta un programa en P0 mediante la pulsación de Prog 0 EXE como en este ejemplo, la función de repetición almacena el mando Prog 0. De este modo, el mismo programa puede ejecutarse cuantas veces se desee mediante la sencilla pulsación de la tecla EXE.

Operación Prog 0 EXE (Ejecución del programa en P0)
 10 EXE (Entrada de 10 para A)
 EXE (Visualización de V cuando A = 10)
 EXE (Otra ejecución)
 7 EXE (Entrada de 7 para A)
 EXE (Visualización de V cuando A = 7)
 ...

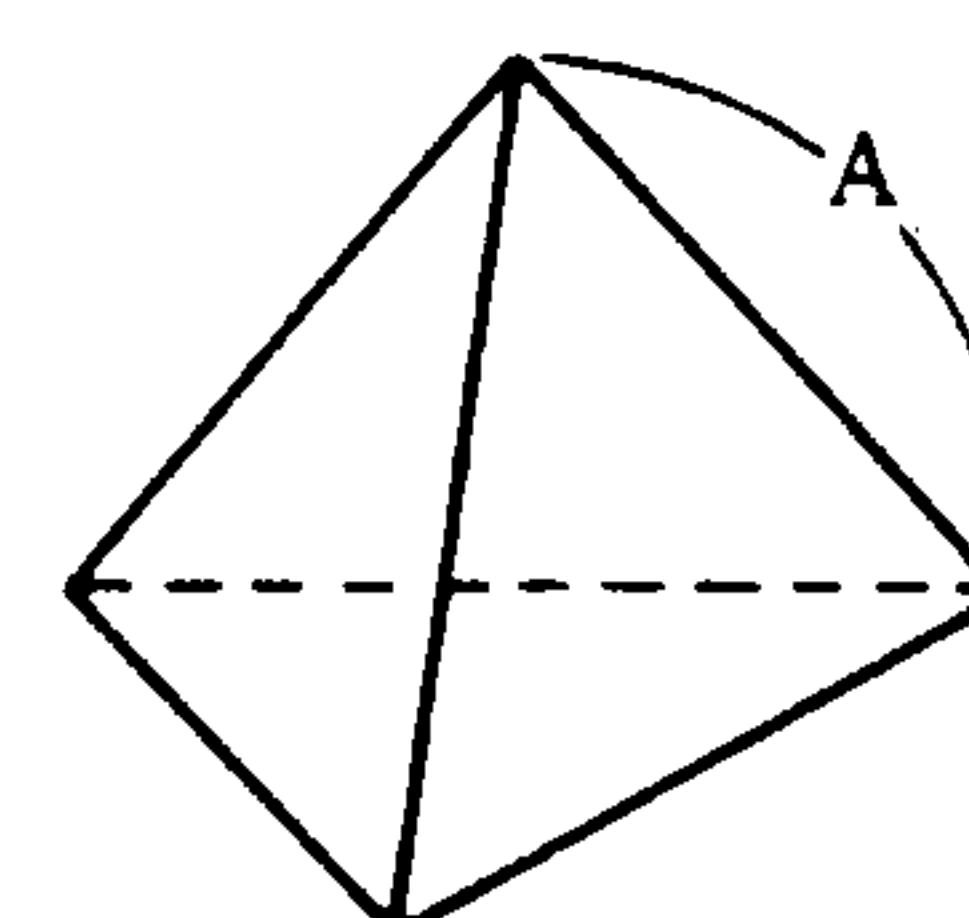
4-2 Verificación y compaginación de programas (corrección, inserción y supresión)

El contenido de los programas almacenados puede verificarse mediante su obtención en el modo WRT. Una vez especificada el área de programa deseada, mediante el uso de \leftarrow y \rightarrow en el modo WRT (MODE 2), el contenido del programa puede visualizarse presionando la tecla EXE. Una vez con el programa en pantalla, el uso de las teclas \rightarrow o \leftarrow permite avanzar por pasos con el fin de verificar el programa.

Cualquier error que se detecte en el programa, podrá corregirse en lo que se denomina "compaginación de programas". A continuación crearemos un programa nuevo modificando el programa del ejemplo anterior (para averiguar la superficie y el volumen de un octaedro regular).

Ejemplo

Averigue la superficie y el volumen de un tetraedro regular dada la longitud de un lado.



Longitud de un lado (A)	Superficie (S)	Volumen (V)
10 cm	() cm ²	() cm ³
7,5	()	()
20	()	()

① Fórmulas

Para una superficie S, el volumen V y el lado A, S y V para un tetraedro regular se definen del siguiente modo:

$$S = \sqrt{3}A^2 \quad V = \frac{\sqrt{2}}{12}A^3$$

② Programación

Como en el ejemplo anterior, se almacena la longitud del lado en la memoria y se confecciona entonces el programa,

Valor numérico A \rightarrow ALPHA A EXE

$\sqrt{3} \times$ ALPHA A x^2 EXE S

$\sqrt{2} \div 12 \times$ ALPHA A x^3 EXE V

Una vez creado el programa, el mismo será del siguiente modo:

ALPHA ? \rightarrow ALPHA A : $\sqrt{3} \times$ ALPHA A x^2 ALPHA \downarrow

$\sqrt{2} \div 12 \times$ ALPHA A x^3

③ Compaginación del programa

Ante todo, comparemos los dos programas vistos hasta el momento.

Octaedro: ALPHA ? → ALPHA A : 2 X ✓ 3 X ALPHA A x² ALPHA ▴
✓ 2 ÷ 3 X ALPHA A x² 3
Tetraedro: SHIFT ? → ALPHA A : ✓ 3 X ALPHA A x² SHIFT ▴
✓ 2 ÷ 12 X ALPHA A x² 3

El programa del octaedro puede modificarse para obtener el programa para el tetraedro mediante la supresión de la parte subrayada con una línea ondulante y cambiando las partes subrayadas con líneas rectas. En la práctica, el procedimiento sería el siguiente:

MODE 2

WRT 380

P _123456789

EXE

WRT 000

? → A : 2 x √ 3 x A² ▴

Cursor ubicado al comienzo. Pulse SHIFT EXE para llevarlo al final.

→ → → →

DEL DEL

WRT 004

? → A : √ 3 x A² ▴ √ 2

Desplace el cursor hasta la posición que desea borrarse y borre dos caracteres.

→ x 9 veces

SHIFT INS 1 2

WRT 015

√ 3 x A² ▴ √ 2 ÷ 12 3

Inserte dos caracteres.

SHIFT INS DEL

WRT 015

√ 3 x A² ▴ √ 2 ÷ 12 X

Borre el 3 innecesario.

MODE 1

WRT

Dé por terminada la corrección. Vuelva al modo RUN.

- 110 -

④ Ejecución del programa

Ejecutemos, entonces, el programa corregido.

Longitud de un lado (A)	Superficie (S)	Volumen (V)
10 cm	(173,2050808)cm²	(117,8511302)cm³
7,5	(97,42785793)	(49,71844555)
20	(692,820323)	(942,8090416)

MODE 1

Prog 0 EXE

?

1 0 EXE

173. 2050808

EXE

117. 8511302

Prog 0 EXE

?

7 . 5 EXE

97. 42785793

EXE

49. 71844555

Prog 0 EXE

?

2 0 EXE

692. 820323

EXE

942. 8090416

- 111 -

< Resumen >

	Operación	Teclas usadas
Verificación de programa	<ul style="list-style-type: none">● Especificación de modo WRT● Especificación de área de programa (omitido si P0)● Inicio de verificación● Verificación de contenidos	<div>MODE 2</div> <div>← →</div> <div>EXE</div> <div>← →</div>
Corrección	<ul style="list-style-type: none">● Mueva el cursor a la posición a ser corregida.● Presione las teclas correctas.	<div>← →</div>
Borrado	<ul style="list-style-type: none">● Mueva el cursor a la posición a ser corregida.● Borre	<div>← →</div> <div>DEL</div>
Inserción	<ul style="list-style-type: none">● Mueva el cursor a la posición en donde se va a realizar la inserción.● Especifique el modo de inserción.● Presione las teclas deseadas.	<div>← →</div> <div>SHIFT INS</div>

4-3 Depuración de programas (corrección de errores)

Suele suceder con frecuencia que la ejecución de un programa hace aparecer un mensaje de error o no da los resultados esperados. Ello se debe, por lo general, a errores inherentes al programa en sí. La corrección de tales errores es lo que se denomina “depuración de programas”.

■ Depuración cuando aparece un mensaje de error

Los mensajes de error aparecen en la pantalla con el siguiente formato:



El mensaje de error le indica al programador en qué área de programa se encuentra el error (P0 a P9). También le informa del tipo de error y le da una idea de las medidas que puede tomar para corregirlo.

■ Mensajes de error

Hay un total de 7 mensajes de error.

- ① **Syn ERROR** (error de sintaxis)
Indica que hay un error en la fórmula o en el uso de alguno de los mandos.
- ② **Ma ERROR** (error matemático)
Indica que el resultado del cálculo es mayor que 10^{100} , que se ha intentado la ejecución de una operación ilógica (división por cero, por ej.) o que la entrada de alguno de los argumentos excede la gama de entrada de la función implicada.
- ③ **Go ERROR** (error de salto)
Indica que falta la Lbl correspondiente al mando Goto (ver la página 120) o que el área de programa (ver la página 117) especificada por el mando Prog (ver la página 127) no contiene programa alguno.
- ④ **Ne ERROR** (error de inclusión de subrutinas)
Indica que se ha excedido el nivel permitido para la inclusión de subrutinas (ver la página 127).
- ⑤ **Stk ERROR** (error de memoria temporal)
Indica que el cálculo que desea realizarse excede la capacidad de las memorias temporales para valores numéricos o para mandos (ver la página 32).

⑥ **Mem ERROR** (error de memoria)

Indica que se ha intentado el uso de las memorias adicionales sin haberlas agregado aún.

⑦ **Arg ERROR** (error de argumento)

Aparece cuando el argumento de un mando o cierta especificación en un programa excede la gama de entrada permitida (por ej.: Sci 10, Goto 11).

Es imposible proseguir con la operación en curso cuando aparece un mensaje de error. En dicho caso, pulse **AC**, **☐**, o **☐**.

La pulsación de **AC** suspende la condición de error y hace posible una nueva entrada desde el teclado. En este caso, se mantiene el modo RUN.

Las teclas **☐** y **☐** hacen que se suspenda el estado de error y que el sistema entre en el modo WRT. En dicho caso, el cursor aparece en la posición donde el error ocurrió, para así permitir la corrección del mismo.

■ Consideraciones para cada tipo de error

A continuación algunas consideraciones para cada tipo de error:

① **Syn ERROR**

Verifique que no haya errores en el programa.

② **Ma ERROR**

Para aquellos cálculos que requieren el uso de memorias, verifique que los valores en las mismas no excedan la gama permitida para los argumentos. Este tipo de error ocurre con frecuencia cuando se intenta una división por 0 o se intenta la obtención de la raíz cuadrada de un número negativo.

③ **Go ERROR**

Verifique la presencia de la etiqueta Lbl *n* correspondiente cuando se utiliza Goto *n*. Corrobore, por otro lado, que el programa en P *n* haya sido entrado correctamente cuando se utiliza Prog *n*.

④ **Ne ERROR**

Verifique que el mando Prog no se utiliza en el área de programa de destino para retornar la ejecución al área del programa original.

⑤ **Stk ERROR**

Verifique que la fórmula no sea tan extensa como para superar la capacidad de las memorias temporales. En caso contrario, divida la fórmula en dos o más partes.

⑥ **Mem ERROR**

Verifique que se hayan agregado las memorias necesarias del modo apropiado, mediante "**MODE** **☐** *n* **EXE**" (Defm). Cuando se utilizan memorias matriciales (ver la página 130), corrobore que no haya errores en los subíndices.

⑦ **Arg ERROR**

Corrobore que los valores especificados por **MODE** **7** (Sci) o **MODE** **8** (Fix) estén dentro de los límites del 0 al 9. Verifique, también, que los valores especificados por los mandos Goto, Lbl y Prog estén dentro de los límites del 0 al 9. Asegúrese, además, de que el número de memorias agregadas por medio de **MODE** **☐** (Defm) no exceda el número remanente de pasos de programa, y que el valor utilizado no sea negativo.

4-4 Cuenta del número de pasos de programa

Esta unidad tiene una capacidad de 400 pasos de programa. El número de pasos, el cual disminuye cada vez que se almacena un programa, indica la capacidad de memoria disponible para el almacenamiento de programas. Este número disminuye, también, cuando se convierten pasos de programa en memorias adicionales (ver la página 41).

Hay 2 métodos para determinar el número remanente de pasos:

- 1 La pulsación de **MODE** **•** **EXE** en el modo RUN hace que se visualicen en la pantalla el número remanente de pasos y la cantidad de memorias disponibles.

Ejemplo

MODE **•** **EXE**

M-26 S-381

- 2 En el modo WRT (**MODE** **2**), el número remanente de pasos aparece en el extremo superior izquierdo. También se visualizan los estados de cada área de programa.

MODE **2**

WRT 381
P 123456789

Número remanente de pasos

Básicamente, cada función equivale a 1 paso; no obstante, algunas funciones requieren el uso de 2 pasos.

- Funciones de 1 paso: sin, cos, tan, log, (,), :, A, B, 1, 2, 3, etc.
- Funciones de 2 pasos: Lbl 1, Goto 2, Prog 8, etc.

Cada paso puede verificarse mediante el desplazamiento del cursor.

Ejemplo

WRT 000
?→A:√3×A² √2

Posición del cursor

Aquí, cada pulsación de las teclas para desplazar el cursor (**←** o **→**) hace que el cursor avance al siguiente paso en una u otra dirección.

← **←** **←** **←** **←** **←**

WRT 006
?→A:√3×A² √2

6° paso

4-5 Areas de programa y modos de cálculo

Esta unidad posee 10 áreas de programa (P0 hasta P9) para el almacenamiento de programas. Estas áreas de programas se utilizan de la misma manera y permiten el almacenamiento simultáneo de 10 programas independientes. Pueden almacenarse, si se desea, una programa o rutina principal y programas secundarios o subrutinas. El número total de pasos disponible para el almacenamiento en las áreas P0 hasta P9 es de 400.

La especificación de las áreas de programa se lleva a cabo del siguiente modo:

Modo RUN: Se presiona cualquier tecla desde 0 hasta 9 después de haber presionado la tecla **Prog**. Finalmente, se pulsa **EXE**.

Ejemplo

P0 **Prog** **0** **EXE**
P8 **Prog** **8** **EXE**

*En este modo, la ejecución del programa comienza con la pulsación de la tecla **EXE**.

Modo WRT: Se desplaza el cursor hasta la posición por debajo del área deseada (usando las teclas **←** o **→**) y se presiona la tecla **EXE**.

En este caso, sólo se visualizan las áreas de programa que se encuentran desocupadas. El símbolo “—” indica que el área contiene ya un programa.

Ejemplo

WRT 292
P 123—67—9

Programas ya almacenados en estas áreas de programa.

■ Especificación del área de programa y del modo de cálculo en el modo WRT

Además de los cálculos con funciones normales, de los cálculos y conversiones con números binarios, octales, decimales y hexadecimales, y de los cálculos de desviación estándar y de regresión por programa, debe especificarse un modo de cálculo. Las especificaciones del modo por programa y del área de programa se llevan a cabo al mismo tiempo.

Primeramente se especifica el modo WRT (**MODE** **2**) y luego se especifica el modo de cálculo. Luego se especifica el área de programa y, al pulsarse **EXE**, se memoriza el modo de cálculo en el área de programa. De este modo, los programas almacenados serán acompañados por el modo de cálculo.

Ejemplo Memorización del modo BASE-N en P2

MODE 2

WRT 400
P 0123456789

Suponiendo que no hay nada almacenado.

⇒⇒

WRT 400
P 0123456789

Especificación de P2.

MODE

BASE-N WRT 400
P 0123456789

Especificación del modo BASE-N.

EXE

BASE-N WRT 000

De este modo, se memoriza el modo de cálculo en el área de programa.

■Precauciones en cuanto a los modos de cálculo

Todas las teclas disponibles en cada modo de cálculo pueden incorporarse en los programas; no obstante, según el modo de cálculo, algunos mandos o funciones no podrán usarse.

Modo BASE-N

- No pueden llevarse a cabo cálculos con funciones.
- No pueden especificarse unidades de medida angular.
- Pueden usarse todos los programas.
- Asegúrese de incluir un ▲ al final del resultado para volver al modo de cálculo previo una vez que termina la ejecución del programa. En caso contrario, podrá aparecer un decimal o un error.

Modo SD1, SD2

- De las funciones, no pueden usarse Abs y $\sqrt[3]{}$.
- De los mandos de programa, no pueden usarse Dsz, > y <.

Modo LR1, LR2

- De las funciones, no pueden usarse Abs y $\sqrt[3]{}$.
- De los mandos de programa, no pueden usarse ⇒, =, ≠, lsz, ≥, ≤, Dsz, > y <.

4-6 Borrado de programas

Los programas se borran en el modo PCL. Para especificar este modo, presione MODE 3 (aparece en la pantalla la indicación "PCL"). Hay 2 métodos para borrar programas: por área de programa individual o los que se encuentran en todas las áreas.

■Borrado de un solo programa

Para borrar un programa que se encuentra en una área determinada, especifique el modo PCL y presione AC después de haber especificado dicha área.

Ejemplo Borrado del programa en P3.

MODE 3

PCL 302
P 12_45678_

Ya hay programas almacenados en P0, P3 y P9.

⇒⇒⇒

PCL 302
P _12_45678_

Coloque el cursor en la posición de P3.

AC

PCL 345
P_12345678_

El número 3 aparece después de borrar.

MODE 1

Vuelva al modo RUN.

■Borrado de todos los programas

Para borrar todos los programas almacenados en las áreas P0 hasta P9, especifique el modo PCL y presione SHIFT Mcl y finalmente MODE 1.

Ejemplo Borrado de los programas almacenados en P0, P4, P8 y P9.

MODE 3

PCL 273
P 123_567_

SHIFT Mcl

PCL 400
P 0123456789

MODE 1

4-7 Convenientes mandos de programa

Los programas que se utilizan en esta unidad se basan en los procedimientos para los cálculos manuales. Se dispone, no obstante, de mandos especiales para programas, los cuales permiten hacer juicios o repetir la ejecución de fórmulas si se cumplen o no determinados condiciones.

He a continuación algunos mandos muy convenientes para su utilización en programas.

■ Mandos de salto

Los mandos de salto se utilizan para modificar el flujo de la ejecución del programa. Los programas se ejecutan ordenadamente, desde el primer paso hasta el último. Este sistema no es muy conveniente cuando se desea repetir un mismo cálculo o cuando es necesario transferir la ejecución a otra fórmula. es, justamente, en estos casos que los mandos de salto son sumamente efectivos. Hay 3 tipos de mandos de salto: los incondicionales simples, los cuales tienen un solo destino; los condicionales, los cuales deciden el destino del salto en base a si se cumplió o no cierta condición; y los saltos con cuenta, los cuales incrementan o disminuyen el valor en una memoria dada y deciden el destino del salto después de verificar si dicho valor es igual a 0 o no.

■ Salto incondicional

Los saltos incondicionales se componen de "Goto" y "Lbl". Cuando un programa llega a una sentencia "Goto n ", donde n es igual a un número desde 0 hasta 9, la ejecución salta a "Lbl n ", donde n tiene el mismo valor que en Goto n . El salto incondicional se utiliza con frecuencia en programas simples para retornar la ejecución al comienzo, con el fin de repetir los cálculos total o parcialmente. Se combinan, también, con saltos condicionales y con cuenta.

Ejemplo Agreguemos "Goto 1" y "Lbl 1" al programa utilizado para averiguar la superficie y el volumen de un tetraedro, con el fin de permitir la repetición de los cálculos.

El programa anterior contenía

$?, \rightarrow, A, :, \sqrt{\quad}, 3, \times, A, x^2, \blacktriangleleft,$
 $\sqrt{\quad}, 2, \div, 1, 2, \times, A, x^3, 3$

19 pasos

**De aquí en más, los pasos los separaremos por medio de comas (,), para una mayor claridad.*

Agregue "Goto 1" al final de programa y "Lbl 1" (destino del salto) al comienzo del mismo.

Sin embargo, si el programa se deja como está, el volumen no aparecerá en la pantalla y la ejecución pasará inmediatamente a la entrada del lado, al comienzo del programa. Para evitar esta situación, inserte un mando de visualización (\blacktriangleleft) frente a "Goto 1".

El programa completo, con el salto incondicional agregado, se verá del siguiente modo:

Lbl, 1, :, ?, \rightarrow , A :, $\sqrt{\quad}$, 3, \times , A, x^2 , \blacktriangleleft ,
 $\sqrt{\quad}$, 2, \div , 1, 2, \times , A, x^3 , 3, \blacktriangleleft , Goto, 1

25 pasos

Ejecutemos el programa.

**Para detalles sobre la entrada y compaginación de los programas, remítase a las secciones 4-1 y 4-2.*

**De aquí en más, sólo se muestran los resultados de los cálculos.*

Prog 0 EXE
1 0 EXE
EXE
EXE
7 . 5 EXE
EXE
EXE

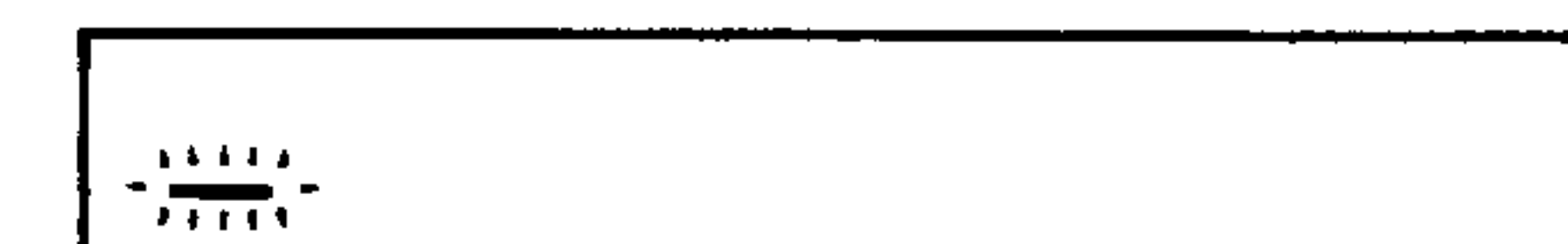
?
173. 2050808
117. 8511302
?
97. 42785793
49. 71844555
?

Almacenado en P0.
La longitud del lado es
igual a 10.

La longitud del lado es
igual a 7,5.

Como el programa entra en un ciclo cerrado, su ejecución continuará indefinidamente. Para terminarlo, presione MODE 1.

MODE 1



Además del comienzo del programa, los destinos pueden designarse en cualquier punto dentro del programa.

Ejemplo Cálculo de $y = ax + b$ cuando el valor de x cambia por cada ejecución, y a y b cambian según el cálculo.

Programa

$?, \rightarrow, A, :, ?, \rightarrow, B, :, \text{Lbl}, 1, :, ?, \rightarrow, X, :,$
 $A, \times, X, +, B, \blacktriangleleft, \text{Goto}, 1$

23 pasos

Al ejecutarse este programa, los valores de a y b se almacenan en las memorias A y B, respectivamente. Luego, sólo puede cambiarse el valor de x .

De este modo, se realiza un salto incondicional por medio de "Goto" y "Lbl" y el flujo del programa cambia. Cuando no se encuentra el "Lbl n " que corresponda con el "Goto n " ejecutado, aparece un mensaje de error (Go ERROR).

■ Salto condicional

El salto condicional compara un valor numérico almacenado en una memoria con una constante o un valor numérico que se encuentra en otra memoria. Si se cumple la condición, se ejecuta la sentencia que se encuentra a continuación del "⇒", en caso contrario, la ejecución salta la sentencia y continúa con el siguiente ":" o "▲".

Los saltos condicionales tienen el siguiente formato:

Lado izquierdo Operador Lado derecho ⇒ Sentencia { : } * Sentencia
comparativo

*Puede usarse cualquiera de los dos.

En el lado izquierdo y derecho se pueden usar un nombre de memoria (carácter alfabético desde A hasta Z), una constante numérica o fórmulas ($A \times 2$, $D - E$, etc). El operador comparativo es un símbolo de comparación. Hay 6 tipos de operadores: =, ≠, ≥, ≤, > y <.

Lado izquierdo = Lado derecho (lados izquierdo y derecho iguales)

Lado izquierdo ≠ Lado derecho (lados izquierdo y derecho desiguales)

Lado izquierdo ≥ Lado derecho (lado izquierdo mayor o igual que el derecho)

Lado izquierdo ≤ Lado derecho (lado izquierdo menor o igual que el derecho)

Lado izquierdo > Lado derecho (lado izquierdo mayor que el derecho)

Lado izquierdo < Lado derecho (lado izquierdo menor que el derecho)

La indicación "⇒" aparece en la pantalla cuando se presionan **SHIFT** **⇒**. Si la condición se cumple, la ejecución avanza a la sentencia que sigue a "⇒". En caso contrario, se salta la sentencia que se encuentra a continuación de "⇒" y la ejecución pasa a la sentencia que sigue al ":" o "▲" siguiente.

Lado izquierdo Operador Lado derecho ⇒ Sentencia { : } Sentencia
comparativo

Si se cumple

Si no se cumple

Una sentencia es una fórmula de cálculo (sin $A \times 5$, etc.) o un mando de programa (Goto, Prog, etc.) y todo lo que se encuentre hasta el siguiente ":" o "▲".

Ejemplo Si el valor numérico entrado es mayor o igual que cero, calcule la raíz cuadrada del mismo. Si el valor es menor que 0, vuelva a solicitar la entrada de un valor nuevo.

Programa

Lbl, 1, :, ?, →, A, :, A, ≥, 0, ⇒, √, A, ▲, Goto, 1 16 pasos

En este programa, el valor numérico entrado se almacena en la memoria A, donde se prueba para determinar si es mayor, igual o menor que 0. Si el contenido de la memoria A es mayor o igual a 0 (y no menor que éste), se ejecutará la sentencia (fórmula de cálculo) ubicada entre "⇒" y "▲", y luego Goto 1 devuelve el control a Lbl 1. Si el contenido de la memoria A es menor que 0, se salta la ejecución de la sentencia a continuación del "▲" y se retorna, por la acción de Goto 1, a la posición donde se encuentre Lbl 1.

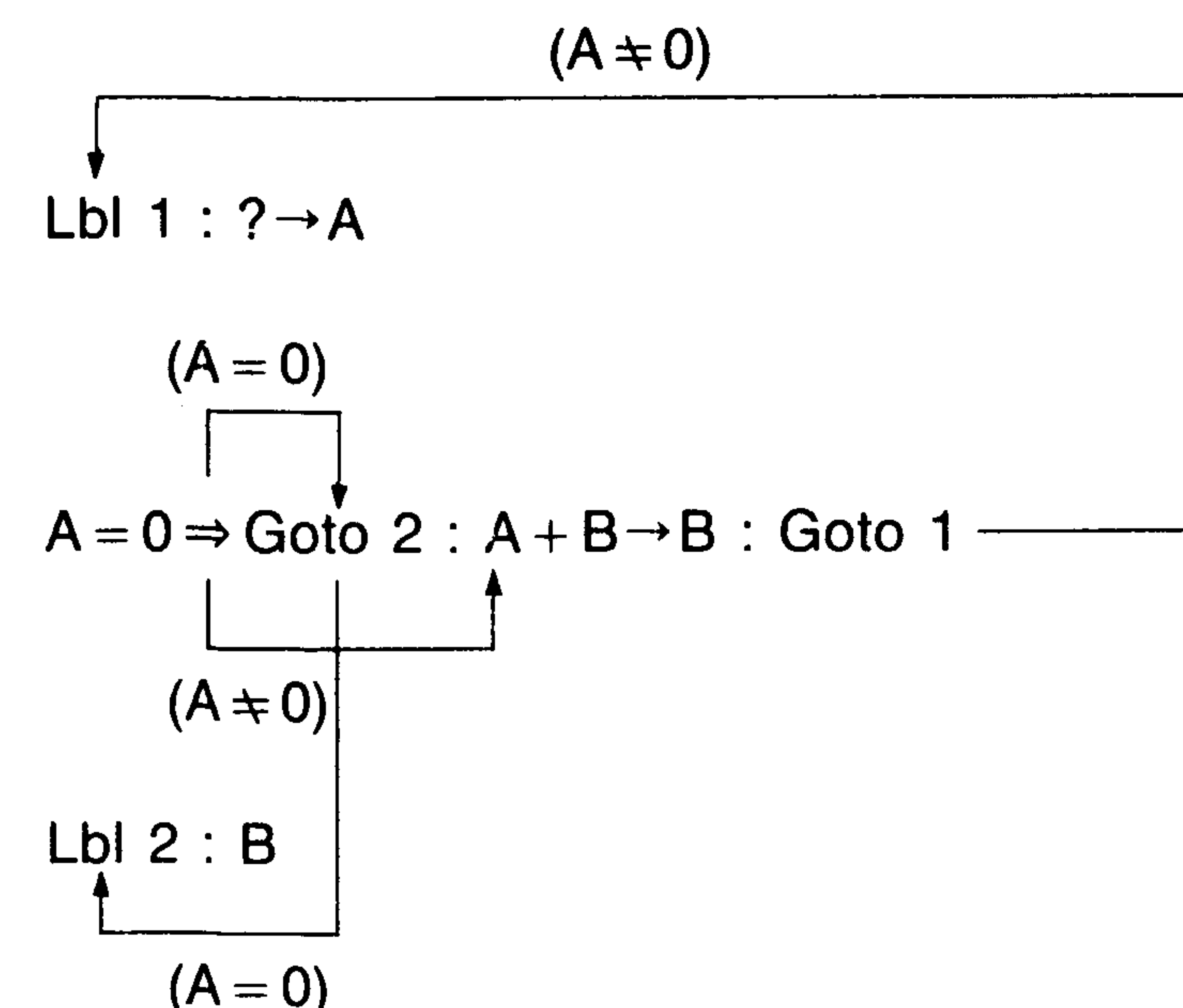
Ejemplo Cálculo de la suma de valores numéricos entrados. Hace que el total se visualice cuando se entra un 0.

Programa

0, →, B, :,
Lbl, 1, :, ?, →, A, :, A, =, 0, ⇒, Goto, 2, :,
A, +, B, →, B, :, Goto, 1, :,
Lbl, 2, :, B

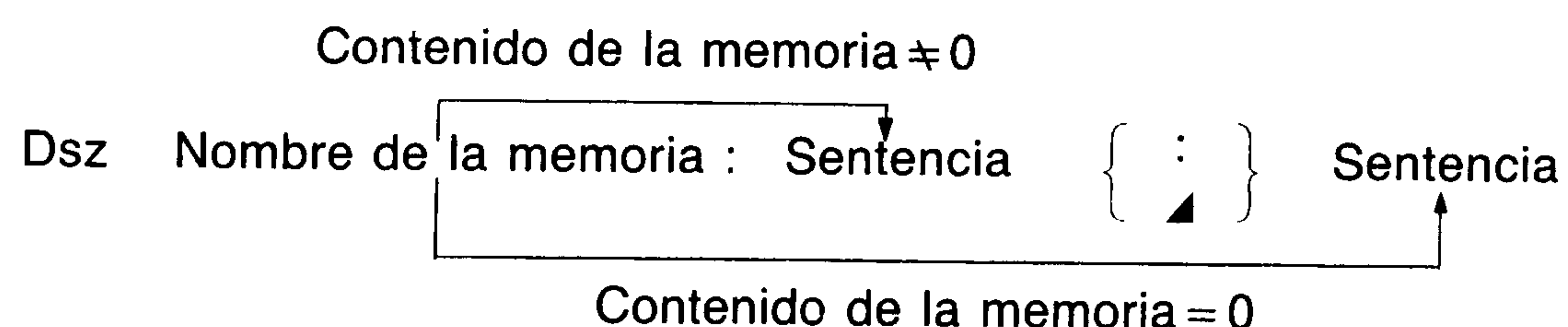
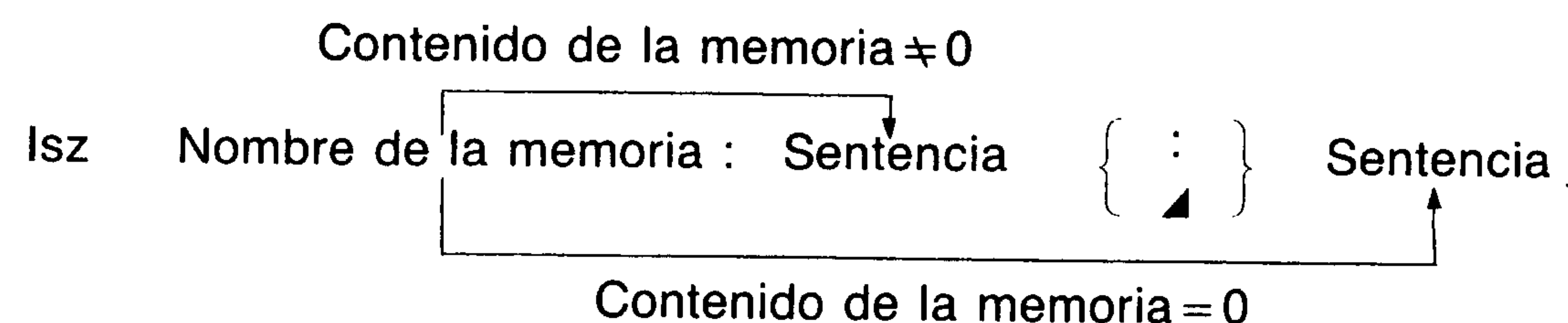
31 pasos

En este programa, se entra un 0 en la memoria B para preparar a la misma para calcular la suma. Luego se determina si el valor entrado por medio de "?→A" es igual a cero mediante el paso " $A = 0 \Rightarrow$ ". Si $A = 0$, Goto 2 hace que se produzca un salto hacia Lbl 2. Si el contenido de la memoria A no es igual a 0, entonces se salta Goto 2 y se ejecuta lo que se encuentra a continuación de los dos puntos siguientes, $A + B \rightarrow B$, y luego el Goto 1 provoca un salto hacia Lbl 1. La ejecución a partir de Lbl 2 hace que se visualice la suma almacenada en la memoria B. Si bien se ha insertado el mando "▲" a continuación de B, el mismo puede omitirse, ya que es al final del programa. La siguiente figura muestra el flujo del programa:



■ Saltos con cuenta

Los saltos con cuenta agregan o restan una unidad al valor almacenado en la memoria especificada. Si el valor llega a 0, se salta la siguiente sentencia y se ejecuta lo que esté a continuación del “:” o “▲” siguientes. El mando “Isz” se utiliza para agregar una unidad al valor en la memoria y decidir lo que se ejecutará a continuación, mientras que el mando “Dsz” resta una unidad y decide.



Ejemplo Agrega una unidad a la memoria A Isz A
 Resta una unidad de la memoria B Dsz B

Ejemplo Determine el promedio de 10 valores numéricos entrados.

Programa

1, 0, →, A, :, 0, →, C, :,
Lbl, 1, :, ?, →, B, :, B, +, C, →, C, :,
Dsz, A, :, Goto, 1, :, C, ÷, 1, 0

32 pasos

En este programa, primeramente se almacena 10 en la memoria A y 0 en la memoria C. La memoria A se utiliza para la cuenta del número de veces que debe ejecutarse el mando Dsz. La memoria C se utiliza para acumular los números entrados, por lo que debe asignársele un 0 al comienzo. El valor numérico entrado en respuesta a “?” se almacena en la memoria B, y luego la suma de los valores entrados se almacena en la memoria C por medio de “B + C → C”. La sentencia Dsz A resta, entonces, una unidad al valor almacenado en la memoria A. Si el resultado no es igual a 0, se ejecuta la siguiente sentencia, Goto 1 en este caso. Si el resultado es igual a 0, se salta el Goto 1 que se encuentra a continuación y se ejecuta “C ÷ 10”.

Ejemplo Determine la altura a intervalos de un segundo para una bola arrojada al aire con una velocidad inicial de V_m/seg y un ángulo de S° . La fórmula se expresa: $h = V \sin \theta t - \frac{1}{2} g t^2$, donde $g = 9,8$ y se hace caso omiso de los efectos de la resistencia al aire.

Programa

Deg, :, 0, →, T, :, ?, →, V, :, ?, →, S, :,
Lbl, 1, :, lsz, T, :, V, ×, sin, S, ×, T, —,
9, •, 8, ×, T, x^2 , ÷, 2, ▲, Goto, 1

38 pasos

En este programa, se especifica la unidad de medida angular y se borra la memoria T. Luego se entran la velocidad inicial y el ángulo en las memorias V y S, respectivamente.

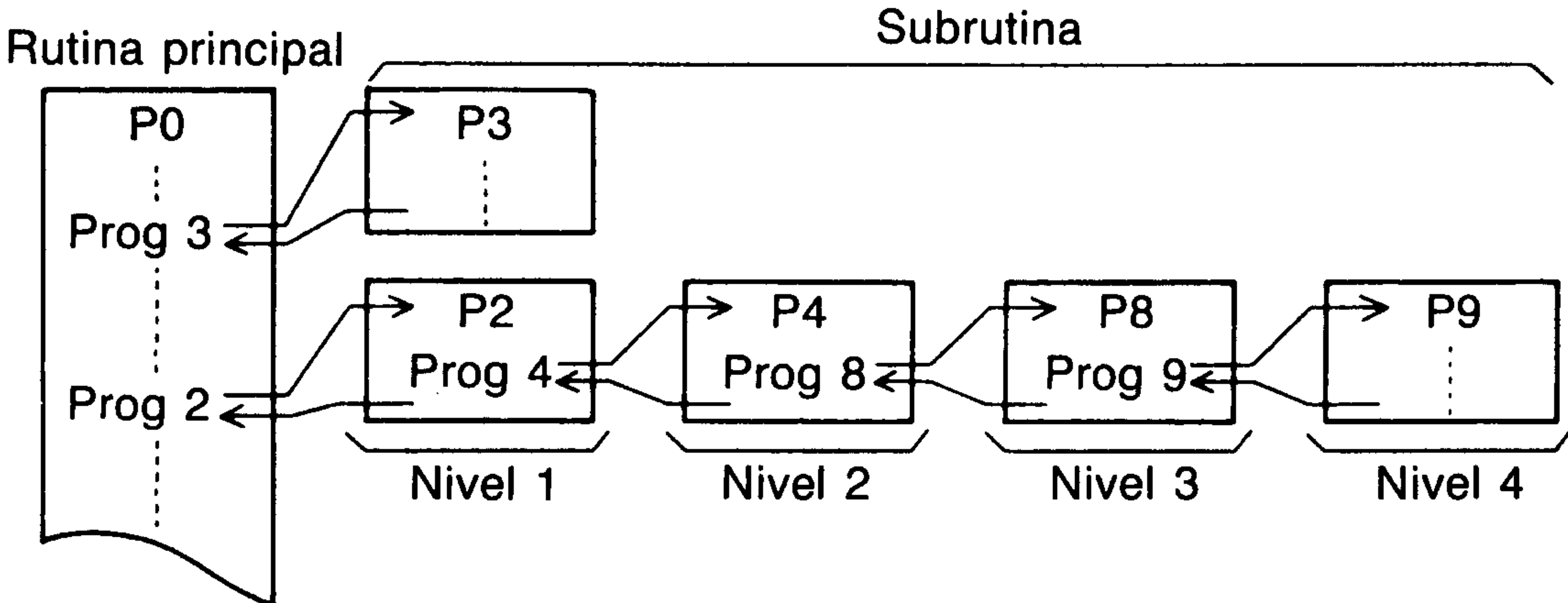
Lbl 1 se utiliza al comienzo del cálculo que debe repetirse. El valor numérico almacenado en T aumenta en una unidad por la acción de Isz T. En este caso, el mando Isz se utiliza con la única finalidad de aumentar el valor en T; el salto subsiguiente no depende de ninguna comparación o decisión. Se bien el mando Isz puede usarse del mismo modo que el mando Dsz, para saltos que requieren decisiones, en este caso se lo utiliza solamente para aumentar un valor. Si en lugar del mando Isz se utiliza otro método como " $T + 1 \rightarrow T$ " por ejemplo, se necesitarían 5 pasos en lugar de los 2 que requiere el método (Isz T). Estos mandos son muy convenientes para ahorrar espacio en la memoria. Cada vez que se aumenta el valor en T, se lleva a cabo el cálculo de acuerdo a la fórmula y se visualiza la altitud. Debe tenerse en cuenta que este programa continua indefinidamente. Cuando desee interrumpirlo, presione **MODE** **1**.

< Resumen >

Mando	Fórmula	Operación
Salto incondicional	Lbl <i>n</i> Goto <i>n</i> (<i>n</i> = número entero desde 0 hasta 9)	Lleva a cabo un salto incondicional hacia el Lbl <i>n</i> correspondiente al Goto <i>n</i> .
Salto condicional	Lado Operador Lado izquierdo comparativo derecho ⇒ Sentencia { : } Sentencia ▲ (Operadores comparativos: =, ≠, >, <, ≥, ≤)	Se comparan los lados izquierdo y derecho. Si se cumple la expresión condicional, se ejecuta la sentencia que se encuentra a continuación de ⇒. Si no se cumple, la ejecución salta a la sentencia que se encuentra a continuación del siguiente : o ▲. Las sentencias incluyen expresiones numéricas, mandos Goto, etc.
Salto con cuenta	Isz Nombre de memoria: Sentencia { : } Sentencia ▲ Dsz Nombre de memoria: Sentencia { : } Sentencia ▲ (Los nombres de memorias consisten de una letra de la A a la Z, A[], etc.).	El valor numérico almacenado en la memoria se aumenta (Isz) o disminuye (Dsz) por una unidad. Si el resultado es igual a 0, se realiza un salto hacia la sentencia a continuación del siguiente : o ▲. Las sentencias incluyen expresiones numéricas, mandos Goto, etc.

■ Subrutinas

Todo programa contenido en una única área de programa se denomina “rutina principal”. Los demás segmentos del programa que se utilizan con frecuencia almacenados en otras áreas de programa se denominan “subrutinas”. Las subrutinas pueden usarse de muchos modos, para facilitar los cálculos. Pueden usarse para almacenar fórmulas que se utilizan reiteradamente; para ello, cada vez que se necesite hacer un cálculo con una subrutina, se lleva a cabo un salto en dirección a la misma.



El mando para saltar a una subrutina es “Prog” seguido por un número de 0 a 9 correspondiente al número de área de programa.

Ejemplo Prog 0 Salto al área P0
 Prog 2 Salto al área P2

Una vez realizado el salto por medio del mando Prog, la ejecución continua desde el comienzo del programa almacenado en el área de programa especificado. Una vez que se ejecuta la subrutina, el programa regresa a la sentencia que se encuentra a continuación del mando Prog *n* que provocó el salto a la subrutina, en el área de programa original. Se puede, además, saltar de una subrutina a otra; esto es lo que se llama “inclusión de subrutinas”. Se permite, en este caso, un máximo de 9 niveles de inclusión (10 saltos de una subrutina a otra, como máximo.) El mensaje Ne ERROR aparece siempre que se excede este límite. Todo intento de saltar a un área donde no hay programa alguno almacenado, hará que aparezca un mensaje Go ERROR.

*Un Goto *n* en una subrutina provocará un salto hacia el Lbl *n* contenido en la misma área de programa.

Ejemplo Ejecute simultáneamente los dos programas presentados previamente para calcular las superficies y los volúmenes de un octaedro y un tetraedro regulares. Expresé el resultado con 3 dígitos decimales.

Este ejemplo hace uso de los dos programas ya explicados. El primer paso a seguir es la especificación del número de dígitos decimales (MODE 7 3).

4-8 Memorias matriciales

■ Uso de memorias matriciales

Hasta el momento, las memorias que hemos utilizado en los programas presentados han sido aquéllas cuyos nombres poseen una única letra, como ser A, B, X o Y.

Las memorias denominadas matriciales permiten el uso de un único nombre (también una letra de la A a la Z) con un subíndice adjunto entre paréntesis.

*Los paréntesis se entran por medio de ALPHA [] y ALPHA [].

Memoria normal	Memoria matricial	
A	A[0]	C[-2]
B	A[1]	C[-1]
C	A[2]	C[0]
D	A[3]	C[1]
E	A[4]	C[2]

El uso apropiado de subíndices acorta los programas y facilita su uso. Los valores negativos usados como subíndices se cuentan en relación a la memoria cero, del modo que se muestra arriba.

Ejemplo Entrada de los números 1 hasta 10 en las memorias A hasta J.

Usando las memorias comunes

1, →, A, :, 2, →, B, :, 3, →, C, :, 4, →, D, :,
5, →, E, :, 6, →, F, :, 7, →, G, :, 8, →, H, :,
9, →, I, :, 10, →, J

40 pasos

Usando las memorias matriciales

0, →, Z, :, Lbl, 1, :, Z, +, 1, →, A, [, Z,], :,
Isz, Z, :, Z, <, 1, 0, ⇒, Goto, 1

26 pasos

En el caso en el cual se utilizan las memorias comunes, la entrada de los valores en cada memoria es ineficiente y demora mucho tiempo. ¿Qué pasaría si deseáramos ver el valor especificado en determinada memoria?

Usando memorias comunes

Lbl, 1, :, ?, →, Z, :,
Z, =, 1, ⇒, A, ▲, Z, =, 2, ⇒, B, ▲,
Z, =, 3, ⇒, C, ▲, Z, =, 4, ⇒, D, ▲,
Z, =, 5, ⇒, E, ▲, Z, =, 6, ⇒, F, ▲,
Z, =, 7, ⇒, G, ▲, Z, =, 8, ⇒, H, ▲,
Z, =, 9, ⇒, I, ▲, Z, =, 10, ⇒, J, ▲,
Goto, 1

70 pasos

Usando memorias matriciales

Lbl, 1, :, ?, →, Z, :, A, [, Z, -, 1,], ▲,
Goto, 1

16 pasos

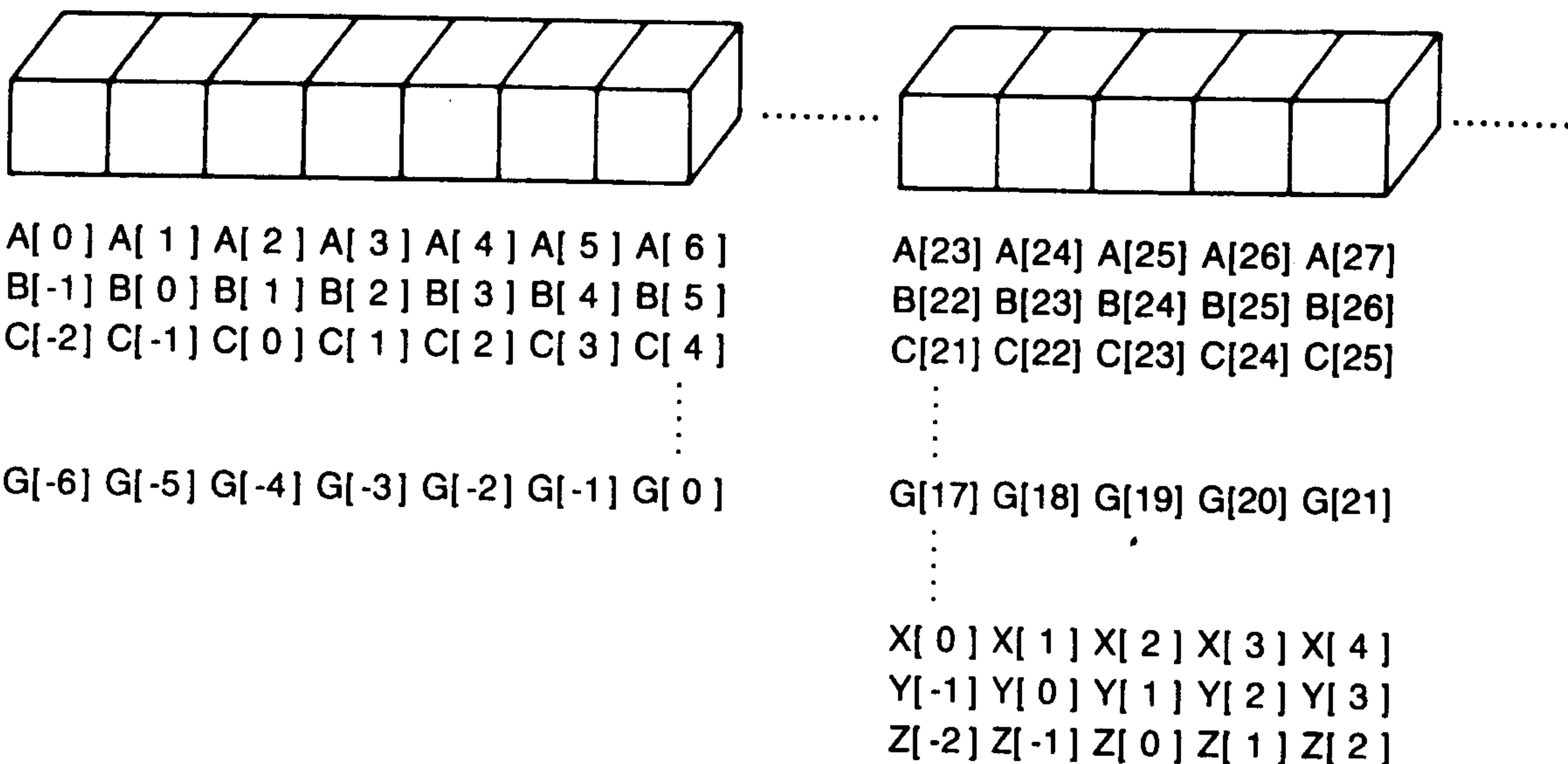
La diferencia está a la vista. Cuando se usan memorias comunes, el valor de entrada se compara uno por uno con el valor asignado a cada memoria (por ej.: A = 1, B = 2, etc.). Con la memorias matriciales, el valor de entrada se almacena inmediatamente, en la memoria apropiada determinada por "[Z - 1]". Para el subíndice, se pueden utilizar fórmulas como Z - 1, A + 10, etc.

■ Precauciones en el uso de memorias matriciales

Cuando se utilizan memorias matriciales, se adjunta un subíndice a un carácter alfabético que representa una memoria común de la A hasta la Z.

Es por ello que debe tenerse cuidado para que no se superpongan las memorias matriciales con las comunes.

La relación entre ambos tipos es la siguiente:



Veamos a continuación un caso de sobreposición de memorias matriciales y memorias comunes. Esta situación debe evitarse siempre.

Ejemplo Almacenamiento de los números 1 al 5 en las memorias A[1] hasta A[5].

```
5, →, C, :, Lbl, 1, :, C, →, A, [, C, ], :,  
Dsz, C, :, Goto, 1, :,  
A, [, 1, ], ▲, A, [, 2, ], ▲, A, [, 3, ], ▲,  
A, [, 4, ], ▲, A, [, 5, ]
```

44 pasos

En este programa, los valores numéricos de 1 al 5 se almacenan en las memorias matriciales A[1] hasta A[5], mientras que la memoria C se utiliza como contador. Al ejecutarse este programa, se obtiene el siguiente resultado:

Prog

0

EXE

EXE

EXE

EXE

EXE

1.
0.
3.
4.
5.

Como podrá observarse, el segundo valor visualizado correspondiente a A[2] debería ser igual a 2 pero no lo es. Esto ha sucedido ya que se han sobrepuesto la memoria A[2] y la memoria C.

A	B	C	D	E	F
	A[1]	A[2]	A[3]	A[4]	A[5]

El contenido de la memoria C (igual que A[2]) va decreciendo de 5 hasta 0 por unidad. Es por ello que el contenido de la memoria A[2] se visualiza como 0.

■Aplicaciones de las memorias matriciales

A veces se requiere el tratamiento de dos tipos diferentes de datos a modo de un único grupo. En este caso, las memorias para procesamiento de datos y las memorias para almacenamiento de datos deben mantenerse separado.

Ejemplo Almacenamiento del dato x e y en memorias. Al entrarse el valor x, se visualiza el valor y. Se trabaja con un total de 15 datos.

Ejemplo de programa 1

La memoria A se utiliza para controlar lo datos y la memoria B para el almacenamiento temporal del dato x. Los datos x se almacenan en las memorias C[1] (memoria D) hasta C[15] (memoria R), mientras que los datos y se almacenan en las memorias C[16] (memoria S) hasta C[30] (memoria Z(7)).

```
1, →, A, :, Defm, 7, :,  
Lbl, 1, :, ?, →, C, [, A, ], :,  
?, →, C, [, A, +, 1, 5, ], :,  
Isz, A, :, A, =, 1, 6, ⇒, Goto, 2, :, Goto, 1, :,  
Lbl, 2, :, 1, 5, →, A, :, ?, →, B, :,  
B, =, 0, ⇒, Goto, 5, :,  
Lbl, 3, :, B, =, C, [, A, ], ⇒, Goto, 4, :,  
Dsz, A, :, Goto, 3, :, Goto, 2, :,  
Lbl, 4, :, C, [, A, +, 1, 5, ], ▲, Goto, 2, :,  
Lbl, 5
```

98 pasos

En este programa, las memorias se utilizan del siguiente modo:

datos x

C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]
D	E	F	G	H	I	J	K
C[9]	C[10]	C[11]	C[12]	C[13]	C[14]	C[15]	
L	M	N	O	P	Q	R	

datos y

C[16]	C[17]	C[18]	C[19]	C[20]	C[21]	C[22]	C[23]
S	T	U	V	W	X	Y	Z
C[24]	C[25]	C[26]	C[27]	C[28]	C[29]	C[30]	
Z(1)	Z(2)	Z(3)	Z(4)	Z(5)	Z(6)	Z(7)	

Ejemplo de programa 2

Se utilizan las mismas memorias que en el ejemplo 1, sólo que se utilizan dos nombres de memoria para mantener separados los datos x e y.

```
1, →, A, :, Defm, 7, :,  
Lbl, 1, :, ?, →, C, [, A, ], :,  
?, →, R, [, A, ], :,  
Isz, A, :, A, =, 1, 6, ⇒, Goto, 2, :, Goto, 1, :,  
Lbl, 2, :, 1, 5, →, A, :, ?, →, B, :,  
B, =, 0, ⇒, Goto, 5, :,  
Lbl, 3, :, B, =, C, [, A, ], ⇒, Goto, 4, :,  
Dsz, A, :, Goto, 3, :, Goto, 2, :,  
Lbl, 4, :, R, [, A, ], ▲, Goto, 2, :,  
Lbl, 5
```

92 pasos

Las memorias se utilizan del siguiente modo:

datos *x*

C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]
D	E	F	G	H	I	J	K
C[9]	C[10]	C[11]	C[12]	C[13]	C[14]	C[15]	
L	M	N	O	P	Q	R	

datos *y*

R[1]	R[2]	R[3]	R[4]	R[5]	R[6]	R[7]	R[8]
S	T	U	V	W	X	Y	Z
R[9]	R[10]	R[11]	R[12]	R[13]	R[14]	R[15]	
Z(1)	Z(2)	Z(3)	Z(4)	Z(5)	Z(6)	Z(7)	

De este modo, se pueden cambiar los nombres de las memorias. Sin embargo, como estos nombres se limitan a las letras A hasta la Z, las memorias adicionales (**MODE** **•**) pueden usarse solamente como memorias matriciales.

*El mando para agregar memorias (*Defm*) puede usarse dentro de los programas.

Ejemplo Agregado de 14 memorias, para un total de 40 disponibles.

Defm, 1, 4, :,

4-9 Visualización de caracteres alfanuméricos y símbolos

Esta unidad permite la visualización de caracteres alfabéticos, números, símbolos de mandos y otros a modo de mensajes. Deben entrarse entre comillas (**ALPHA** **99**).

■ Caracteres alfanuméricos y símbolos

● Caracteres y símbolos que se visualizan cuando se presiona la tecla después de haber presionado **ALPHA**:

[,], k, m, μ, n, p, f, space (espacio),
A, B, C, D, E, F, G, H, I, J, K, L, M, N,
O, P, Q, R, S, T, U, V, W, X, Y, Z

● Otros números, símbolos, mandos de cálculo y mandos de programa

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
(,), √, E, +, -, ×, ÷, ...
sin, cos, tan, log, ln, ...
=, ≠, ≥, ≤, >, <, ...
A, IB, C, D, E, F, d, h, b, o
Neg, Not, and, or, xor
 \bar{x} , \bar{y} , $x\sigma_n$, $x\sigma_{n-1}$, ...
° (**SHIFT** **MODE** **4**), ° (**SHIFT** **MODE** **5**), ° (**SHIFT** **MODE** **6**)

*Todos los caracteres citados arriba pueden usarse del mismo modo que los caracteres alfabéticos.

En el ejemplo anterior, cuando se solicita la entrada de los datos *x* e *y*, sólo aparece la indicación "?", sin mensaje alguno sobre la información que debe entrarse. Es posible, no obstante, incluir un mensaje antes del símbolo "?" para que quien use el programa sepa qué información debe entrar en cada caso.

Lbl, 1, :, ?, →, X, :, ?, →, Y, :, ...

Agreguemos en este programa los mensajes "X=" e "Y=".

Lbl, 1, :, ", X, =, ", ?, →, X, :,
", Y, =, ", ?, →, Y, :, ...

Al agregar los mensajes mencionados, la pantalla visualizará lo siguiente (suponiendo que el programa está almacenado en P1):

Prog1EXE

10EXE

:

X=?

Y=?

:

Los mensajes son también convenientes cuando se visualizan resultados en cálculos por programa.

Ejemplo

```
Lbl, 0, :, ", N, =, ", ?, →, B, ~, C, :,  
0, →, A, :,  
Lbl, 1, :, C, ÷, 2, →, C, :, Frac, C, ≠, 0, ⇒, Goto, 3,  
:, Isz, A, :, C, =, 1, ⇒, Goto, 2, :, Goto, 1, :,  
Lbl, 2, :, ", X, =, ", ▲, A, ▲, Goto, 0, :,  
Lbl, 3, :, ", N, O, ", ▲, Goto, 0
```

70 pasos

Este programa calcula el valor x elevado al cuadrado. El mensaje “N=?” aparece al solicitarse la entrada del dato. El resultado se visualiza pulsando [EXE] mientras se encuentra la indicación “X=” en la pantalla. Cuando se entra un dato que no pueda elevarse al cuadrado, aparece el mensaje “NO” y la ejecución vuelve al comienzo para repetir la entrada del dato.

**Siempre a continuación de un mensaje con un ▲ cuando después de la fórmula haya un mensaje.*

Suponiendo que el programa está almacenado en P2:

Prog2EXE

4096EXE

EXE

EXE

3124EXE

EXE

512EXE

EXE

N=?

X=

12.

N=?

NO

N=?

X=

9.

La presentación es capaz de mostrar hasta 12 caracteres alfanuméricos al mismo tiempo. Para los mensajes con una longitud mayor de 12 caracteres, utilice [Disp] para separar el mensaje.

4-10 Uso de la función para gráficos en programas

El uso de la función para gráficos en programas hace posible representar gráficamente ecuaciones complejas y sobreponer numerosos gráficos en una misma pantalla. En un programa se pueden incluir todos los mandos gráficos (excepto el de restreo). En ellos se puede, además escribir los valores límites. Por lo general, las operaciones para gráficos manuales pueden usarse directamente en programas, sin modificación alguna.

Ejemplo 1 Determinación gráfica del número de soluciones (raíces reales) que satisfacen las dos ecuaciones siguientes.

$y = x^4 - x^3 - 24x^2 + 4x + 80$
 $y = 10x - 30$

Los valores de los límites son los siguientes:

Xmin : - 10	Ymin : - 120
Xmax: 10	Ymax: 150
Xscl : 2	Yscl : 50

Primeramente, programe los límites. Tenga en cuenta que los valores se separan por medio de comas “,”.

```
Range, (-), 1, 0, , 1, 0, , 2, , (-), 1, 2, 0, , 1, 5, 0, , 5, 0
```

A continuación, programe la ecuación para el primer gráfico.

```
Graph, X, x^, 4, -, X, x^, 3, -, 2, 4, X, x^2, +, 4, X, +, 8, 0
```

Por último, programe la ecuación para el segundo gráfico.

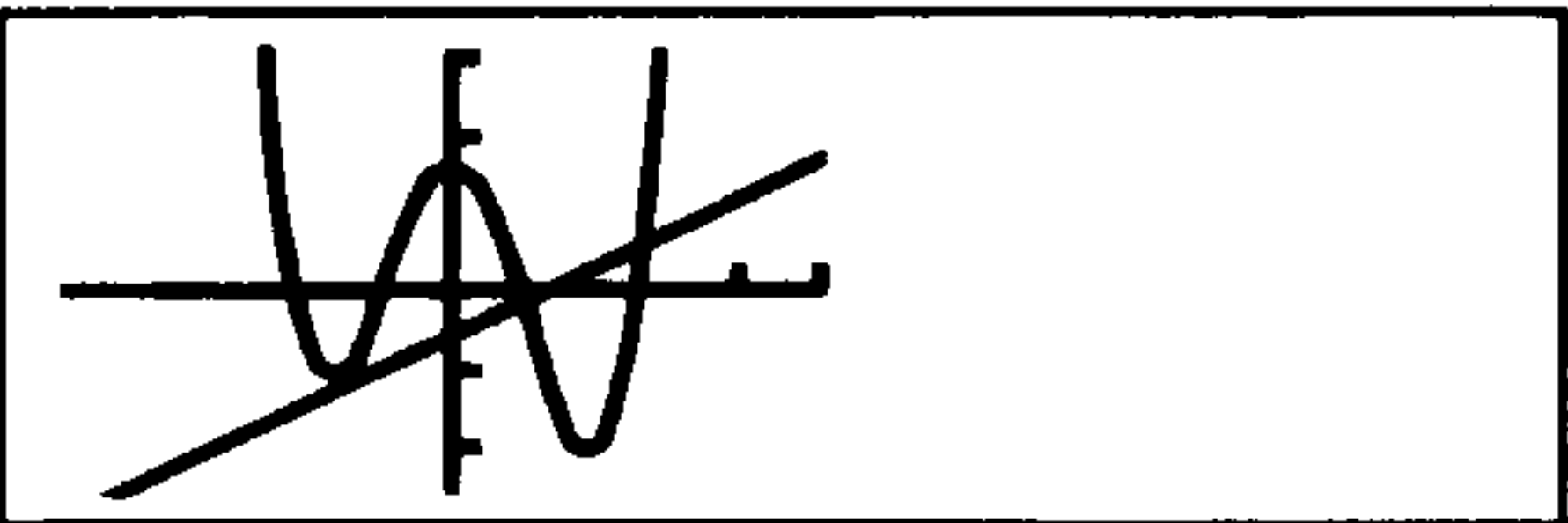
```
Graph, 1, 0, X, -, 3, 0
```

49 pasos

Después de la entrada de los límites y de la primera ecuación, recuerde pulsar la tecla [EXE].

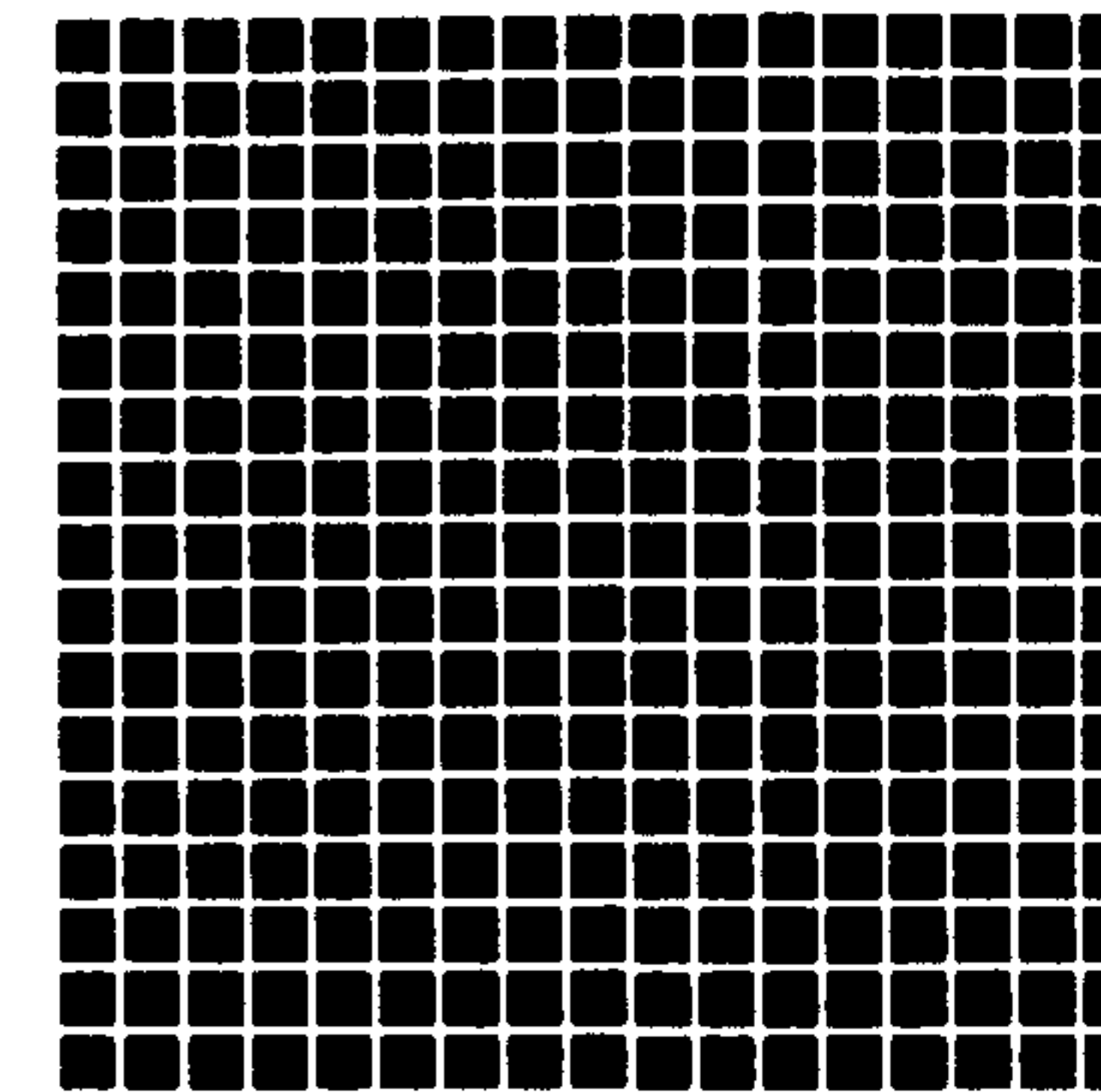
Al ejecutarse el programa, en la pantalla debe aparecer lo siguiente:

```
Prog0EXE
```



Después de la primera ecuación se puede entrar “▲”, a fin de suspender la ejecución después de trazado el primer gráfico. Para continuar con la ejecución (el segundo gráfico), pulse la tecla [EXE].

El procedimiento que se acaba de describir sirve para el trazado de una amplia variedad de gráficos. La programoteca de este manual, da algunos ejemplos sobre el trazado de gráficos por medio de programas.



Programoteca

Análisis de factor primo

Enteros definidos usando la regla de Simpson

Conversión $\Delta \leftrightarrow Y$

Igualación de impedancias con pérdidas mínimas

Viga voladiza bajo carga concentrada

Distribución normal

Variación de gráficos por parámetros

Ciclo de histéresis

Curva de regresión

Diagrama "parade"

< Antes del uso >

- Antes de almacenar programas, verifique sin falta el número remanente de pasos de programa.
- La programoteca se divide en dos partes: una para cálculos y otra para gráficos. La primera muestra solamente las respuestas, mientras que la segunda muestra todo lo que en realidad se visualiza en la pantalla.
- Siempre que en un programa aparezca "Graph" ("Graph Y = " en pantalla), púlsese la tecla **Graph**.
- Si en un programa es necesario especificar el modo de cálculo (por ej., BASE-N o SD1), asegúrese de hacerlo después de pulsar **MODE** **2** (modo WRT). Luego, comience a programar pulsando la tecla **EXE**.

CASIO HOJA DE PROGRAMA

Programa para		Análisis de factor primo		No. 1																																																																			
<div>Descripción</div> <p>Se producen factores primos de enteros positivos arbitrarios.</p> <p>Para $1 < m < 10^{10}$</p> <p>los números primos se producen a partir del primer valor más bajo. El mensaje "END" aparece al final del programa.</p> <p>< Planteo ></p> <p>m se divide por 2 y todos los números impares que le siguen ($d = 3, 5, 7, 9, 11$, etc.) para ver si son divisibles.</p> <p>Cuando d es un factor primo, se supone que $m_i = m_{i-1}/d$, y la división se repite hasta que $\sqrt{m_i} + 1 \leq d$.</p> <div>Ejemplo</div> <p><1></p> <p>$119 = 7 \times 17$</p> <p><2></p> <p>$1234567890 = 2 \times 3 \times 3 \times 5 \times 3607 \times 3803$</p> <p><3></p> <p>$987654321 = 3 \times 3 \times 17 \times 17 \times 379721$</p> <div>Preparativos y procedimiento</div> <ul style="list-style-type: none">●Almacene el programa de la página siguiente.●Ejécútelo del modo descrito a continuación en el modo RUN (MODE 1). <table><tr><th>Paso</th><th>Entrada desde el teclado</th><th>Pantalla</th><th>Paso</th><th>Entrada desde el teclado</th><th>Pantalla</th></tr><tr><td>1</td><td>Prog 0 EXE</td><td>M?</td><td>11</td><td>EXE</td><td>3803.</td></tr><tr><td>2</td><td>119 EXE</td><td>7.</td><td>12</td><td>EXE</td><td>END</td></tr><tr><td>3</td><td>EXE</td><td>17.</td><td>13</td><td>EXE</td><td>M?</td></tr><tr><td>4</td><td>EXE</td><td>END</td><td>14</td><td>987654321 EXE</td><td>3.</td></tr><tr><td>5</td><td>EXE</td><td>M?</td><td>15</td><td>EXE</td><td>3.</td></tr><tr><td>6</td><td>1234567890 EXE</td><td>2.</td><td>16</td><td>EXE</td><td>17.</td></tr><tr><td>7</td><td>EXE</td><td>3.</td><td>17</td><td>EXE</td><td>17.</td></tr><tr><td>8</td><td>EXE</td><td>3.</td><td>18</td><td>EXE</td><td>(15 segundos después) 379721.</td></tr><tr><td>9</td><td>EXE</td><td>5.</td><td>19</td><td>EXE</td><td>END</td></tr><tr><td>10</td><td>EXE</td><td>(80 segundos después) 3607.</td><td>20</td><td></td><td></td></tr></table>						Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla	1	Prog 0 EXE	M?	11	EXE	3803.	2	119 EXE	7.	12	EXE	END	3	EXE	17.	13	EXE	M?	4	EXE	END	14	987654321 EXE	3.	5	EXE	M?	15	EXE	3.	6	1234567890 EXE	2.	16	EXE	17.	7	EXE	3.	17	EXE	17.	8	EXE	3.	18	EXE	(15 segundos después) 379721.	9	EXE	5.	19	EXE	END	10	EXE	(80 segundos después) 3607.	20		
Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla																																																																		
1	Prog 0 EXE	M?	11	EXE	3803.																																																																		
2	119 EXE	7.	12	EXE	END																																																																		
3	EXE	17.	13	EXE	M?																																																																		
4	EXE	END	14	987654321 EXE	3.																																																																		
5	EXE	M?	15	EXE	3.																																																																		
6	1234567890 EXE	2.	16	EXE	17.																																																																		
7	EXE	3.	17	EXE	17.																																																																		
8	EXE	3.	18	EXE	(15 segundos después) 379721.																																																																		
9	EXE	5.	19	EXE	END																																																																		
10	EXE	(80 segundos después) 3607.	20																																																																				

No. 1													Notas	Número de pasos
Línea	MODE 2	Programa												
1	Mcl	:												2
2	Lbl	0	:	"	M	"	?	→	A	:	Goto	2	:	15
3	Lbl	1	:	2	▲	A	÷	2	→	A	:	A	=	1 ⇒ 30
4	Goto	9	:											33
5	Lbl	2	:	Frac	(A	÷	2)	=	0	⇒	Goto	1 : 48
6	3	→	B	:										52
7	Lbl	3	:	√	A	+	1	→	C	:				62
8	Lbl	4	:	B	≥	C	⇒	Goto	8	:	Frac	(A	÷ B 77
9)	=	0	⇒	Goto	6	:							84
10	Lbl	5	:	B	+	2	→	B	:	Goto	4	:		96
11	Lbl	6	:	A	÷	B	×	B	-	A	=	0	⇒	Goto 7 111
12	:	Goto	5	:										115
13	Lbl	7	:	B	▲	A	÷	B	→	A	:	Goto	3	: 129
14	Lbl	8	:	A	▲									134
15	Lbl	9	:	"	E	N	D	"	▲	Goto	0			145
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
Contenido de las memorias	A	m_i			H				O				V	
	B	d			I				P				W	
	C	$\sqrt{m_i} + 1$			J				Q				X	
	D				K				R				Y	
	E				L				S				Z	
	F				M				T					
	G				N				U					

CASIO HOJA DE PROGRAMA

Programa para

Enteros definidos usando la regla de Simpson

No.

2

Descripción

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

$$h = \frac{b-a}{2m}$$

La porción de la derecha en la ecuación de arriba puede transformarse del siguiente modo.

$$I = \frac{h}{3} \{ y_0 + \sum_{i=1}^m (4y_{2i-1} + 2y_{2i}) - y_{2m} \}$$

$$\text{Sea } f(x) = \frac{1}{x^2 + 1}$$

Ejemplo

<1> $a=0, b=1, 2m=10$

$$I = \int_0^1 \frac{1}{x^2 + 1} dx = 0,7853981537$$

<2> $a=2, b=5, 2m=20$

$$I = \int_2^5 \frac{1}{x^2 + 1} dx = 0,2662526769$$

Preparativos y procedimiento

- Almacene el programa de la página siguiente.
- Ejecútelo del modo descrito a continuación en el modo RUN (**MODE** **1**).

Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla
1	Prog 0 EXE	A?	11		
2	0 EXE	B?	12		
3	1 EXE	2M?	13		
4	10 EXE	0.7853981535	14		
5	EXE	A?	15		
6	2 EXE	B?	16		
7	5 EXE	2M?	17		
8	20 EXE	0.2662526768	18		
9			19		
10			20		

No.

2

Línea	MODE 2 Programa															Notas	Número de pasos
1	P0																
2	Lbl	1	:	Mcl	:												5
3	"	A	"	?	→	A	:	"	B	"	?	→	B	:	"		20
4	2	M	"	?	→	M	:										27
5	A	→	G	:	Prog	1	:	P	→	I	:	(B	-	A		42
6)	÷	M	→	D	:	M	÷	2	→	O	:					54
7	Lbl	2	:	G	+	D	→	G	:	Prog	1	:	I	+	P		69
8	×	4	→	I	:												74
9	G	+	D	→	G	:	Prog	1	:	I	+	P	×	2	→		89
10	I	:	O	-	1	→	O	:									97
11	O	≠	0	⇒	Goto	2	:										104
12	B	→	G	:	Prog	1	:	I	-	P	→	I	:				117
13	D	×	I	÷	3	▲											123
14	Goto	1															125
15																	
16	P1																
17	1	÷	(G	×	G	+	1)	→	P						11
18																	
19																Total 136 pasos	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
Contenido de las memorias	A	a				H					O	m (número de repeticiones)				V	
	B	b				I	l				P					W	
	C					J					Q					X	
	D	$h = \frac{b-a}{2m}$				K					R					Y	
	E					L					S					Z	
	F					M	2m				T						
	G	x				N					U						

CASIO HOJA DE PROGRAMA

Programa para		Conversión $\Delta \leftrightarrow Y$		No. 3	
Descripción					
				\Rightarrow \Leftarrow	
1) $\Delta \rightarrow Y$		2) $Y \rightarrow \Delta$			
$R_4 = \frac{R_1 \cdot R_2}{R_1 + R_2 + R_3}$ $R_5 = \frac{R_2 \cdot R_3}{R_1 + R_2 + R_3}$ $R_6 = \frac{R_3 \cdot R_1}{R_1 + R_2 + R_3}$		$R_1 = \frac{R_4 R_5 + R_5 R_6 + R_6 R_4}{R_5}$ $R_2 = \frac{R_4 R_5 + R_5 R_6 + R_6 R_4}{R_6}$ $R_3 = \frac{R_4 R_5 + R_5 R_6 + R_6 R_4}{R_4}$			
Ejemplo					
<1>		<2>			
$R_1 = 12 (\Omega)$		$R_4 = 100 (\Omega)$			
$R_2 = 47 (\Omega)$		$R_5 = 150 (\Omega)$			
$R_3 = 82 (\Omega)$		$R_6 = 220 (\Omega)$			
Preparativos y procedimiento					
● Almacene el programa de la página siguiente.					
● Ejecútelo del modo descrito a continuación en el modo RUN (MODE 1).					
Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla
1	Prog 0 EXE	D→Y:1, Y→D:2?	11	EXE	D→Y:1, Y→D:2?
2	1 EXE	R1 = ?	12	2 EXE	R4 = ?
3	12 EXE	R2 = ?	13	100 EXE	R5 = ?
4	47 EXE	R3 = ?	14	150 EXE	R6 = ?
5	82 EXE	R4 =	15	220 EXE	R1 =
6	EXE	4.	16	EXE	466.6666667
7	EXE	R5 =	17	EXE	R2 =
8	EXE	27.33333333	18	EXE	318.1818182
9	EXE	R6 =	19	EXE	R3 =
10	EXE	6.978723404	20	EXE	700.

													No. 3				
Línea	MODE 2 Programa													Notas	Número de pasos		
1	Lbl	1	:	"	D	→	Y	:	1	,	Y	→	D	:	2		15
2	"	?	→	N	:												20
3	N	=	2	⇒	Goto	2	:	N	≠	1	⇒	Goto	1	:			34
4	"	R	1	=	"	?	→	A	:								43
5	"	R	2	=	"	?	→	B	:								52
6	"	R	3	=	"	?	→	C	:								61
7	A	+	B	+	C	→	D	:									69
8	"	R	4	=	"	▲	A	×	B	÷	D	▲					81
9	"	R	5	=	"	▲	B	×	C	÷	D	▲					93
10	"	R	6	=	"	▲	A	×	C	÷	D	▲					105
11	Goto	1	:														108
12	Lbl	2	:														111
13	"	R	4	=	"	?	→	E	:								120
14	"	R	5	=	"	?	→	F	:								129
15	"	R	6	=	"	?	→	G	:								138
16	E	×	F	+	F	×	G	+	G	×	E	→	H	:			152
17	"	R	1	=	"	▲	H	÷	F	▲							162
18	"	R	2	=	"	▲	H	÷	G	▲							172
19	"	R	3	=	"	▲	H	÷	E	▲							182
20	Goto	1															184
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
Contenido de las memorias	A	R ₁			H	R ₄ R ₅ + R ₅ R ₆ + R ₆ R ₄			O				V				
	B	R ₂			I				P				W				
	C	R ₃			J				Q				X				
	D	R ₁ + R ₂ + R ₃			K				R				Y				
	E	R ₄			L				S				Z				
	F	R ₅			M				T								
	G	R ₆			N	Para juicio			U								

CASIO HOJA DE PROGRAMA

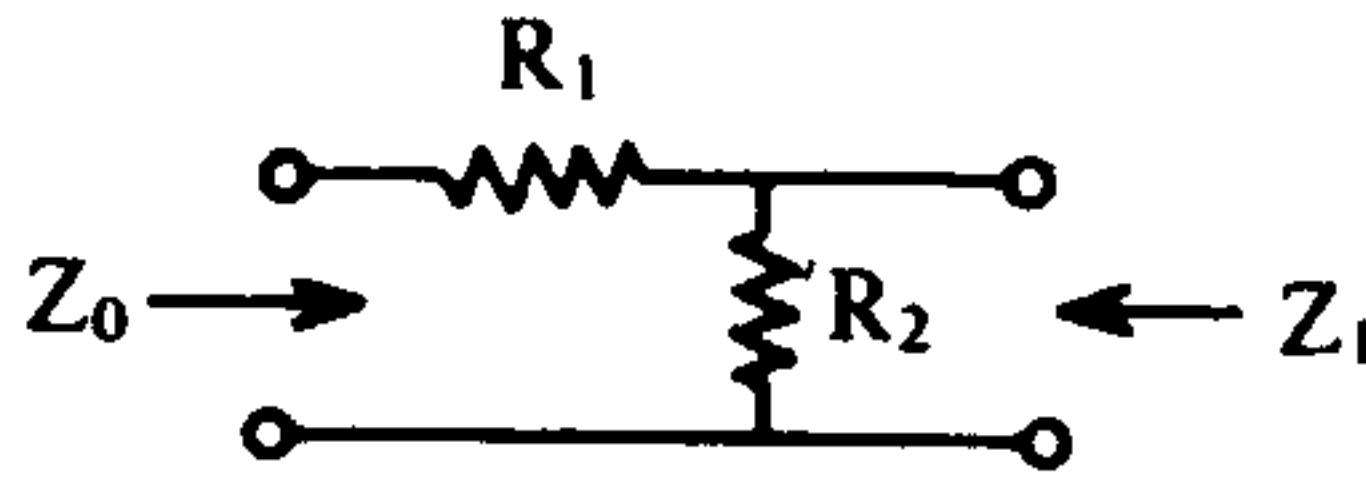
Programa para

Igualación de impedancias con pérdidas mínimas

No. 4

Descripción

Calcular R_1 y R_2 para que igualen a Z_0 y Z_1 un mínimo de pérdidas ($Z_0 > Z_1$)



$$R_1 = Z_0 \sqrt{1 - \frac{Z_1}{Z_0}}$$
$$R_2 = \frac{Z_1}{\sqrt{1 - \frac{Z_1}{Z_0}}}$$

$$\text{Pérdida mínima } L_{\min} = 20 \log \left(\sqrt{\frac{Z_0}{Z_1}} + \sqrt{\frac{Z_0}{Z_1} - 1} \right) \text{ [dB]}$$

Ejemplo

Calcule los valores de R_1 , R_2 y L_{\min} para $Z_0 = 500$ ohmios y $Z_1 = 200$ ohmios.

Preparativos y procedimiento

- Almacene el programa de la página siguiente.
- Ejécútelo del modo descrito a continuación en el modo RUN (MODE 1).

Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla
1	Prog 0 EXE	Z0 = ?	11		
2	500 EXE	Z1 = ?	12		
3	200 EXE	R1 =	13		
4	EXE	387.2983346	14		
5	EXE	R2 =	15		
6	EXE	258.1988897	16		
7	EXE	LMIN =	17		
8	EXE	8.961393328	18		
9			19		
10			20		

- 146 -

No. 4

Línea	MODE 2	Programa	Notas	Número de pasos				
1	" Z 0 = " ? → Y :			9				
2	" Z 1 = " ? → Z :			18				
3	√ (1 - Z ÷ Y) → A :			29				
4	Y × A → R : Z ÷ A → S : Y ÷ Z			44				
5	→ B : 2 0 × log (√ B + √ (B -			59				
6	1)) → T :			65				
7	" R 1 = " ▲ R ▲			73				
8	" R 2 = " ▲ S ▲			81				
9	" L M I N = " ▲ T			90				
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
Contenido de las memorias	A	√ 1 - Z1 / Z0	H		O		V	
	B	Z0 / Z1	I		P		W	
	C		J		Q		X	
	D		K		R	R1	Y	Z0
	E		L		S	R2	Z	Z1
	F		M		T	Lmin		
	G		N		U			

- 147 -

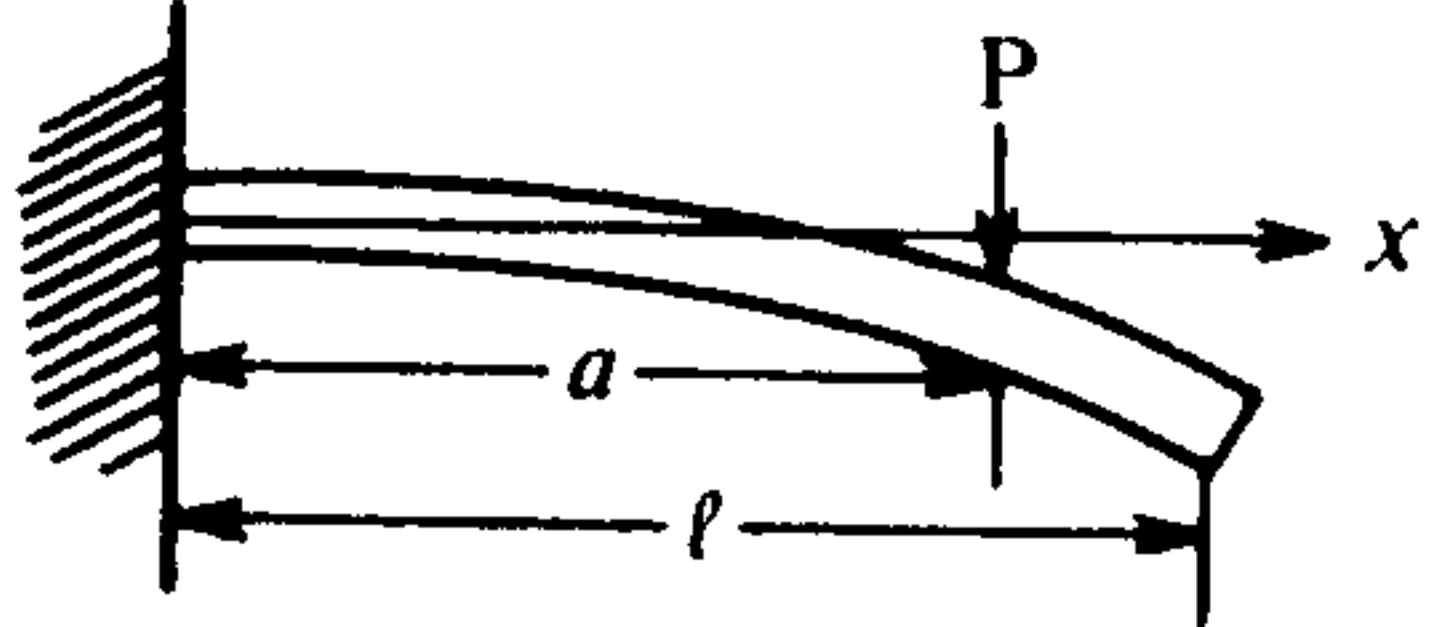
CASIO HOJA DE PROGRAMA

Programa para

Viga voladiza bajo carga concentrada

No.5

Descripción



E : Módulo de Young [kg/mm²]
I : Momento de inercia geométrica [mm⁴]
a : Distancia de la carga concentrada desde el apoyo [mm]
P : Carga [kg]
x : Distancia del punto de interés desde el apoyo [mm]

Deflexión y [mm], ángulo de deflexión s [°], y momento de torsión M [kg·mm]

① l > x > a

$$y = \frac{Pa^3}{6EI} - \frac{Pa^2}{2EI}x$$
$$s = \tan^{-1} \left[-\frac{Pa^2}{2EI} \right]$$

M = 0 (Carga tangencial Ws = 0)

② x ≤ a

$$y = \frac{P}{6EI}x^3 - \frac{Pa}{2EI}x^2$$
$$s = \tan^{-1} \left[\frac{Px}{2EI}(x - 2a) \right]$$

M = P(x - a) (Carga tangencial Ws = P)

Ejemplo

E = 4000 kg/mm²
I = 5 mm⁴
a = 30 mm
P = 2 kg

Averigue la deflexión, el ángulo de deflexión, el momento de torsión y la carga tangencial para x = 25 mm y x = 32 mm.

Preparativos y procedimiento

- Almacene el programa de la página siguiente.
- Ejecútelo del modo descrito a continuación en el modo RUN (MODE 1).

Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla
1	Prog 0 EXE	E = ?	11	EXE	- 10.
2	4000 EXE	I = ?	12	EXE	X = ?
3	5 EXE	A = ?	13	32 EXE	Y =
4	30 EXE	P = ?	14	EXE	- 0.99
5	2 EXE	X = ?	15	EXE	S =
6	25 EXE	Y =	16	EXE	- 2.57657183
7	EXE	- 0.677083333	17	EXE	M =
8	EXE	S =	18	EXE	0.
9	EXE	- 2.505092867	19	Repita desde el paso 5.	
10	EXE	M =	20		

- 148 -

No.5

Linea

MODE 2

Programa

Notas

Número de pasos

1	Deg	:	"	E	=	"	?	→	E	:	"	I	=	"	?		15
2	→	I	:	"	A	=	"	?	→	A	:	"	P	=	"		30
3	?	→	P	:													34
4	Lbl	1	:	"	X	=	"	?	→	X	:						45
5	X	≤	A	⇒	Goto	2	:										52
6	"	Y	=	"	▲	P	×	A	x ²	÷	(2	×	E	×		67
7	I)	×	(A	÷	3	-	X)	▲						78
8	"	S	=	"	▲	tan ⁻¹	((-)	P	×	A	x ²	÷	(2		93
9	×	E	×	I))	▲	"	M	=	"	▲	0	▲			107
10	Goto	1	:														110
11	Lbl	2	:														113
12	"	Y	=	"	▲	P	×	X	x ²	÷	(2	×	E	×		129
13	I)	×	(X	÷	3	-	A)	▲						139
14	"	S	=	"	▲	tan ⁻¹	(P	×	X	÷	(2	×	E		154
15	×	I)	×	(X	-	2	×	A))	▲				167
16	"	M	=	"	▲	P	×	(X	-	A)	▲				180
17	Goto	1															182
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
Contenido de las memorias	A		a	H				O				V					
	B			I		I		P		P		W					
	C			J				Q				X		x			
	D			K				R				Y					
	E		E	L				S				Z					
	F			M				T									
	G			N				U									

- 149 -

CASIO HOJA DE PROGRAMA

Programa para		Distribución normal		No. 6																																																																			
<div>Descripción</div> <div>Obtenga la función de distribución normal $\phi(x)$ (por medio de la aproximación de Hastings).</div> <div>$\phi(x) = \int_{-\infty}^x \phi(t) dx$$\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$<div>Sea $t = \frac{1}{1 + Px}$</div>$\phi(x) = 1 - \phi(t) (c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5)$<div><div>$P = 0,2316419$$C_1 = 0,31938153$$C_2 = -0,356563782$</div><div>$C_3 = 1,78147937$$C_4 = -1,821255978$$C_5 = 1,330274429$</div></div></div> <div><div>Ejemplo</div><div>Calcule los valores de $\phi(x)$ para $x = 1,18$ y $x = 0,7$.</div></div> <div><div>Preparativos y procedimiento</div><div><div>•Almacene el programa de la página siguiente.</div><div>•Ejécútelo del modo descrito a continuación en el modo RUN (MODE 1).</div></div><table><tr><td>Paso</td><td>Entrada desde el teclado</td><td>Pantalla</td><td>Paso</td><td>Entrada desde el teclado</td><td>Pantalla</td></tr><tr><td>1</td><td><div>Prog 0 EXE</div></td><td>X = ?</td><td>11</td><td></td><td></td></tr><tr><td>2</td><td><div>1.18 EXE</div></td><td>PX =</td><td>12</td><td></td><td></td></tr><tr><td>3</td><td><div>EXE</div></td><td>0.880999696</td><td>13</td><td></td><td></td></tr><tr><td>4</td><td><div>EXE</div></td><td>X = ?</td><td>14</td><td></td><td></td></tr><tr><td>5</td><td><div>0.7 EXE</div></td><td>PX =</td><td>15</td><td></td><td></td></tr><tr><td>6</td><td><div>EXE</div></td><td>0.7580361368</td><td>16</td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td>17</td><td></td><td></td></tr><tr><td>8</td><td></td><td></td><td>18</td><td></td><td></td></tr><tr><td>9</td><td></td><td></td><td>19</td><td></td><td></td></tr><tr><td>10</td><td></td><td></td><td>20</td><td></td><td></td></tr></table></div>						Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla	1	<div>Prog 0 EXE</div>	X = ?	11			2	<div>1.18 EXE</div>	PX =	12			3	<div>EXE</div>	0.880999696	13			4	<div>EXE</div>	X = ?	14			5	<div>0.7 EXE</div>	PX =	15			6	<div>EXE</div>	0.7580361368	16			7			17			8			18			9			19			10			20		
Paso	Entrada desde el teclado	Pantalla	Paso	Entrada desde el teclado	Pantalla																																																																		
1	<div>Prog 0 EXE</div>	X = ?	11																																																																				
2	<div>1.18 EXE</div>	PX =	12																																																																				
3	<div>EXE</div>	0.880999696	13																																																																				
4	<div>EXE</div>	X = ?	14																																																																				
5	<div>0.7 EXE</div>	PX =	15																																																																				
6	<div>EXE</div>	0.7580361368	16																																																																				
7			17																																																																				
8			18																																																																				
9			19																																																																				
10			20																																																																				

														No.	6				
Línea	MODE 2 Programa															Notas	Número de pasos		
1	"	X	=	"	?	→	X	:								8			
2	1	÷	(1	+	0	.	2	3	1	6	4	1	9	×	23			
3	X)	→	T	:	1	÷	√	(2	×	π)	×	e ^x	38			
4	((-)	X	x ²	÷	2)	→	Q	:						48			
5	0	.	3	1	9	3	8	1	5	3	→	A	:			61			
6	(-)	0	.	3	5	6	5	6	3	7	8	2	→	B	:	76			
7	1	.	7	8	1	4	7	9	3	7	→	C	:			89			
8	(-)	1	.	8	2	1	2	5	5	9	7	8	→	D	:	104			
9	1	.	3	3	0	2	7	4	4	2	9	→	E	:		118			
10	"	P	X	=	"	▲	1	-	Q	(A	T	+	B	T	133			
11	x ²	+	C	T	x ^y	3	+	D	T	x ^y	4	+	E	T	x ^y	148			
12	5)														150			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			
Contenido de las memorias	A						H						O				V		
	B						I						P				W		
	C						J						Q	φt			X	x	
	D						K						R				Y		
	E						L						S				Z		
	F						M						T	t					
	G						N						U						

CASIO HOJA DE PROGRAMA

Programa para

Variación de gráficos por parámetros

No.

7

Descripción

Vibración amortiguada

(i) $\epsilon > n$ (sobreamortiguamiento)

$$P_1 = -\epsilon + \sqrt{\epsilon^2 - n^2}, \quad P_2 = -\epsilon - \sqrt{\epsilon^2 - n^2}$$

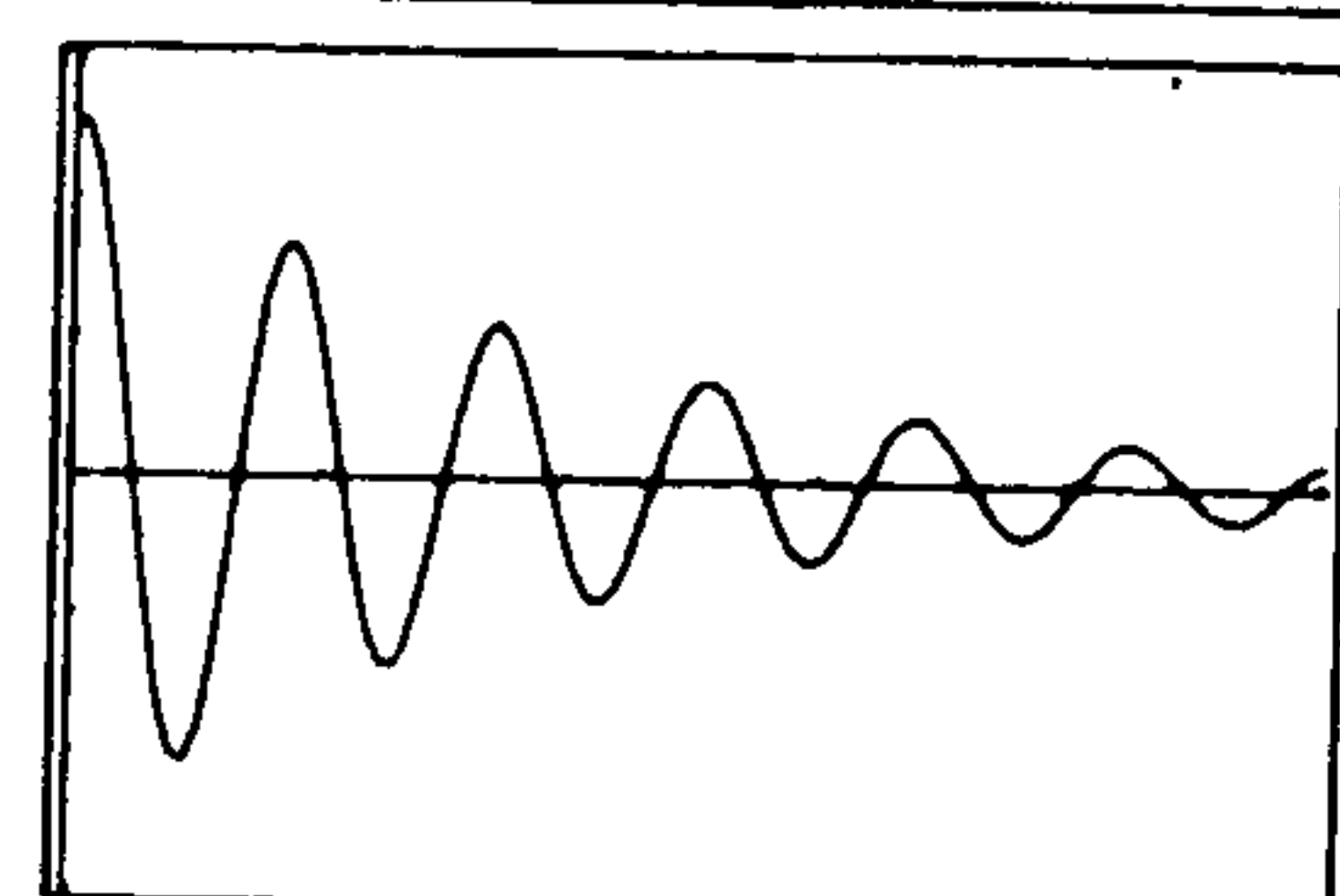
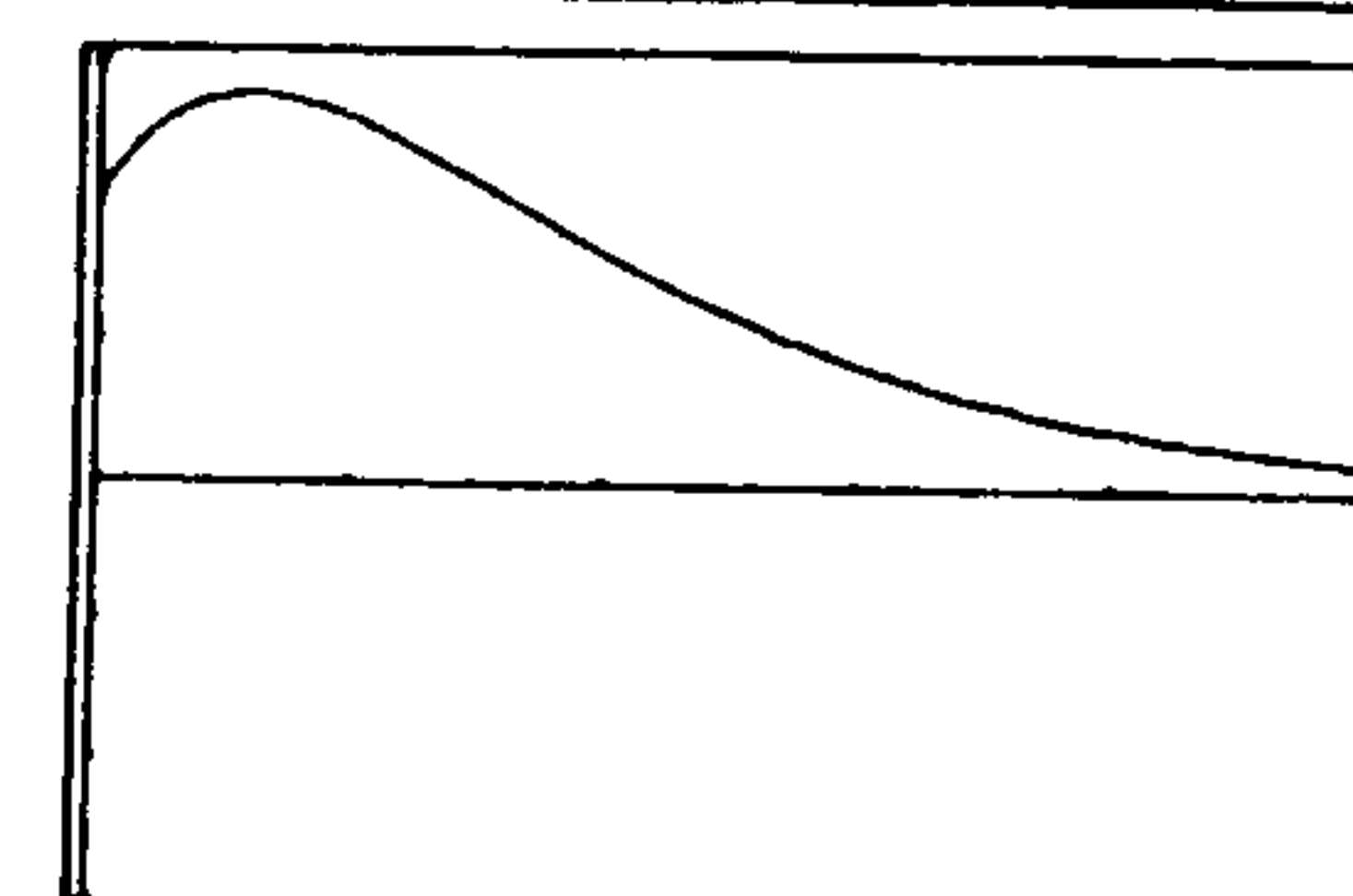
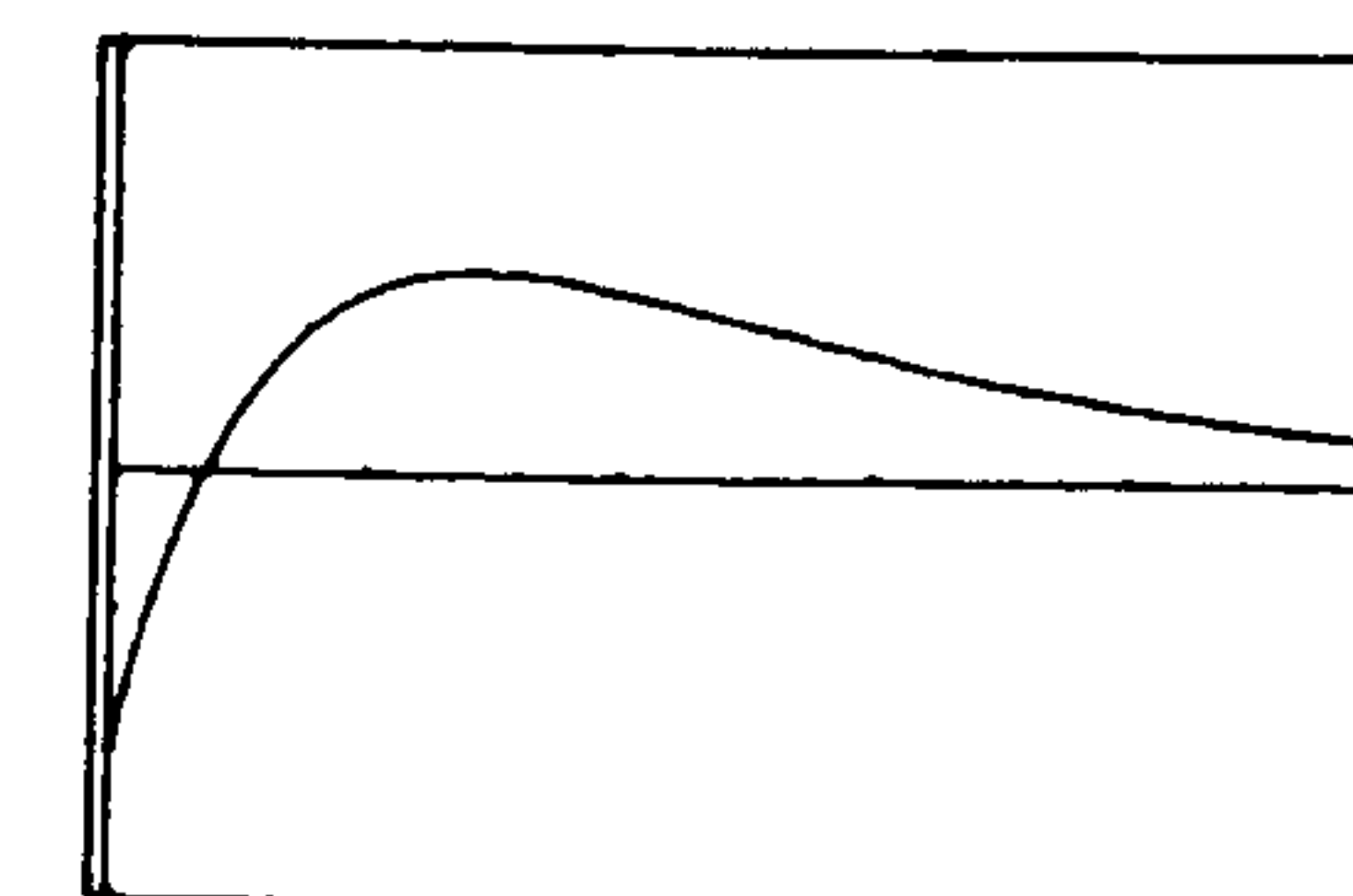
$$x = \frac{v_0 - x_0 P_2}{P_1 - P_2} e^{P_1 t} - \frac{v_0 - x_0 P_1}{P_1 - P_2} e^{P_2 t}$$

(ii) $\epsilon = n$ (amortiguamiento crítico)

$$x = [x_0 + (v_0 + \epsilon x_0)t] e^{-\epsilon t}$$

(iii) $\epsilon < n$ (vibración de amortiguamiento)

$$x = e^{-\epsilon t} \left\{ x_0 \cos \sqrt{n^2 - \epsilon^2} t + \frac{v_0 + \epsilon x_0}{\sqrt{n^2 - \epsilon^2}} \sin \sqrt{n^2 - \epsilon^2} t \right\}$$



Ejemplo

Trazado de un gráfico de la vibración de amortiguamiento con los siguientes parámetros:

(1) $\epsilon = 0,1$	(2) $\epsilon = 0,2$	(3) $\epsilon = 0,2$
$n = 1,5$	$n = 0,2$	$n = 0,18$
$x_0 = 2,5$	$x_0 = 2$	$x_0 = -2$
$v_0 = 1$	$v_0 = 0,6$	$v_0 = 1,5$

Preparativos y procedimiento

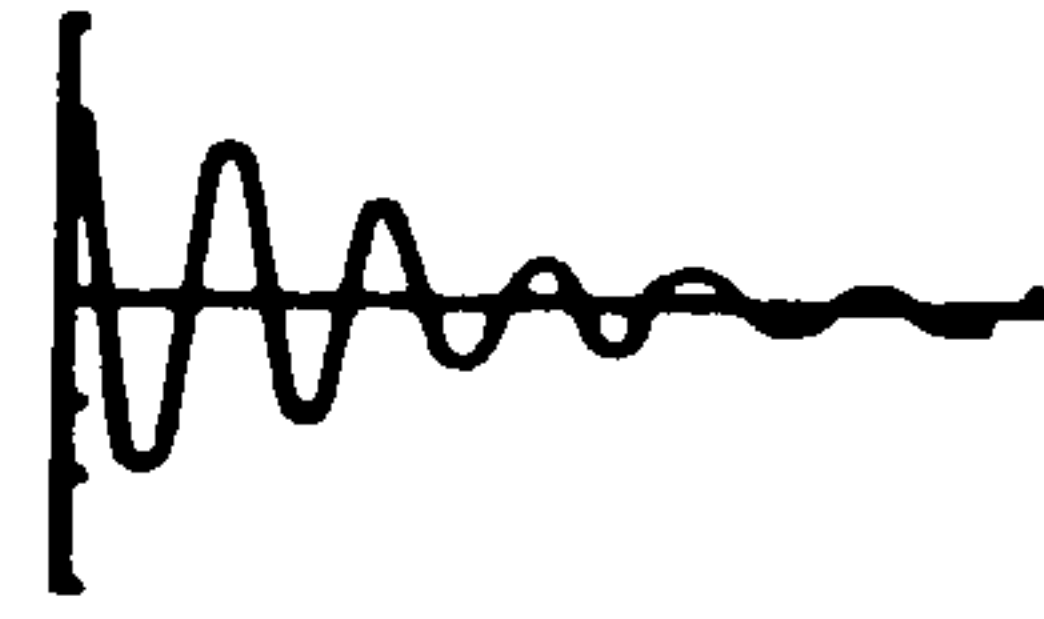
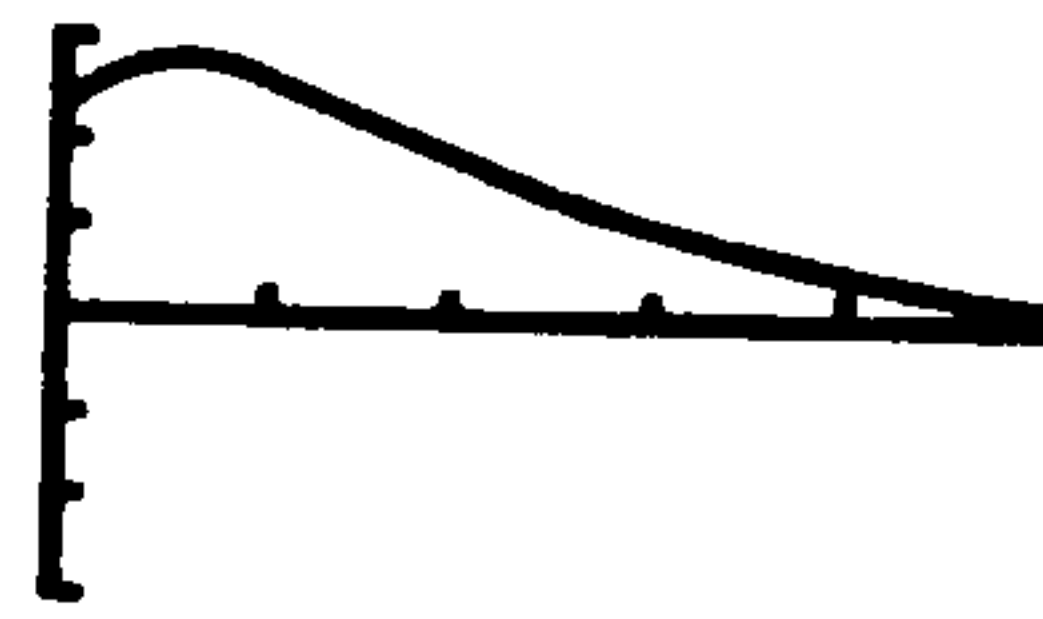
• Almacene el programa de la página siguiente.

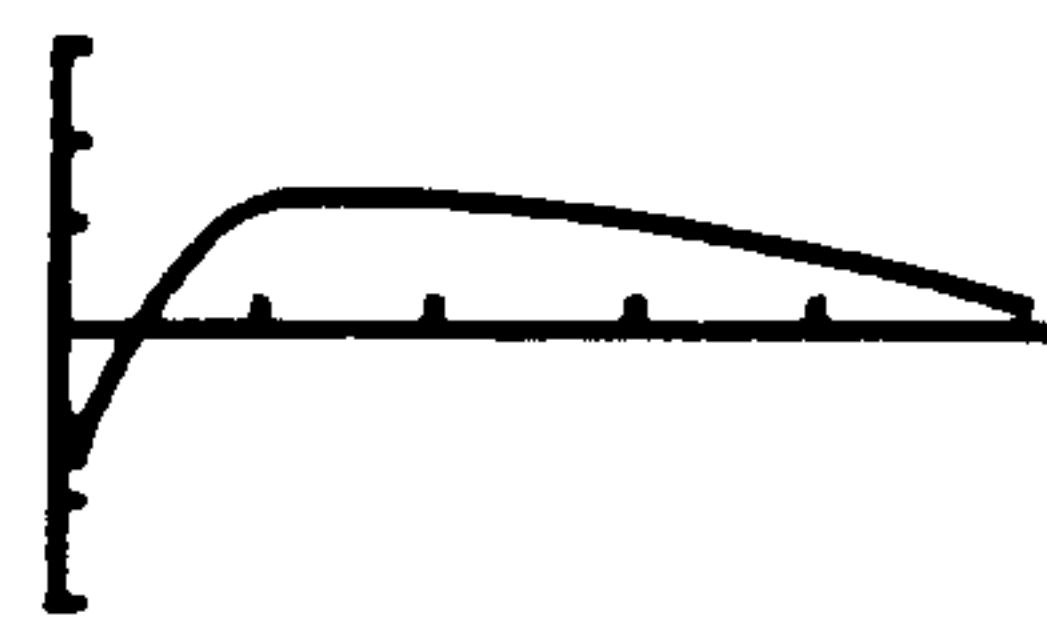
Contenido de las memorias	A	x_0	H		O	V	
	B	v_0	I		P	$P_1 = -\epsilon + \sqrt{\epsilon^2 - n^2}$	W
	C	$\sqrt{n^2 - \epsilon^2}$	J		Q	$P_2 = -\epsilon - \sqrt{\epsilon^2 - n^2}$	X
	D		K		R		Y
	E	ϵ	L		S		Z
	F		M		T		
	G		N	n	U		

No.

7

Línea	MODE	2	Programa											Notas	Número de pasos		
1	Rad	:													2		
2	Range	0	,	2	5	,	5	,	(-)	3	,	3	,	1	:	17	
3	"	E	P	S	I	L	O	N	=	"	?	→	E	:		31	
4	"	N	=	"	?	→	N	:								39	
5	"	X	0	=	"	?	→	A	:							48	
6	"	V	0	=	"	?	→	B	:							57	
7	E	>	N	⇒	Goto	1	:									64	
8	E	=	N	⇒	Goto	2	:									71	
9	√	(N	x ²	-	E	x ²)	→	C	:					82	
10	Graph	e ^x	((-)	E	X)	(A	cos	(C	X)	+		97
11	(B	+	E	A)	C	x ⁻¹	sin	(C	X))	:		112
12	Goto	0	:													115	
13	Lbl	1	:													118	
14	(-)	E	+	√	(E	x ²	-	N	x ²)	→	P	:		132	
15	(-)	E	-	√	(E	x ²	-	N	x ²)	→	Q	:		146	
16	Graph	(B	-	A	Q)	(P	-	Q)	x ⁻¹	e ^x	(161
17	P	X)	-	(B	-	A	P)	(P	-	Q)		176
18	x ⁻¹	e ^x	(Q	X)	:									183	
19	Goto	0	:													186	
20	Lbl	2	:													189	
21	Graph	(A	+	(B	+	E	A)	X)	e ^x	((-)		204
22	E	X)	:												208	
23	Lbl	0														210	
24																	
25															Total 210 pasos		
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	
35																	

Programa para Variación de gráficos por parámetros		No. 7
Paso	Entrada desde el teclado	Pantalla
1	Prog 0 EXE	EPSILON = ?
2	0.1 EXE	N = ?
3	1.5 EXE	X0 = ?
4	2.5 EXE	V0 = ?
5	1 EXE	
6	EXE	EPSILON = ?
7	0.2 EXE	N = ?
8	0.2 EXE	X0 = ?
9	2 EXE	V0 = ?
10	0.6 EXE	

Programa para Variación de gráficos por parámetros		No. 7
Paso	Entrada desde el teclado	Pantalla
11	EXE	EPSILON = ?
12	0.2 EXE	N = ?
13	0.18 EXE	X0 = ?
14	SHIFT (←) 2 EXE	V0 = ?
15	1.5 EXE	
16		
17		
18		
19		
20		

CASIO HOJA DE PROGRAMA

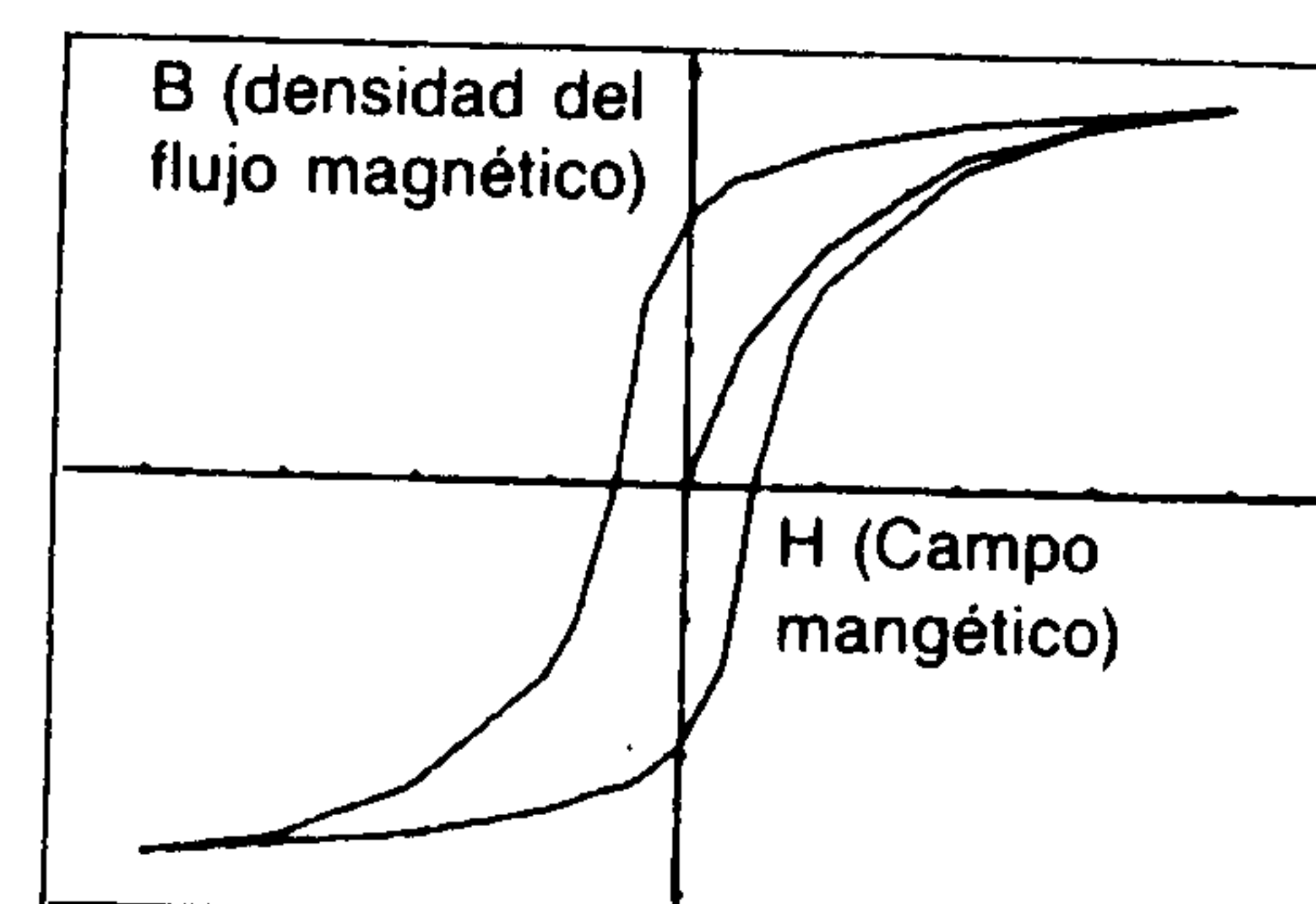
Programa para

Ciclo de histéresis

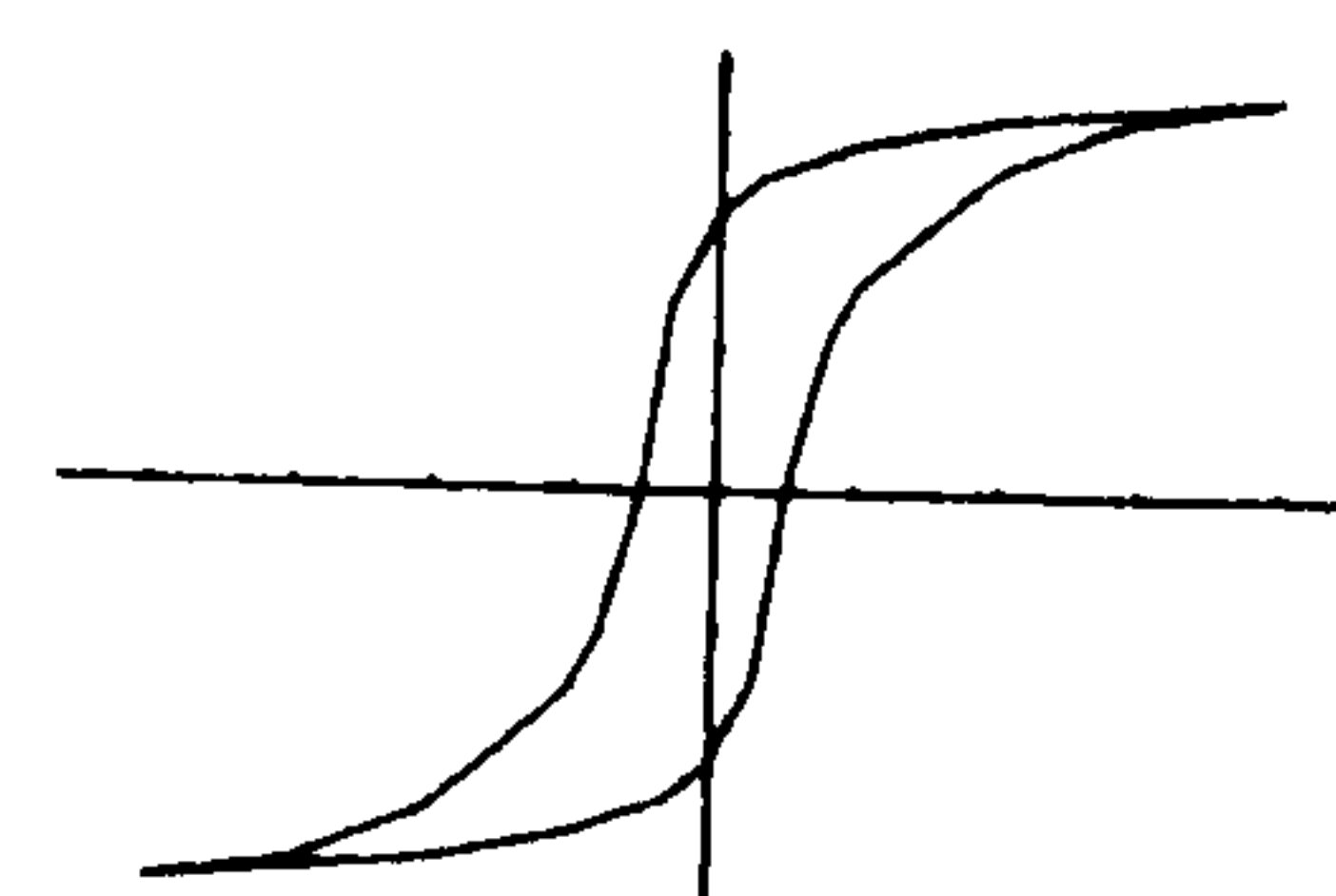
No.

8

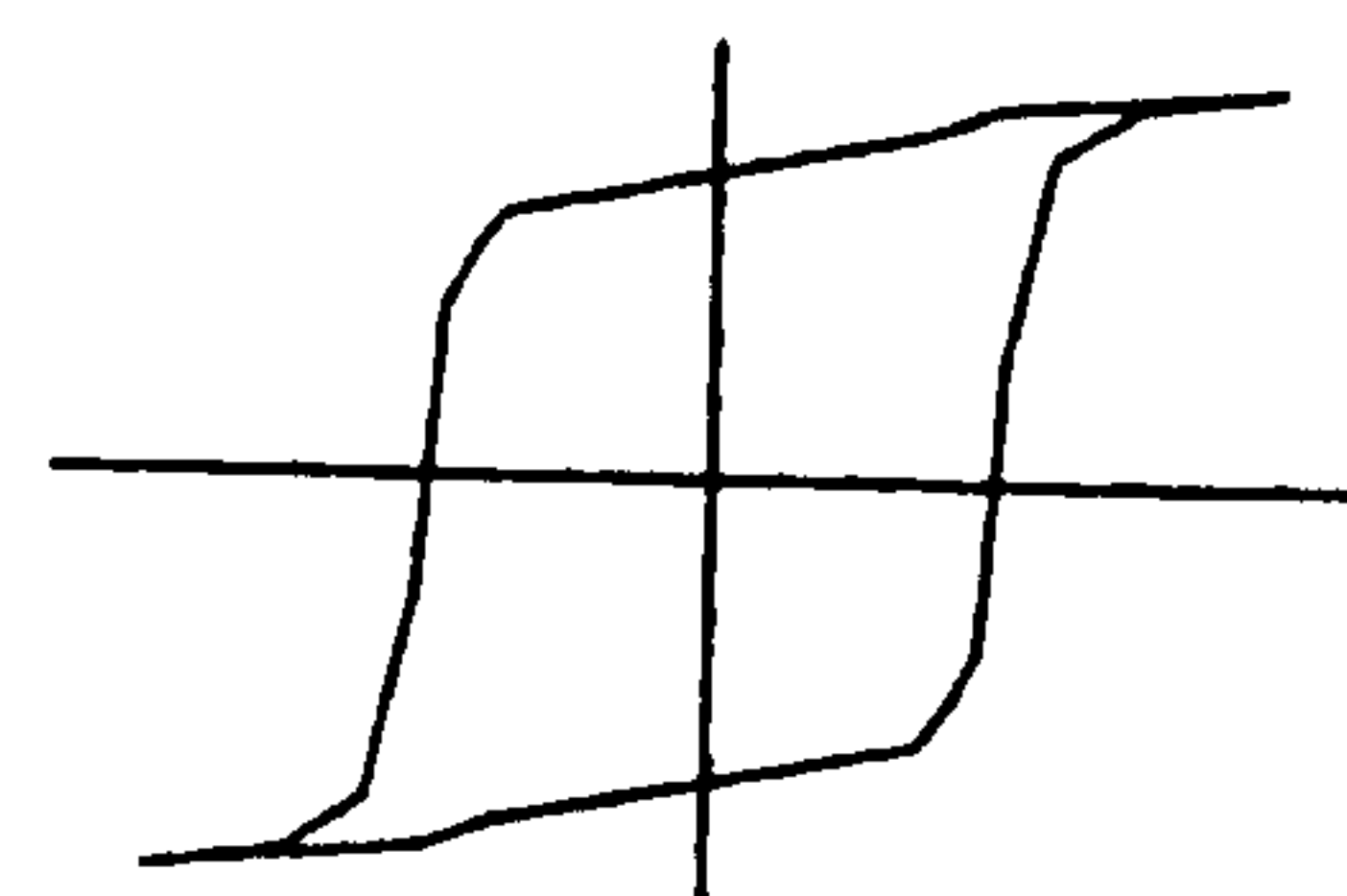
Descripción



Cuando se somete una substancia ferromagnética a los efectos de un campo magnético, el primero se magnetiza. La relación B—H puede representarse por medio de una curva de histéresis.



Substancia magnética suave



Substancia ferromagnética

Ejemplo

Curva de histéresis de un material magnético suave

	1	2	3	4	5	6	7	8	9
H	0,4	1,0	2,0	3,0	4,0	2,0	1,0	0,5	0,3
B	0,5	0,86	1,2	1,32	1,4	1,31	1,22	1,13	1,1

•Número de datos: 17

•Número de datos en el ciclo principal: 12

	10	11	12	13	14	15	16	17
H	0	-0,3	-0,5	-0,8	-1,0	-2,0	-3,0	-4,0
B	0,96	0,66	0	-0,53	-0,72	-1,15	-1,33	-1,4

•Hasta 20 datos.

Preparativos y procedimiento

•Almacene el programa de la página siguiente.

Contenido de las memorias	A	Número de datos	H		O		V	
	B	Número de datos en el bucle principal	I		P		W	
	C		J		Q		X	
	D		K		R		Y	
	E		L		S		Z	
	F		M		T			
	G	F[1] ~ F[20] H	N		U		Z[1] ~ Z[20] B	

No.

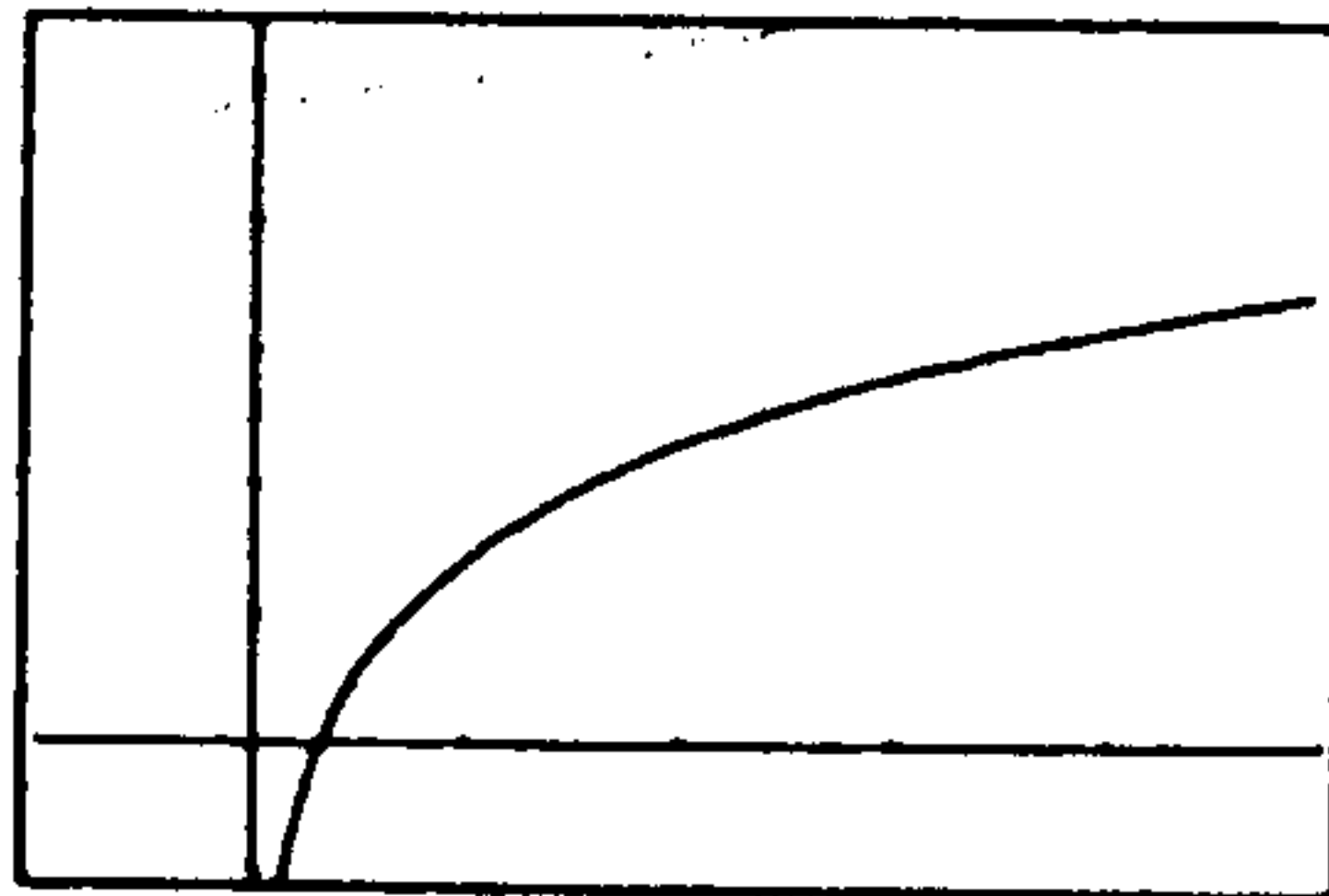
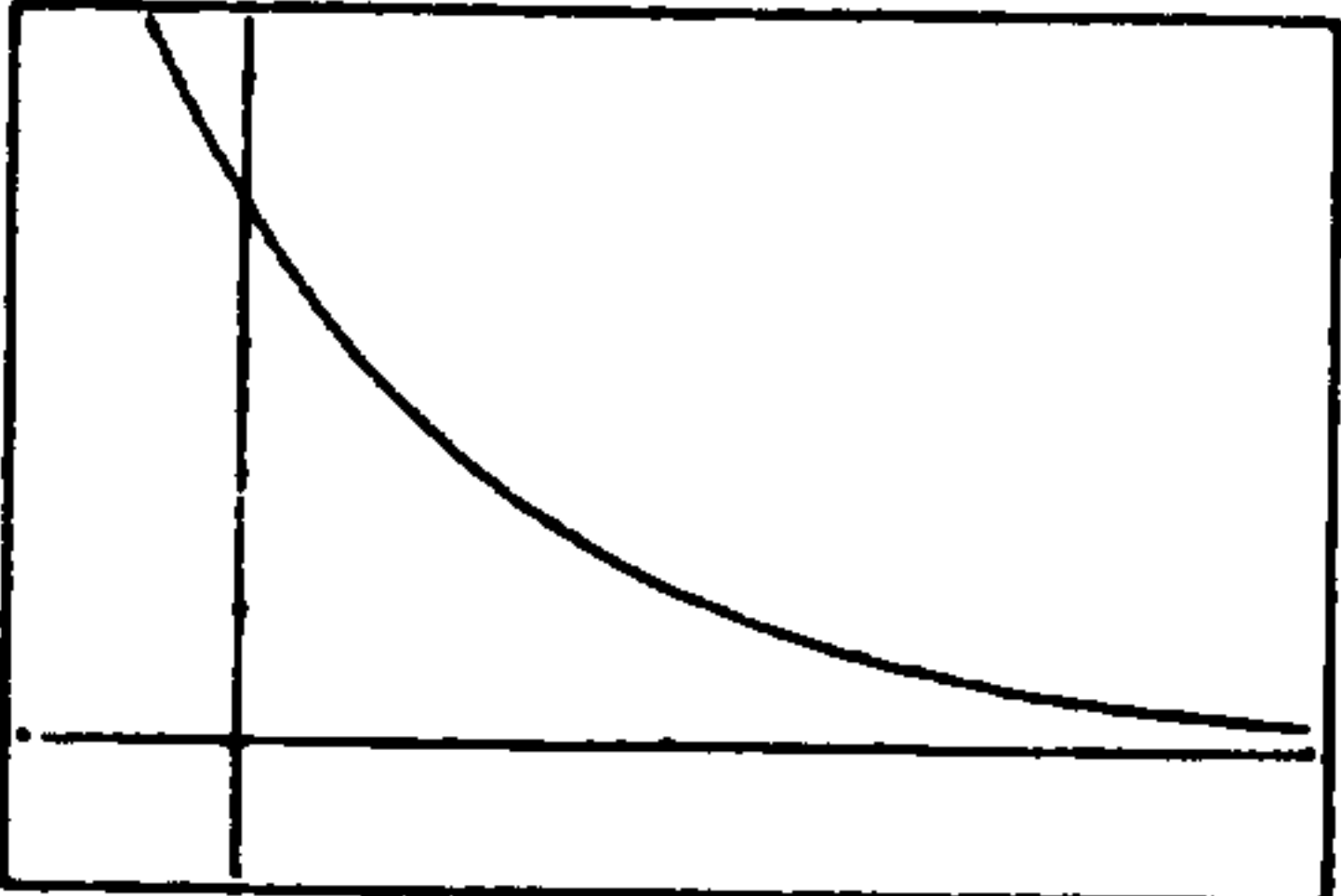
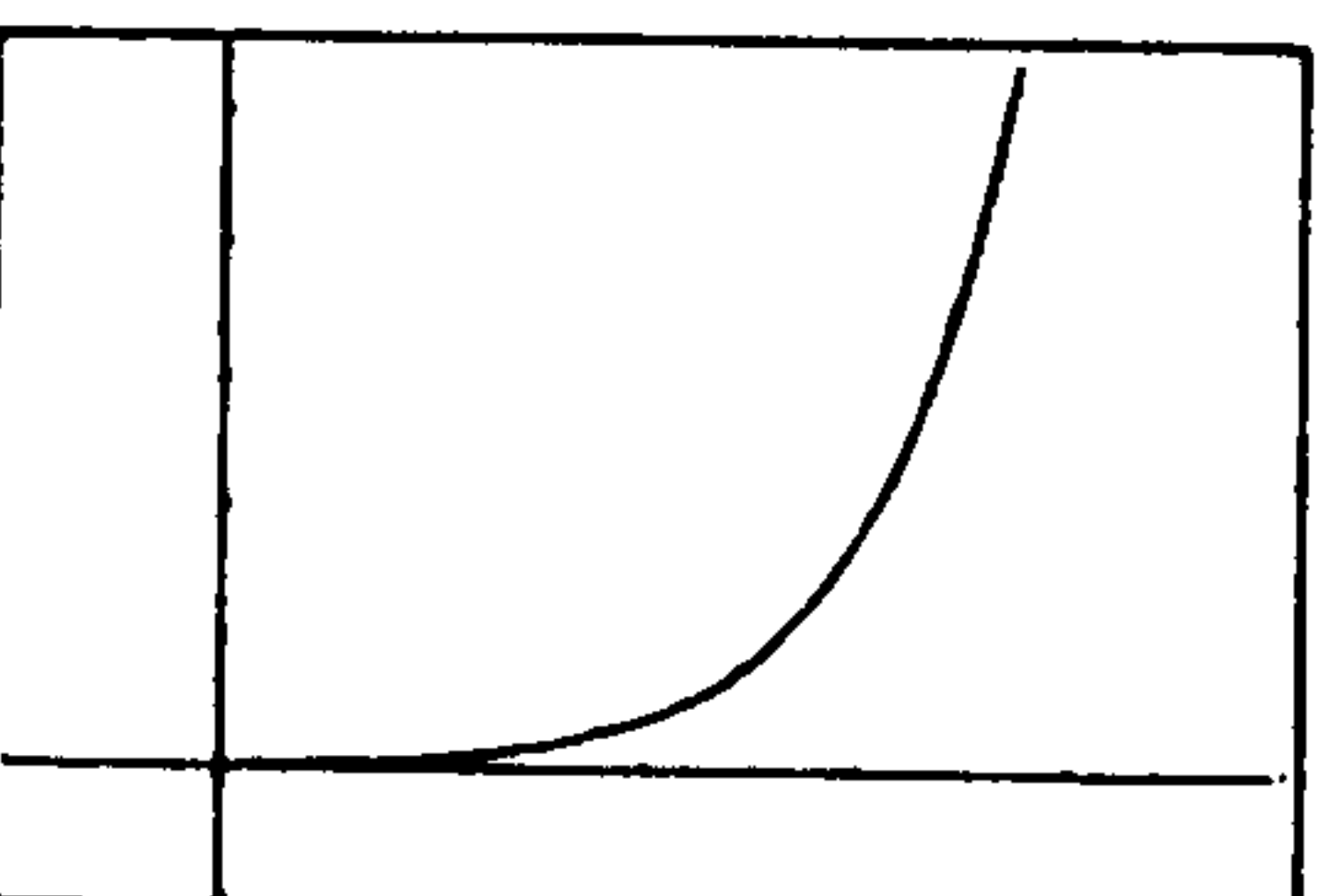
8

Línea	MODE	2	Programa																Notas	Número de pasos
1	Range	(-)	4	.	7	.	4	.	7	.	1	.	(-)	1	.					15
2		5	5	.	1	.	5	5	.	0	.	5	:							27
3	Defm	2	0	:																31
4	"	N	O	.	O	F	SPACE	D	A	T	A	"	?	→	A					46
5	:																			47
6	Lbl	9	:																	50
7	"	M	A	I	N	SPACE	L	O	O	P	SPACE	N	"	?	→					65
8	B	:																		67
9	B	>	2	0	⇒	Goto	9	:												75
10	1	→	C	:	Plot	0	,	0	:											84
11	Lbl	0	:	"	H	=	"	?	→	F	[C]	:						98
12	"	B	=	"	?	→	Z	[C]	:									109
13	Plot	F	[C]	,	Z	[C]	:	Line	▲							122
14	C	+	1	→	C	:														128
15	C	≠	A	+	1	⇒	Goto	0	:											137
16	A	-	B	+	1	→	D	:												145
17	Lbl	1	:	Plot	(-)	F	[D]	,	(-)	Z	[D]					160
18	:	Line	:																	163
19	D	+	1	→	D	:														169
20	D	≠	A	+	1	⇒	Goto	1	:											178
21	"	E	N	D	"															183
22																				
23																				
24																				
25																				
26																				
27																				
28																				
29																				
30																				
31																				
32																				
33																				
34																				
35																				

Memoria 20 × 8 = 160

Total 343 pasos

CASIO HOJA DE PROGRAMA

Programa para		Curva de regresión		No.		9		
Descripción								
i Curva de regresión logarítmica								
Fórmula de regresión: $y = A + B \ln x$								
$B = \frac{n \cdot \Sigma (y \cdot \ln x) - \Sigma \ln x \cdot \Sigma y}{n \Sigma (\ln x)^2 - (\Sigma \ln x)^2}$								
$A = \frac{\Sigma y - B \cdot \Sigma \ln x}{n}$								
								
ii Curva de regresión exponencial								
Fórmula de regresión: $y = A \cdot e^{Bx}$								
$B = \frac{n \Sigma (x \ln y) - \Sigma x \cdot \Sigma \ln y}{n \cdot \Sigma x^2 - (\Sigma x)^2}$								
$A = e^{\left(\frac{\Sigma \ln y - B \cdot \Sigma x}{n} \right)}$								
								
iii Curva de regresión de potencia								
Fórmula de regresión: $y = A \cdot x^B$								
$B = \frac{n \Sigma (\ln x \cdot \ln y) - \Sigma \ln x \cdot \Sigma \ln y}{n \cdot \Sigma (\ln x)^2 - (\Sigma \ln x)^2}$								
$A = \frac{\Sigma \ln y - B \cdot \Sigma \ln x}{n}$								
								
*Vea el ejemplo en la página 162.								
Preparativos y procedimiento								
●Almacene el programa de la página siguiente.								
Contenido de las memorias	A	A o ln A	H	$\Sigma (\ln x)^2$	O		V	Σx
	B	B	I		P	Σy^2	W	n
	C	$\Sigma \ln x$	J		Q	Σy	X	Dato x
	D	$\Sigma \ln y$	K		R	Σxy	Y	Dato y
	E	$X \Sigma \ln y$	L		S	Selección de 1 ~ 3	Z	
	F	$Y \Sigma \ln x$	M		T			
	G	$\Sigma (\ln x \cdot \ln y)$	N		U	Σx^2		

		No.	
		9	
Línea	Programa	Notas	Número de pasos
1	P0 [SHIFT] [MODE] [F1] → LR 2		
2	Scl : Cls : 0 → C ~ H :		10
3	" Range O K ? " ▲		17
4	" A C → Prog 1 SPACE X : ? " :		29
5	Lbl 1 :		32
6	" X : " ? → X :		40
7	" Y : " ? → Y :		48
8	In X + C → C : In Y + D → D : X		63
9	In Y + E → E : Y In X + F → F :		78
10	In X × In Y + G → G : (In X) x ²		93
11	+ H → H :		98
12	X , Y DT ▲		103
13	Goto 1		105
14			
15	P1 [MODE] [F2] → COMP		
16	" L → 1 SPACE E → 2 SPACE P → 3 : " ?		15
17	→ S :		18
18	S = 1 ⇒ Prog 7 :		25
19	S = 2 ⇒ Prog 8 :		32
20	S = 3 ⇒ Prog 9 :		39
21	" E N D "		44
22			
23	P7 [SHIFT] [MODE] [F1] → LR 2		
24	(W F - C Q) (W H - C x ²) x ⁻¹		15
25	→ B : (Q - B C) W x ⁻¹ → A :		29
26	Graph A + B In X ▲		36
27	" A : " ▲ A ▲		43
28	" B : " ▲ B ▲		50
29			
30			
31			
32			
33			
34			
35			

CASIO HOJA DE PROGRAMA

Programa para	Curva de regresión	No.	9
---------------	--------------------	-----	---

Ejemplo

Regresión exponencial de los siguientes datos:

x_i	2,2	5,6	9,5	13,8	18,0	23,2	29,9	37,8
y_i	35,6	28,1	23,0	17,9	12,9	10,2	6,2	4,0

Trazado de una curva de regresión exponencial y uso de la función de rastreo para estimar el valor de y cuando $x = 20$. Obtención de los valores de A y B de la fórmula de regresión.

Límites:

X min : -10	Y min : -10
X max : 50	Y max : 55
X scl : 10	Y scl : 10

Preparativos y procedimiento

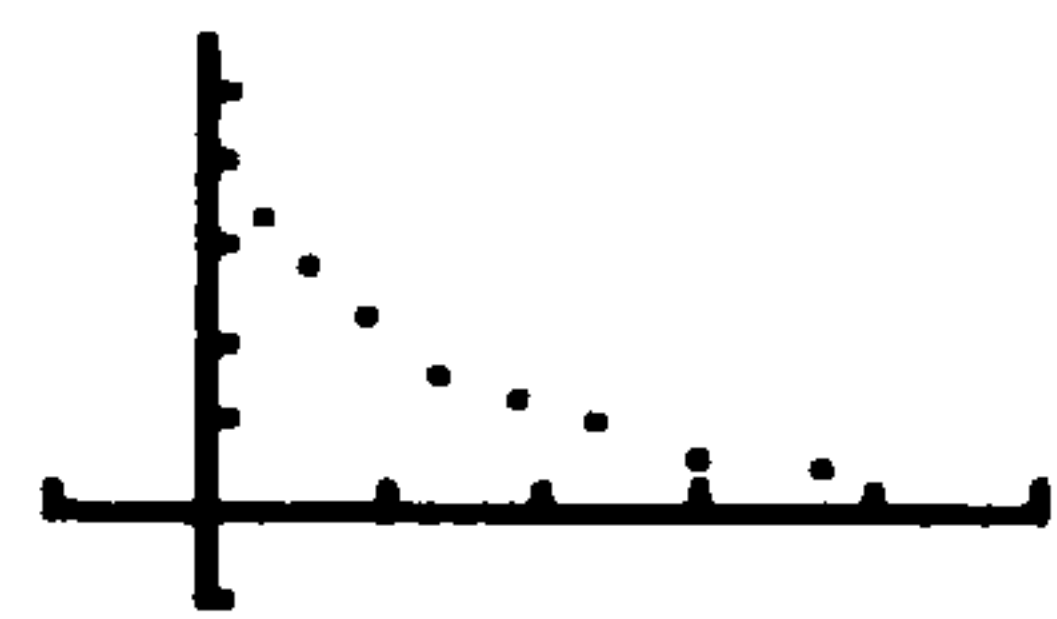
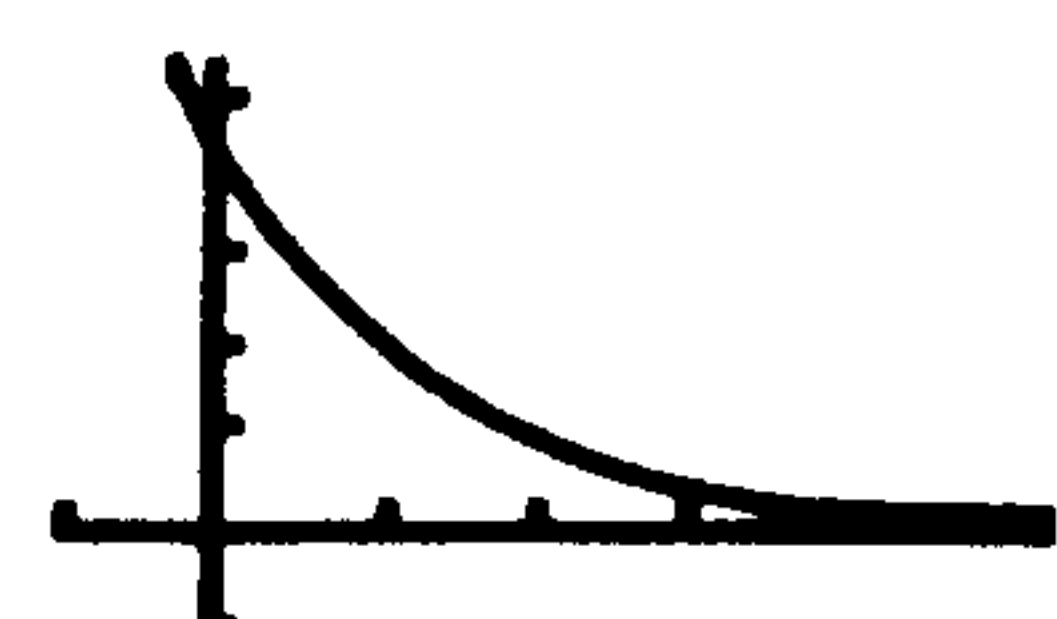
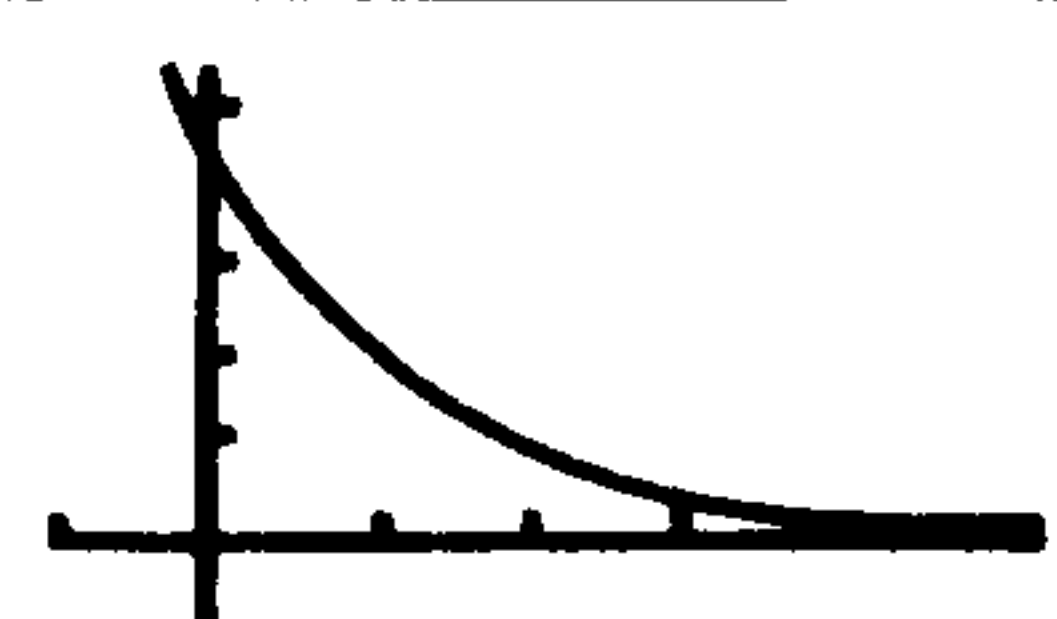
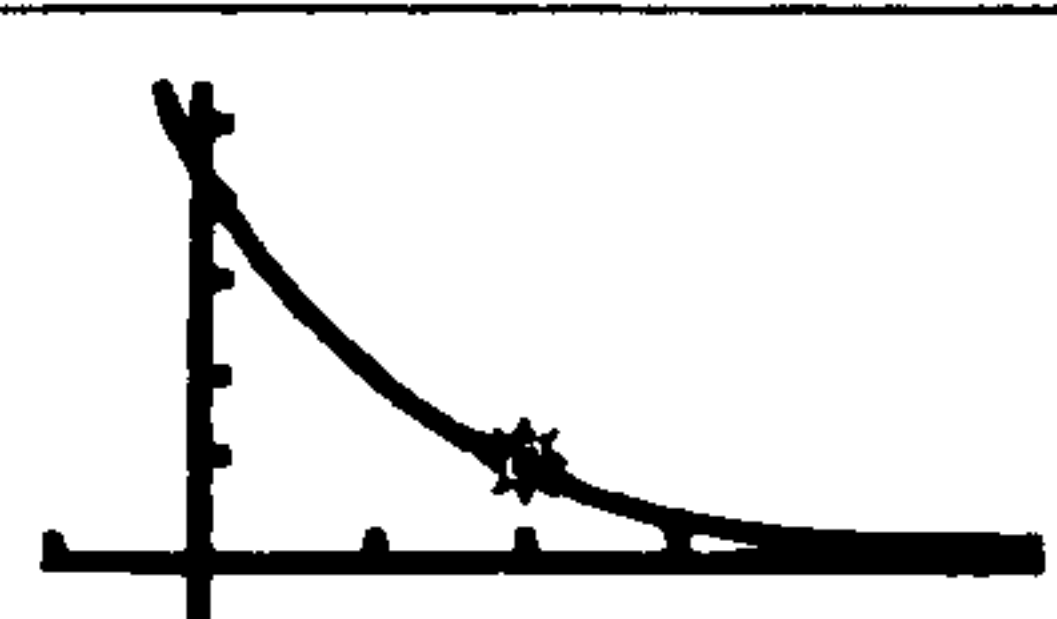
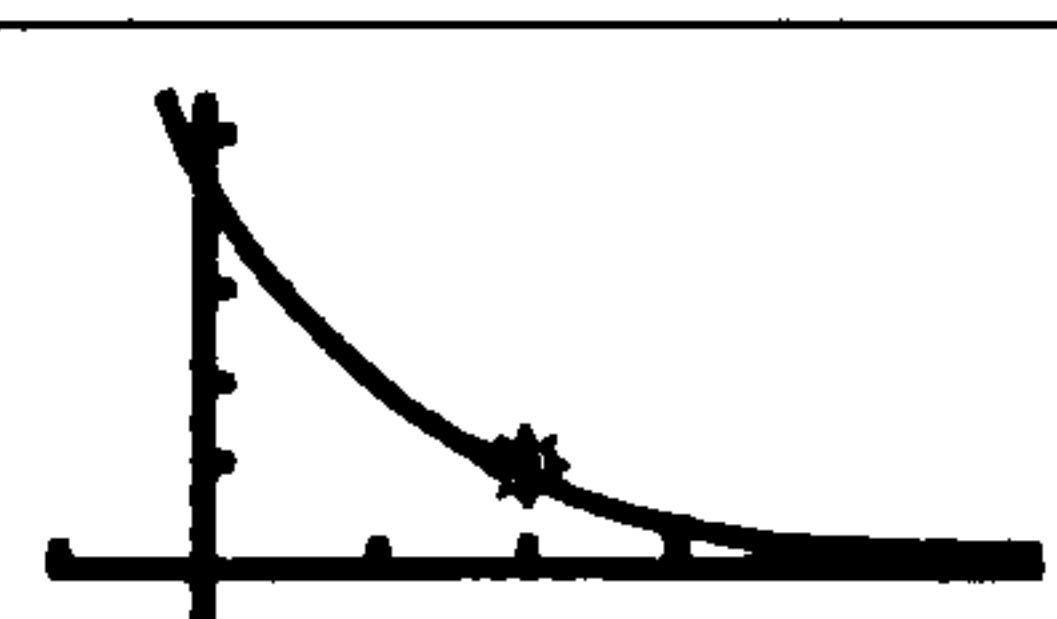
●Almacene el programa de la página siguiente.

Contenido de las memorias	A		H		O		V	
	B		I		P		W	
	C		J		Q		X	
	D		K		R		Y	
	E		L		S		Z	
	F		M		T			
	G		N		U			

No. 9

Línea	MODE 2	Programa	Notas	Número de pasos
1	P8	SHIFT MODE → LR 2		
2	(W E - V D) (W	U - V x^2) x^{-1}		15
3	→ B : (D - B V)	W x^{-1} → A :		29
4	Graph e^x A × e^x B X			37
5	" A : " \blacktriangle e^x A \blacktriangle			45
6	" B : " \blacktriangle B \blacktriangle			52
7				
8	P9	SHIFT MODE → LR 2		
9	(W G - C D) (W	H - C x^2) x^{-1}		15
10	→ B : (D - B C)	W x^{-1} → A :		29
11	Graph e^x A × X x^{-1} B			37
12	" A : " \blacktriangle e^x A \blacktriangle			45
13	" B : " \blacktriangle B \blacktriangle			52
14				
15			Total 303 pasos	
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

Programa para Curva de regresión		No. 9
Paso	Entrada desde el teclado	Pantalla
1	Prog 0 EXE (Verificación de los límites)	LR2 Range OK? Disp
2	Establecimiento de los límites AC Range	LR2 Xmin? -3.8
	SHIFT (←) 10 EXE 50 EXE 10 EXE SHIFT (←) 10 EXE 55 EXE 10	
3	EXE Prog 0 EXE	LR2 Range OK?
4	EXE	LR2 X : ?
5	2.2 EXE	LR2 Y : ?
6	35.6 EXE	LR2 Disp 2.2
7	EXE	LR2 X : ?
8	(Entrada de los datos por orden.) ⋮	
9	4.0 EXE	LR2 Disp 37.8

Programa para Curva de regresión		No. 9
Paso	Entrada desde el teclado	Pantalla
10	G-T	 Disp
11	AC Prog 1 EXE	L → 1 E → 2 P → 3 ?
12	2 EXE (Selección de la regresión exponencial)	 Disp
13	Traca	 X = Disp -5.26315
14	⇨ ~ Desplazamiento del puntero a X = 20.	 X = Disp 20.
15	SHIFT X↔Y	 Y = Disp 11.86149
16	EXE	LR2 A : Disp
17	EXE	LR2 Disp 40.68214076
18	EXE	LR2 B : Disp
19	EXE	LR2 Disp -0.061624605

Programa para		No. 9	
Paso	Entrada desde el teclado	Pantalla	
20	<div>EXE</div>	LR2 END	
21			
22			
23			
24			
25			
26			
27			
28			
29			

Programa para		No.	
Paso	Entrada desde el teclado	Pantalla	

CASIO HOJA DE PROGRAMA

Programa para

Diagrama "parade"

No.10

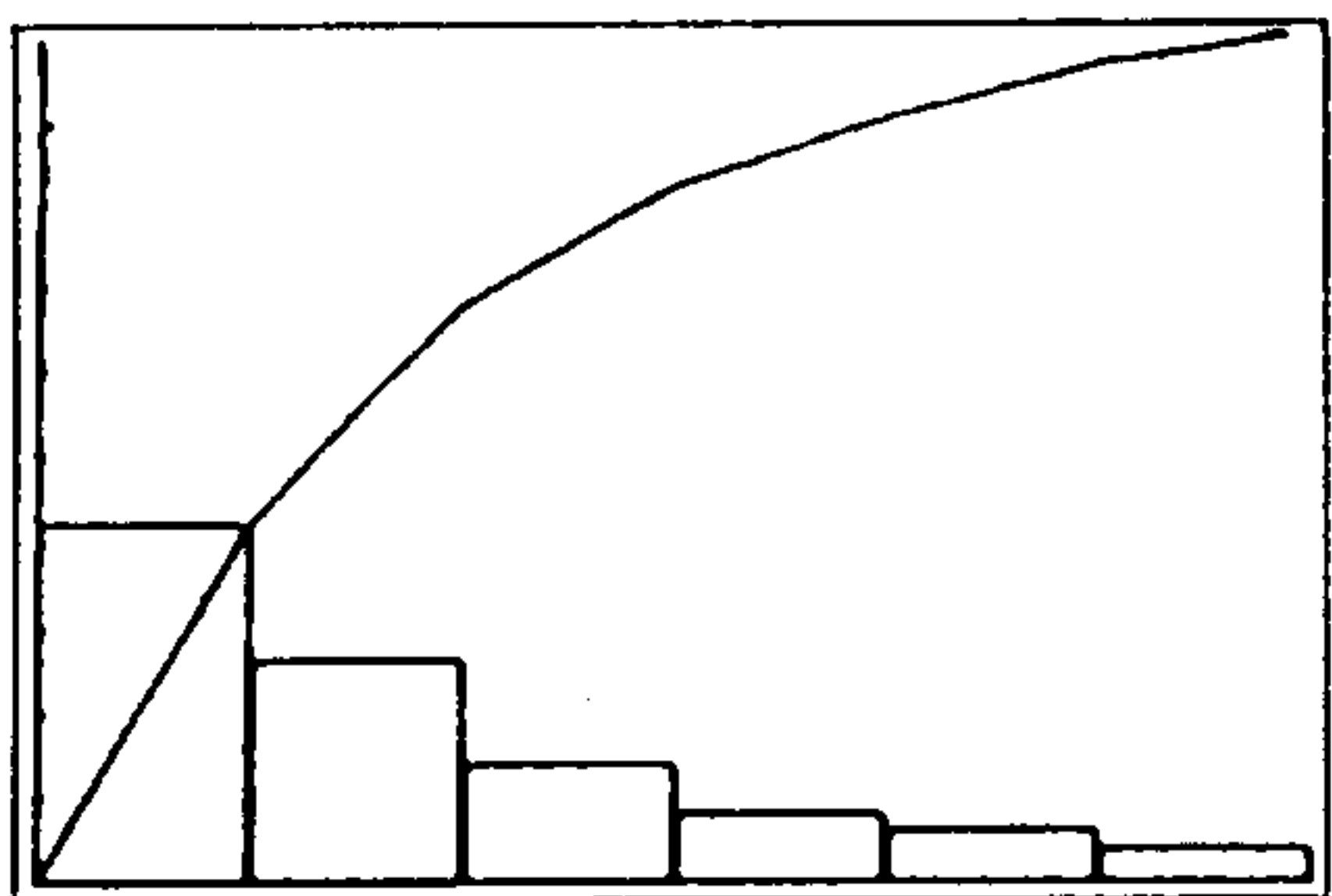
Descripción

Un ejemplo en la aplicación de diagramas "parade" es la solución de problemas de control de calidad. El problema se analiza cuantitativamente en base a datos referentes a su alcance, para determinar los puntos que más atención requieren.

Eje horizontal: Clasificación de problemas
(Punto 6 en este ejemplo)

Vertical: (Derecha) Relación de ocupación
(Izquierda) Alcance del problema en cada clasificación

Frecuencia del problema



Clasificaciones
(áreas del problema)

Ejemplo

Trazado de un diagrama usando los datos que se encuentran a la derecha.

Areas del problema	Frecuencia del problema
A	105
B	65
C	35
D	20
E	15
Otros	10

Preparativos y procedimiento

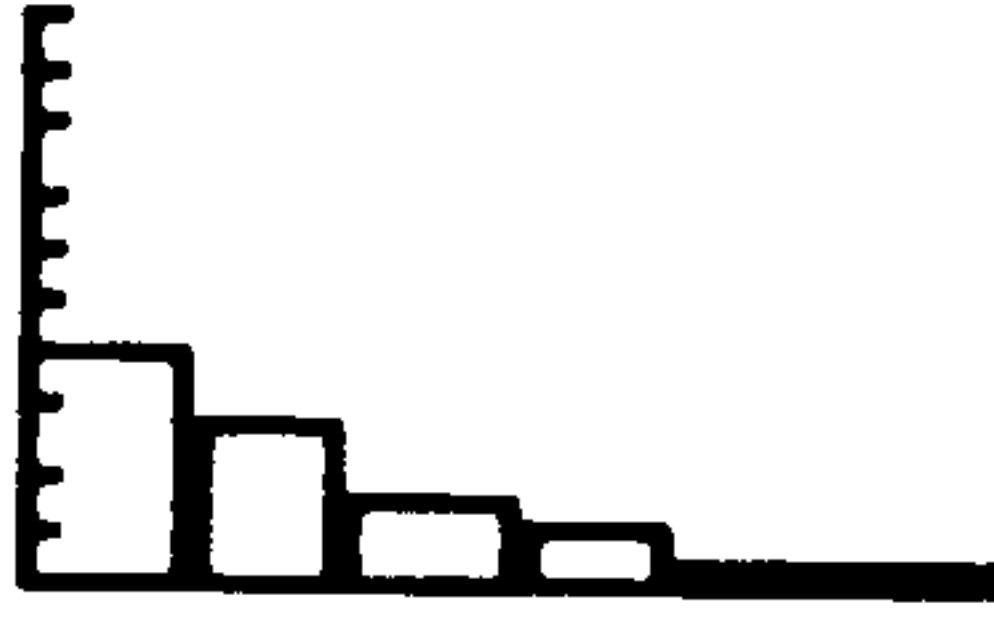
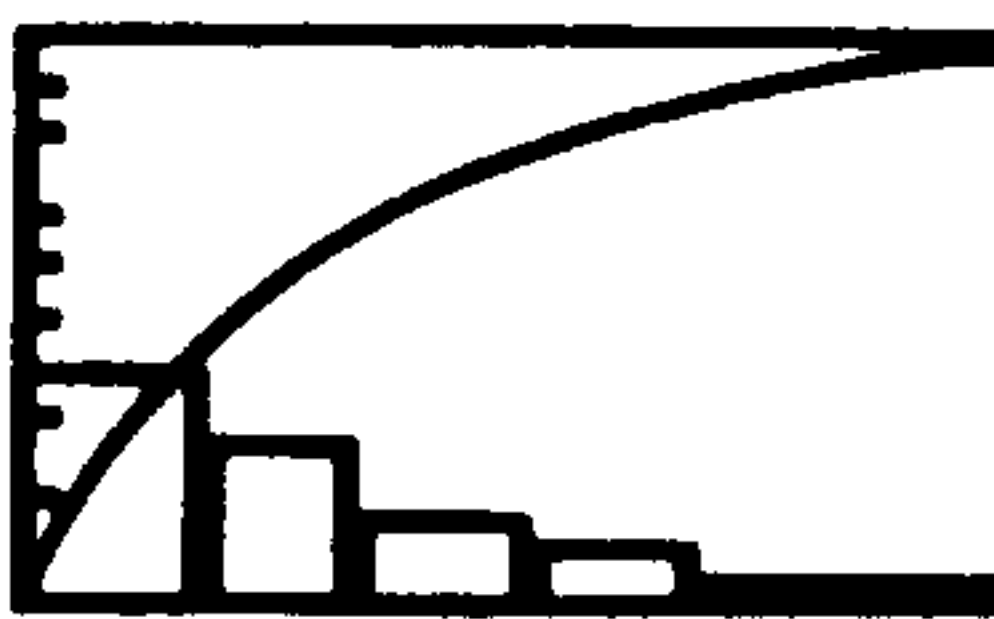
•Almacene el programa de la página siguiente.

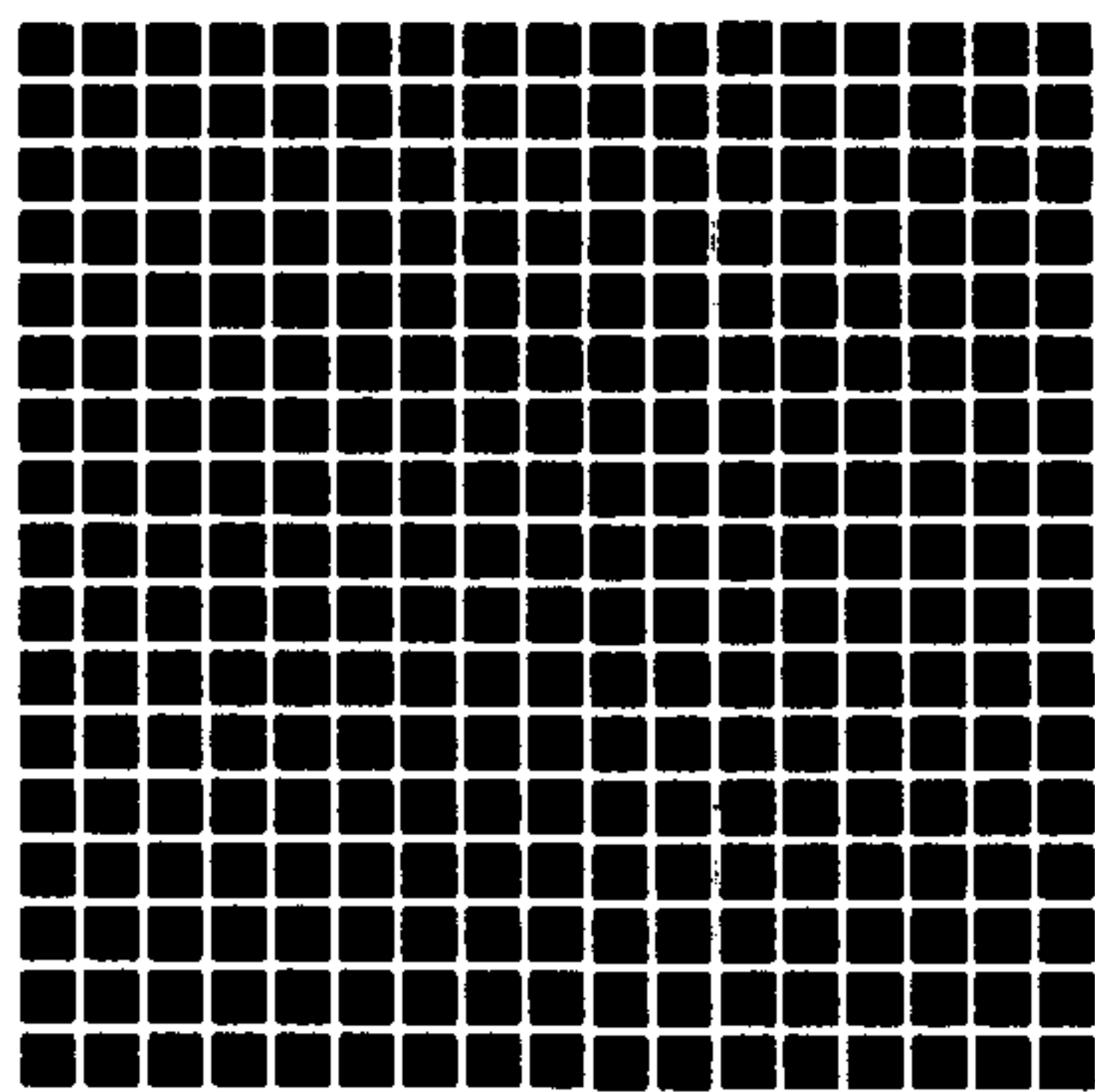
Contenido de las memorias	A	Entrada de los datos	H		O		V	
	B		I		P		W	n
	C		J		Q		X	Cuenta de los datos
	D		K		R		Y	
	E		L		S	Visualización de la cuenta	Z	Suma de los datos
	F		M		T			Z[1] ~ Z[6]
	G		N		U			

- 168 -

		No. 10										Notas	Número de pasos
Linea	MODE 2	Programa											
1	P0	SHIFT	MODE	X	→	SD2							
2	Scl	:	Mcl	:	Defm	6	:						7
3	Range	0	,	6	,	1	,	0	,	2	0	,	21
4	Lbl	1	:										24
5	"	D	A	T	A	"	?	→	A	:			34
6	X	:	A	DT	:								39
7	X	+	1	→	X	:	X	≤	5	⇒	Goto	1	52
8	Range	,	,	,	,	W	,	W	÷	1	0	:	64
9	Graph	▲											66
10	Plot	0	,	0	:								71
11	1	→	S	:									75
12	Lbl	2	:										78
13	Z	[S]	+	Z	→	Z	:				87
14	Plot	S	,	Z	:	Line	:						94
15	S	+	1	→	S	:	S	≤	6	⇒	Goto	2	107
16	Graph	W											109
17													
18													
19													
20												Memoria 6 × 8 = 48	
21												Total 157 pasos	
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													
34													
35													

- 169 -

Programa para Diagrama “parade”		No. 10
Paso	Entrada desde el teclado	Pantalla
1	Prog 0 EXE	SD 2 DATA ?
2	105 EXE	SD 2 DATA ?
3	65 EXE	SD 2 DATA ?
4	(Entrada de los datos en orden.) ⋮	
5	10 EXE (Visualización de gráfico de barras)	 Disp
6	EXE (Visualización del diagrama “parade”)	
7		
8		
9		
10		



Apéndice

- Referencia de funciones
- Tabla de mensajes de errores
- Gamas de ingreso de funciones
- Especificaciones

Referencia de funciones

■ Cálculos manuales

Especificación del modo	Modo COMP (MODE +)	Cálculos con las 4 operaciones matemáticas fundamentales y con funciones.
	Modo BASE-N (MODE =)	Conversiones, cálculos y operaciones lógicas con números binarios, octales, decimales y hexadecimales.
	Modo SD1 (MODE X)	Cálculos de desviación estándar (estadísticas con 1 variable)
	Modo LR1 (MODE ÷)	Cálculos de regresión (estadísticas con pares de variables)
	Modo SD2 (SHIFT MODE X)	Para la creación de gráficos estadísticos con una sola variable. (Gráficos de barras y curvas de distribución normal)
	Modo LR2 (SHIFT MODE ÷)	Para la creación de gráficos estadísticos con dos variables. (Líneas de regresión)
Funciones	Funciones tipo A	Mandos de funciones entrados inmediatamente antes que el valor numérico. [x^2 , x^{-1} , $x!$, °, r, g, ° ’ ”]
	Funciones tipo B	Mandos de funciones entrados inmediatamente después que el valor numérico. [sen, cos, tan, sen^{-1} , cos^{-1} , tan^{-1} , sinh, cosh, tanh, sinh^{-1} , cosh^{-1} , tanh^{-1} , log, ln, e^x , 10^x , $\sqrt{}$, $\sqrt[3]{}$, Abs, Int, Frac, etc.]
	Funciones con pares de variables	Entrada de mandos de funciones entre dos valores numéricos. Entrada de valores numéricos entre paréntesis después que el mando de la función. [$A x^y B$ (A elevado a potencia B), $B \sqrt[x]{} A$ (A elevado a la potencia 1/B), Pol (A, B), Rec (A, B)] <i>*A y B son valores numéricos.</i>
	Funciones de ejecución inmediata	Valor visualizado que cambia por cada pulsación de una tecla. [ENG, ENG, ° ’ ”]

Cálculos con números binarios, octales, decimales y hexadecimales MODE —	Establecimiento del sistema numérico	Decimales Dec EXE Hexadecimales ... Hex EXE Binarios Bin EXE Octales Oct EXE
	Especificación del sistema numérico	Sea cual fuere el sistema numérico en curso, se pueden entrar números individuales de cualquier sistema. Para especificar: Decimales SHIFT d Hexadecimales ... SHIFT h Binarios SHIFT b Octales SHIFT o
	Operaciones lógicas	Los valores numéricos ingresados se convierten en binarios y cada bit es comprobado. El resultado se convierte al sistema usado en la entrada y se visualiza. Not Inversión de cada bit and Producto lógico de bit or Suma lógico de bit xor Suma lógica exclusiva de cada bit xnor Suma lógica negativa exclusiva de cada bit
Cálculos de desviación estándar MODE ×	Borrado de los datos	SHIFT ScI EXE
	Entrada de los datos	Dato [;frecuencia] DT <i>*La frecuencia puede omitirse.</i>
	Supresión de datos	Dato [;frecuencia] CL <i>*La frecuencia puede omitirse.</i>
	Visualización del resultado	Número de datos (n) ALPHA 3 (n) EXE Sumatoria (Σx) ALPHA 2 (Σx) EXE Sumatoria de los cuadrados (Σx^2) ALPHA 1 (Σx^2) EXE Media (\bar{x}) SHIFT \bar{x} EXE Desviación estándar de una población ($x\sigma_n$) SHIFT $x\sigma_n$ EXE Desviación estándar por muestreo ($x\sigma_{n-1}$) SHIFT $x\sigma_{n-1}$ EXE

Cálculos de regresión MODE	Borrado de los datos	SHIFT SCI EXE
	Entrada de los datos	dato x dato y [;frecuencia] DT *La frecuencia puede omitirse.
	Supresión de datos	dato x dato y [;frecuencia] CL *La frecuencia puede omitirse.
	Visualización del resultado	Número de datos (n) .. ALPHA 3 (n) EXE Sumatoria de x (Σx) ... ALPHA 2 (Σx) EXE Sumatoria del y (Σy) .. ALPHA 5 (Σy) EXE Sumatoria del cuadrado de x (Σx^2) ALPHA 1 (Σx^2) EXE Sumatoria del cuadrado del y (Σy^2) ALPHA 4 (Σy^2) EXE Sumatoria del producto de x e y (Σxy) ALPHA 6 (Σxy) EXE Media de x (\bar{x}) SHIFT X EXE Media de y (\bar{y}) SHIFT Y EXE Desviación estándar de población para x ($x\sigma_n$) SHIFT xσn EXE Desviación estándar de población para y ($y\sigma_n$) SHIFT yσn EXE Desviación estándar por muestreo para x ($x\sigma_{n-1}$) SHIFT xσn-1 EXE Desviación estándar por muestreo para y ($y\sigma_{n-1}$) SHIFT yσn-1 EXE Término constante de la fórmula de regresión (A) SHIFT A EXE Coeficiente de regresión (B) SHIFT B EXE Coeficiente de correlación (r) SHIFT r EXE Valor estimado de x (\hat{x}) SHIFT X EXE Valor estimado de y (\hat{y}) SHIFT Y EXE

Funciones especiales	Ans	Se almacena en la memoria el último resultado obtenido en los cálculos manuales o por programa. El mismo se obtiene presionando Ans . *La mantisa del valor numérico tiene 10 dígitos.
	Repetición	<ul style="list-style-type: none"> •Una vez obtenido el resultado de un cálculo, la fórmula puede obtenerse presionando ⇐ o ⇒. •La función de repetición no será borrada aun cuando se presione la tecla AC o se apague la unidad. •Siempre que aparezca un error, la pulsación de ⇐ o ⇒ lo cancela y hace que aparezca en la pantalla el paso de la fórmula que contiene el error, con el cursor parpadeando.
	Sentencias múltiples	Se usan dos puntos (:) para unir sentencias o fórmulas de cálculo. Si se unen por medio de “▲”, se visualiza el resultado del cálculo.
	Memorias	A las 26 memorias originales se pueden agregar de a una un máximo de 50 memorias adicionales, para un total de 76. Cada memoria resta 8 pasos de programa. MODE ◻ número de memorias EXE .

Función de gráficos	Gamas	Límites del gráfico Xmin Valor mínimo de x Xmax Valor máximo de x Xscl Escala del eje x (separación entre puntos) Ymin Valor mínimo de y Ymax Valor máximo de y Yscl Escala del eje y (separación entre puntos)
	Trazado	Mueve el indicador sobre el gráfico. Se visualiza la ubicación corriente.
	Marcación de puntos	Enciende el puntero (punto parpadeante) en cualquier coordenada del gráfico en la pantalla.
	Líneas	Une por medio de una línea recta los dos puntos creados mediante la función de trazado.
	Factor	Define un factor para enfocar ampliando/reduciendo el detalle.
	Enfoque de detalles (Zoom)	Zoom $\times f$ Enfoca ampliando el gráfico de acuerdo con los factores de enfoque. Zoom $\times 1/f$ Enfoca reduciendo el gráfico de acuerdo con la inversa de los factores de enfoque. Zoom Org Retorna el gráfico visto en detalles a las dimensiones original.
	Desplazamiento de presentación (Scroll)	Desplaza la presentación para ver las partes de los gráficos que se encuentran fuera de la presentación.

■ Cálculos por programa

Entrada de programas	Modo de entrada	Modo WRT (MODE 2)
	Modo de cálculo	Modo conforme con el programa especificado por: MODE +, MODE −, MODE ×, MODE ÷.
	Especificación del área de programa	Se lleva el cursor al nombre de área de programa deseado (P0 hasta P9) por medio de las teclas ⇨ o ⇩, y se presiona EXE.
Ejecución de programas	Modo de ejecución	Modo RUN (MODE 1)
	Especificación del área de programa	La ejecución comienza con [Prog] nombre de área de programa EXE. Nombre de área de programa: P0 hasta P9
Compaginación de programas	Modo de entrada	Modo WRT (MODE 2)
	Especificación del área de programa	El cursor se desplaza al nombre de área de programa deseado (P0 hasta P9) usando ⇨ o ⇩, y se presiona EXE.
	Compaginación	El cursor se lleva a la posición donde se desea corregir por medio de las teclas ⇩ o ⇨. •Presione la tecla correcta para corregir. •Presione la tecla DEL para suprimir. •Presione SHIFT INS para realizar las inserciones deseadas.
Borrado de programas	Modo de borrado	Modo PCL (MODE 3)
	Borra un programa específico	El cursor se lleva al nombre del área de programa deseado (P0 hasta P9) por medio de las teclas ⇨ o ⇩, y se presiona AC.
	Borra todos los programas	Presione SHIFT MC.

Mandos de programa	Salto incondicional	La ejecución del programa pasa al Lbl n correspondiente al Goto n . * $n = 0$ hasta 9
	Salto condicional	<p>Si la expresión comparativa se cumple, se ejecuta la sentencia que se encuentra a continuación de "⇒". En caso contrario, la ejecución salta a la sentencia que se encuentra a continuación del siguiente ":" o "▲".</p> <p>Se cumple</p> <p>No se cumple</p> <p>(F): Fórmula (R): Operador comparativo (S): Sentencia</p> <p>*Los operadores comparativos son: =, ≠, >, <, ≥, ≤.</p>
	Salto con cuenta	<p>Se aumenta o disminuye un valor almacenado en una memoria. Cuando el valor es desigual a 0, se ejecuta la sentencia siguiente. Si es igual a 0, se lleva a cabo un salto a la sentencia que se encuentra a continuación del ":" o "▲" siguiente.</p> <p>Aumento</p> <p>Isz Cuando (V) ≠ 0</p> <p>Nombre de la memoria : S</p> <p>Cuando (V) = 0</p> <p>Disminución</p> <p>Dsz Cuando (V) ≠ 0</p> <p>Nombre de la memoria : S</p> <p>Cuando (V) = 0</p> <p>(S): Sentencia (V): Valor en la memoria</p>
	Subrutinas	La ejecución del programa salta de la rutina principal a la subrutina indicada por Prog n ($n = 0$ hasta 9). Una vez completada la ejecución de la subrutina, se vuelve a la sentencia que se encuentra a continuación del Prog n que provocó el salto.

Tabla de mensajes de errores

Mensaje	Significado	Medidas
Syn ERROR	<ol style="list-style-type: none"> ① Fórmula de cálculo con un error. ② La fórmula en el programa contiene un error. 	<ol style="list-style-type: none"> ① Use o para visualizar y corregir el error. ② Use o para visualizar la porción donde se encuentra el error, presione AC y corrija el programa en el mode WRT.
Ma ERROR	<ol style="list-style-type: none"> ① El resultado de cálculo excede los límites permitidos. ② El cálculo se lleva a cabo excediendo el límite de entrada de una función. ③ Operación ilegal (división por cero, etc). 	<ol style="list-style-type: none"> ① ② ③ Verifique el valor numérico entrado y corrijalo. Cuando se utilicen memorias, verifique que los valores numéricos almacenados en ellas sean correctos.
Go ERROR	<ol style="list-style-type: none"> ① Falta el Lbl n correspondiente al Goto n. ② No hay programa almacenado en el área P n correspondiente al Prog n. 	<ol style="list-style-type: none"> ① Entre el Lbl n correspondiente al Goto n, o borre este último en caso de que no se necesite. ② Almacene un programa en al área P n correspondiente a Prog n, o borre este último en caso de que no se necesite.
Ne ERROR	<ul style="list-style-type: none"> • Se han superado los 10 niveles permitidos para la inclusión de subrutinas. 	<ul style="list-style-type: none"> • Asegúrese de que no se haya intentado el uso de un Prog n para retornar de una subrutina a la rutina principal. Borre todo Prog n que no sea necesario. • Verifique los destinos de los saltos a las subrutinas y asegúrese de que no haya saltos hacia el área de programa original. Asegúrese de que el retorno al área original sea el correcto.

Stk ERROR	<ul style="list-style-type: none"> Ejecución de cálculos que exceden la capacidad de las memorias temporales para valores numéricos y para los cálculos. 	<ul style="list-style-type: none"> Simplifique las fórmulas para no exceder la capacidad de las memorias temporales, de 10 niveles para los valores numéricos y de 24 para los cálculos. Divida la fórmula en 2 o más partes.
Mem ERROR	<ul style="list-style-type: none"> La ampliación de la memoria excede el nivel restante en el programa. Intento de usar una memoria tal como Z[5] cuando la memoria no ha sido ampliada. 	<ul style="list-style-type: none"> Presione MODE □ (Defm) para ampliar la memoria al nivel necesario. Utilice memorias dentro del número corriente de memorias.
Arg ERROR	<ul style="list-style-type: none"> Ingreso incorrecto del argumento. Ej. Ingreso de valor negativo para Defm, se ingresa otro valor diferente de 1~9 para n, etc. 	<ul style="list-style-type: none"> Ingrese correctamente el argumento.

Gamas de ingreso de funciones

Función	Gama de entrada	Dígitos internos	Precisión	Notas
sen x cos x tan x	(Deg) $ x < 9 \times 10^{90}$ (Rad) $ x < 5 \times 10^7 \pi \text{ rad}$ (Gra) $ x < 1 \times 10^{10} \text{ grad}$	12 dígitos	Como regla, la precisión es ± 1 en el 10mo. dígito.	Sin embargo, para tan x : $ x \approx 90(2n+1)$: Deg $ x \approx \pi/2(2n+1)$: Rad $ x \approx 100(2n+1)$: Gra
sen ⁻¹ x cos ⁻¹ x tan ⁻¹ x	$ x \leq 1$ $ x < 1 \times 10^{100}$	„	„	
senh x cosh x tanh x	$ x \leq 230,2585092$ $ x < 1 \times 10^{100}$	„	„	Nota: Para senh y tanh, cuando $x=0$, los errores se acumulan y la precisión se afecta en un cierto punto.
senh ⁻¹ x cosh ⁻¹ x tanh ⁻¹ x	$ x < 5 \times 10^{99}$ $1 \leq x < 5 \times 10^{99}$ $ x < 1$	„	„	
log x ln x	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$	„	„	
10 ^{x} e^x	$-1 \times 10^{100} < x < 100$ $-1 \times 10^{100} < x \leq 230,2585092$	„	„	
\sqrt{x} x^2	$0 \leq x < 1 \times 10^{100}$ $ x < 1 \times 10^{50}$	„	„	
1/ x $\sqrt[3]{x}$	$ x < 1 \times 10^{100}, x \neq 0$ $ x < 1 \times 10^{100}$	„	„	
$x!$	$0 \leq x \leq 69$ (x es un número entero)	„	„	

Función	Gama de entrada	Dígitos internos	Precisión	Notas
Pol (x,y)	$\sqrt{x^2 + y^2} < 1 \times 10^{100}$	12 dígitos	Como regla, la precisión es ± 1 en el 10mo. dígito.	
Rec (r,θ)	$0 \leq r < 1 \times 10^{100}$ (Deg) $ \theta < 9 \times 10^{90}$ (Rad) $ \theta < 5 \times 10^7 \pi \text{ rad}$ (Gra) $ \theta < 1 \times 10^{10} \text{ grad}$	„	„	Sin embargo, para $\tan \theta$: $ \theta \neq 90(2n+1)$: Deg $ \theta \neq \pi/2(2n+1)$: Rad $ \theta \neq 100(2n+1)$: Gra
o, „ $\overleftarrow{o}, „$	$ a , b, c < 1 \times 10^{100}$ $0 \leq b, c$ $ x < 2.777777777 \times 10^{96}$ Presentación hexadecimal: $ x \leq 2777777,777$	„	„	
x^y	$x > 0$: $-1 \times 10^{100} < y \log x < 100$ $x = 0$: $y > 0$ $x < 0$: $y = n, \frac{1}{2n+1}$ (n es un número entero) Sin embargo; $-1 \times 10^{100} < \frac{1}{y} \log x < 100$	„	„	
$\sqrt[n]{y}$	$y > 0$: $x \neq 0$ $-1 \times 10^{100} < \frac{1}{x} \log y < 100$ $y = 0$: $x > 0$ $y < 0$: $x = 2n+1, \frac{1}{n}$ ($n \neq 0$, n es un número entero) Sin embargo; $-1 \times 10^{100} < \frac{1}{x} \log y < 100$	„	„	

Función	Gama de entrada	Dígitos internos	Precisión
$a^{b/c}$	<ul style="list-style-type: none"> •Resultados Total de números enteros, numerador y denominador deben estar dentro de 10 dígitos (incluyendo las marcas de división). <ul style="list-style-type: none"> •Ingreso El resultado se visualiza como una fracción para el número entero cuando el entero, numerador y denominador son menores de 1×10^{10} .	12 dígitos	Como regla, la precisión es ± 1 en el 10mo. dígito.
SD (LR)	$ x < 1 \times 10^{50}$ $ y < 1 \times 10^{50}$ $ n < 1 \times 10^{100}$ $x\sigma_n, y\sigma_n, \bar{x}, \bar{y}, A, B, r: n \neq 0$ $x\sigma_{n-1}, y\sigma_{n-1}: n \neq 0, 1$	„	„

Función	Gama de entrada
BASE-N	Valores luego de una variable en la siguiente gama: Dec: $-2147483648 \leq x \leq 2147483647$ Bin: $100000000000 \leq x \leq 111111111111$ (negativo) $0 \leq x \leq 011111111111$ (0, positivo) Oct: $20000000000 \leq x \leq 37777777777$ (negativo) $0 \leq x \leq 17777777777$ (0, positivo) Hex: $80000000 \leq x \leq \text{FFFFFFFF}$ (negativo) $0 \leq x \leq 7\text{FFFFFFF}$ (0, positivo)

* Los errores pueden ser acumulativos con los cálculos continuos internos tales como x^y , $\sqrt[n]{y}$, $x!$, $\sqrt[n]{x}$ afectando algunas veces la precisión.

Especificaciones

Modelo: fx-6300G

Funciones gráficas

Funciones de gráficos incorporadas: (20 tipos) \sin , \cos , \tan , \sin^{-1} , \cos^{-1} , \tan^{-1} , \sinh , \cosh , \tanh , \sinh^{-1} , \cosh^{-1} , \tanh^{-1} , \log , \ln , 10^x , e^x , x^2 , $\sqrt{}$, $\sqrt[3]{}$, x^{-1}

Tipos de gráficos: Gráficos de función generados por el usuario
Coordenadas rectangulares
Estadísticas con una sola variable: gráficos de barra, curvas de distribución normal
Estadísticas de variables en pares: líneas de regresión

Funciones gráficas: Especificación de gama, sobreposición de gráfico, trazado, enfoque de detalle ($\times f$, $\times 1/t$, factor, original (reanudar)), marcado de punto, línea, desplazamiento de presentación

Cálculos

Funciones básicas: Números negativos, exponentes, suma/resta/multiplicación/división con paréntesis (prioridad por lógica algebraica real)

Funciones científicas incorporadas: Trigonométricas/trigonométricas inversas (unidades de medición angular; grados, radianes y grados centesimales), hiperbólicas/hiperbólicas inversas, logarítmicas/exponenciales, recíprocos, factoriales, raíces cuadradas, raíces cúbicas, potencias, raíces, cuadrados, conversiones decimal — sexagesimal, cálculos en binarios/octales/hexadecimales, conversiones de coordenadas, π , números aleatorios, valores absolutos, enteros, fracciones.

Estadísticas: Desviación estándar — número de datos, sumatoria, sumatoria de cuadrados, media, desviación estándar (dos tipos).
Regresión lineal — número de datos, sumatoria de x , sumatoria de y , sumatoria del cuadrado de x , sumatoria del cuadrado de y , media de x , media de y , desviación estándar de x (dos tipos), desviación estándar de y (dos tipos), términos constantes, coeficiente de regresión, coeficiente de correlación, valor estimado de x , valor estimado de y .

Funciones especiales: Inserción, borrado, funciones de repetición, sustitución (=), instrucciones múltiples (: y \blacktriangle)

Memorias: Originalmente 26 (máximo 76), memoria Ans

Gama de cálculo: $\pm 1 \times 10^{-99} \sim \pm 9,999999999 \times 10^{99}$ y 0.
Operación interna con una mantisa de 12 dígitos.

Redondeo: En base al número de dígitos significativos o al número de dígitos decimales especificado.

Presentación exponencial: Norm 1 — $10^{-2} > |x|$, $|x| \geq 10^{10}$
Norm 2 — $10^{-9} > |x|$, $|x| \geq 10^{10}$

Función de programación

Número máximo de pasos: 400

Funciones de salto: Incondicionales (Goto), 10 máximo
Condicionales ($=$, \neq , $>$, $<$, \geq , \leq)
Salto con cuenta (Isz, Dsz)

Subrutinas: 9 niveles

Areas de programa: 10 máximo (P0 a P9)

Funciones de verificación: Verificación de programa, depuración, borrado, suma, inserción, etc.

Generalidades

Alimentación:	Dos pilas de litio CR2032
Consumo:	0,009 W
Duración de las pilas:	Unas 350 horas (CR2032)
Apagado automático:	Se activa unos 6 minutos después de la última operación.
Temperatura ambiente:	0°C a 40°C
Dimensiones:	9,9 × 73 × 141,5mm (Al/An/Pr)
Peso:	84 g con las pilas

Glosario

A

Ajuste del modo de reposición, 10
Alimentación, 8
Ampliación del número de memorias, 41
And, 30, 66
Antilogaritmo, 28, 56
Apagado automático, 10
Area de programa, 117
Aumentando las memorias, 41, 98

B

BASE-N, conversiones, 64
BASE-N, operaciones aritméticas, 65
BASE-N, operaciones lógicas, 66
BASE-N, valores negativos, 64
Binario, 30, 62, 64, 65
Borrado, 24, 38
Borrado completo (AC), 24
Borrado de memoria, 39
Borrado de programa, 21, 119

C

Cálculos aritméticos, 44, 65
Cálculos con memoria, 46
Cálculo de tiempo, 19, 28
Cálculos en el modo de BASE-N, 62
Cálculos estadísticos, 67
Cálculos estadísticos, una sola variable, 67

Cálculos estadísticos, variables en pares, 69
Cálculos manuales, 44, 172
Cambio de pila, 8
Compaginación, 38
Contraste, 30
Conversión de coordenadas, 58
Coordenadas polares, 25, 58
Coordenadas rectangulares, 25, 58
Corrección, 38
Coseno, 29, 55
Cursor, 23
Curva de distribución normal, 97, 98

D

Decimal, 33, 62, 64, 65
Desplazamiento de presentación de gráficos, 96
Desviación estándar, 67
Dígitos significantes, 16, 21, 47
Dígitos de entrada, 34
Dígitos de salida, 34
Disp, 16, 90
División, 25

E

Ejecución, 24
Ejemplos de gráficos, 96
Enfoque de detalles (x_f , $x \frac{1}{f}$, Org), 25, 84, 86, 90, 95
Enfoque de detalles, factor, 84, 86

Glosario

Especificaciones, 184
Estadísticas con variables en pares, 69
Estadísticas con una sola variable, 67
Estratos de registros, 32
Exponente, 18, 24, 34

F

Factor, 26, 84
Factorial, 59
Fijación de función alfabética, 22
Fix, 16, 21, 47
Formato de presentación, 18, 19, 21
Fracciones, 19, 29, 60
Función de repetición, 23, 51
Función de trazado, 26, 87
Funciones exponenciales, 27, 56
Funciones hiperbólicas, 16, 29, 57
Funciones hiperbólicas inversas, 16, 57
Funciones logarítmicas, 28, 56
Funciones trigonométricas, 29, 55
Funciones trigonométricas inversas, 29, 55
Funciones: Tipo A, 31
Funciones: Tipo B, 31

G

Gama, 26, 78, 80, 82
Gamas de ingreso, 181

Gradientes/grados centesimales, 16, 21, 22, 54, 55
Grados, 16, 21, 22, 54
Grados-minutos-segundos (GMS), 28
Gráfico de barras, 97, 98
Gráficos, 26, 76
Gráfico, gama, 26, 78
Gráficos de las funciones ingresadas manualmente, 82
Gráficos estadísticos con variables en pares, 100
Gráficos estadísticos con una sola variable, 97
Gráficos incorporados en las funciones científicas, 76
Gráfico, programa, 137
Gráficos, superposición, 77, 83

H

Hexadecimal, 19, 28, 29, 62, 64, 65

I

Inclusiones, 127
Ingeniería, 27
Inicializar, 10, 41, 82
Inserción, 39
Instrucciones múltiples, 27, 53

L

Línea, 93
ln, 28
Logaritmo común, 28, 56

Logaritmo natural, 28, 56
Lógica algebraica verdadera, 31
Lugares decimales, 16, 21, 47

M

Mandos de programa, 120
Mandos de salto, 120
Mantisa, 18, 34
Marcación de función de teclas, 14
Marcación de puntos, punto, 26, 91
Matemáticas para informática, 22, 62, 118
Medición angular, 22, 54
Memoria, borrado, 24, 40
Memoria restante, 22, 41, 98
Memorias de ordenaciones, 41, 131, 132
Memorias estadísticas, borrado, 29
Menos (-), 24
Mensajes de error, 32, 35, 52, 81, 99, 101, 113, 114, 127, 179
Mensajes textuales, 135
Modo de BASE-N, 22, 33, 118
Modo COMP, 22
Modo de cálculo, 21, 118
Modo de norma 1/norma 2, 18, 21
Modo LR, 22, 25, 33, 69, 100, 118
Modo PCL, 16, 21, 33, 119

Glosario

Modo RUN, 21, 33
Modo SD, 22, 25, 33, 67, 97, 118
Modo WRT, 16, 21, 33, 106, 117
Multiplicación, 25, 31

N

Neg, 29, 64
Negación, 27, 66
Not, 26, 66
Número aleatorio, 59

O

Octal, 62, 64, 65
Off, 24
On, 24
Operaciones lógicas, 66
Or, 30, 66

P

Paréntesis, 29, 45
Pasos, 17, 36, 106
Pasos de cálculo, 35
Pasos de programa, 17, 106, 116
Pasos de memoria, 36
Pi, 24, 55, 80
Potencia, 30
Presentación de texto, 37
Presentación gráfica, borrado, 26
Presentación de parámetro de gama, 78, 80
Presentación de posición de error, 52

Glosario

Presentación gráfica, 37
Presentación de texto, borrado,
24
Presentación exponencial, 18, 47
Programa, borrado, 21, 119
Programa, compaginación, 110
Programa, ejecución, 107
Programa, gráfico, 137
Programa, ingreso, 106
Programa, memoria, 106
Programación, 104

R

Radianes/rads, 16, 21, 22, 54
Raíz, 27, 59
Raíz cúbica, 30, 59
Recíproca, 28, 59
Regresión, 69, 100
Regresión de potencia, 69, 73
Regresión exponencial, 72
Regresión lineal, 70
Regresión logarítmica, 71
Reposición, 3, 10
Respuesta (Función Ans), 24, 49
Resta, 25
Rutina principal, 127

S

Saltos de cuenta, 124
Saltos condicionales, 122
Saltos incondicionales, 120
Sci, 16, 21
Scl, 29, 67, 69

Secuencia de prioridad de
cálculo, 31
Seno, 29, 55, 76
Sexagesimal, 19, 28, 55
Símbolo de puntuación,
(" , ~), 40, 135
Subrutina, 127
Suma, 25
Superación de capacidad, 35

T

Tangente, 29, 55
Tecla alfabética, 16, 22
Tecla de asignación, 29
Tecla CL, 30, 67, 69
Tecla de cambio, 16, 21
Tecla de cuadrado, 27, 59
Tecla de gráfico-texto (G-T), 26,
37
Tecla de modo, 21
Tecla de raíz cuadrada, 27, 61
Tecla DT, 30, 67, 69
Tecla numérica, 24
Tecla para enteros, 27, 60

V

Valores negativos, 24, 29, 64
Verificación de estado de
memoria, 22

X

xnor, 27, 66
xor, 27, 66
 $X \leftrightarrow Y$, 23, 89

MEMO