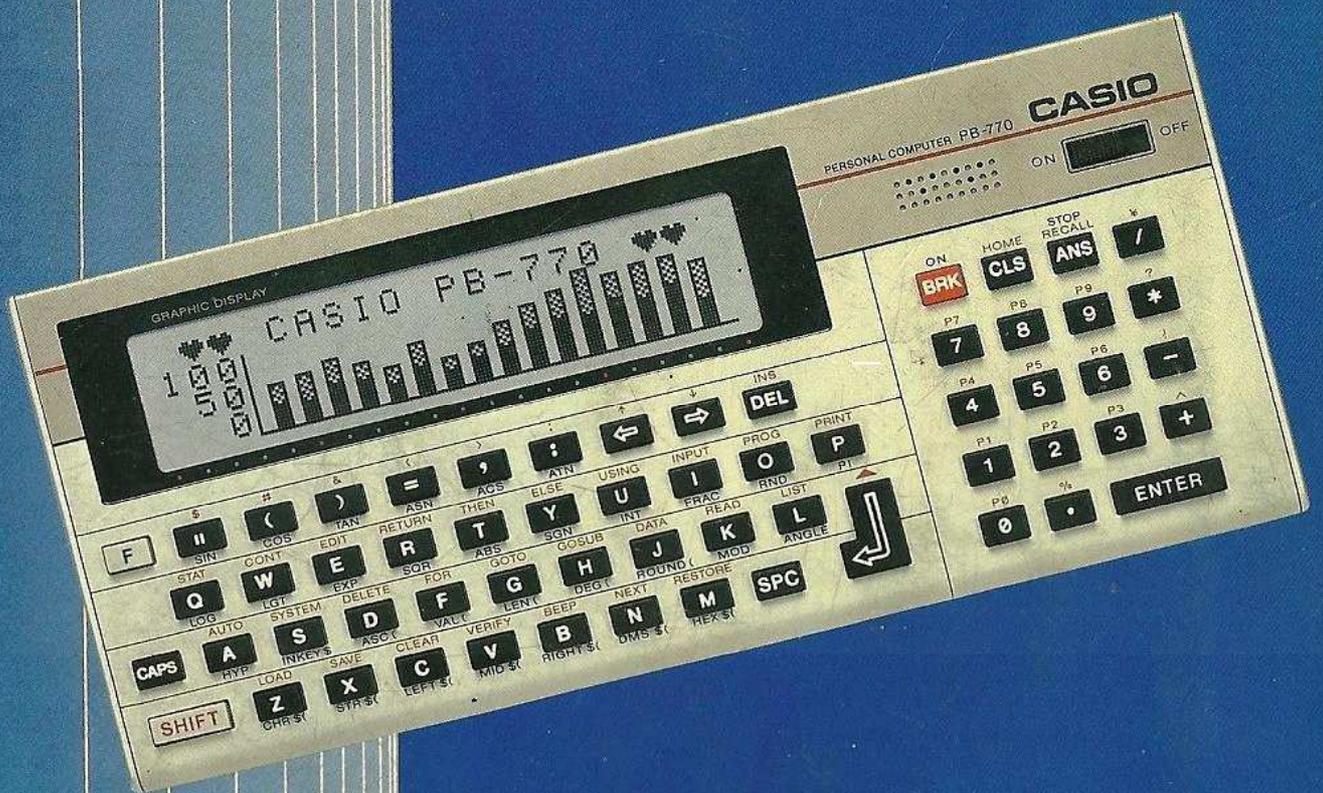


Computadora Personal

PB-770

MANUAL DEL PROPIETARIO



CASIO®

Computadora Personal

PB-770

MANUAL DEL PROPIETARIO

CASIO®

INDICE

CAPITULO 1 GUIA GENERAL

| | | |
|-----|---|----|
| 1-1 | ANTES DEL USO | 12 |
| 1-2 | CONFIGURACION Y CONEXIONES DEL SISTEMA | 13 |
| 1-3 | MANTENIMIENTO DE LAS PILAS | 15 |
| 1-4 | MODULO DE EXPANSION DE LA MEMORIA RAM (OPCIONAL) | 17 |
| 1-5 | NOMENCLATURA Y FUNCION DE CADA SECCION ... | 18 |
| 1-6 | PROGRAMA DE PRUEBA | 19 |

CAPITULO 2 OPERACION DEL TECLADO Y LA PANTALLA

| | | |
|-----|--|----|
| 2-1 | FUNCIONES DE LAS TECLAS EN EL MODO DIRECTO .. | 22 |
| 2-2 | FUNCIONES DE LAS TECLAS EN EL MODO SHIFT | 23 |
| 2-3 | TECLAS CAPS | 23 |
| 2-4 | TECLAS EN EL MODO PARA FUNCIONES | 24 |
| 2-5 | FUNCIONES DE LAS TECLAS ESPECIALES Y DE COMPAGINACION | 24 |
| 2-6 | FUNCIONES DE CALCULOS | 26 |
| 2-7 | VARIABLES | 29 |
| 2-8 | PANTALLA | 31 |
| 2-9 | NUMERO DE BYTES USADOS EN LAS VARIABLES ... | 32 |

CAPITULO 3 REFERENCIAS SOBRE "BASIC"

| | | |
|-----|--------------------------------------|----|
| 3-1 | INTRODUCCION AL LENGUAJE BASIC | 34 |
| 3-2 | USO DE LAS TECLAS | 35 |
| 3-3 | VARIABLES Y SU ASIGNACION | 37 |
| 3-4 | USO DE LAS VARIABLES | 39 |
| 3-5 | ENTRADA DE PROGRAMAS | 39 |
| 3-6 | PROGRAMACION EN BASIC [1] | 41 |

| | | |
|------|--|-----|
| 3-7 | PROGRAMACION EN BASIC [2] | 44 |
| 3-8 | EJECUCION DE PROGRAMAS | 50 |
| 3-9 | CONFIGURACION DE LA PANTALLA | 52 |
| 3-10 | OPERACIONES REITERATIVAS EN PROGRAMAS | 54 |
| 3-11 | PROGRAMA PARA LA SUMA DEL TOTAL | 57 |
| 3-12 | VARIABLES DE CARACTERES | 60 |
| 3-13 | ¿QUE ES UNA DIMENSION? | 62 |
| 3-14 | MATRICES DE VARIABLES NUMERICAS | 65 |
| 3-15 | PROGRAMACION CON MATRICES NUMERICAS | 70 |
| 3-16 | VARIABLES MATRICIALES DE CARACTERES | 77 |
| 3-17 | COMBINACION DE MATRICES NUMERICAS Y DE CARACTERES | 81 |
| 3-18 | FUNCIONES ESTADISTICAS | 85 |
| 3-19 | USO DE LOS CARACTERES GRAFICOS | 91 |
| 3-20 | VISUALIZACION DE PATRONES GRAFICOS | 93 |
| 3-21 | FUNCIONES GRAFICAS DE LA PB-770 | 98 |
| 3-22 | MANDOS GRAFICOS Y COORDENADAS DE LA PANTALLA | 99 |
| 3-23 | TRAZADO DE CURVAS | 104 |
| 3-24 | TRAZADO DE GRAFICOS LINEALES | 106 |
| 3-25 | PREPARACIONES PARA EL TRAZADO DE GRAFICOS DE BARRAS | 108 |
| 3-26 | DOS EJEMPLOS DE PROGRAMAS PARA GRAFICOS DE BARRAS | 110 |
| 3-27 | GRAFICOS EN MOVIMIENTO | 113 |
| 3-28 | JUEGOS (TENNIS) | 117 |
| 3-29 | TRAZADO DE GRAFICOS EN LA IMPRESORA POR COORDENADAS | 120 |
| 3-30 | USO DE LA IMPRESORA POR COORDENADAS | 123 |
| 3-31 | USO DE LOS PROGRAMAS PARA LA PB-700 | 125 |
| | Prefacio al Capítulo 4 | 126 |

CAPITULO 4 GUIA DE MANDOS

| | | |
|-----|----------------------------|-----|
| 4-1 | MANDOS MANUALES | 128 |
| | AUTO | 128 |
| | CONT | 129 |
| | DELETE | 130 |
| | EDIT | 132 |
| | LIST/LLIST | 135 |
| | LOAD | 138 |
| | NEW/NEW ALL | 142 |
| | PASS | 143 |
| | PROG | 145 |
| | RUN | 146 |
| | SAVE | 147 |
| | SYSTEM | 150 |
| | VERIFY | 152 |
| 4-2 | MANDOS DE PROGRAMA | 153 |
| | ANGLE | 153 |
| | BEEP | 154 |
| | CHAIN | 155 |
| | CLEAR | 157 |
| | CLS | 161 |
| | DIM | 162 |
| | DRAW/DRAWC | 167 |
| | END | 170 |
| | ERASE | 171 |
| | FOR ~ TO ~ STEP/NEXT | 172 |
| | GET | 177 |
| | GOSUB/RETURN | 180 |
| | GOTO | 184 |
| | IF ~ THEN ~ ELSE | 186 |

| | | |
|-----|-------------------------------|-----|
| | INPUT | 189 |
| | LET | 195 |
| | LOCATE | 196 |
| | POKE | 197 |
| | PRINT/LPRINT | 198 |
| | PUT | 203 |
| | READ/DATA/RESTORE | 205 |
| | REM | 209 |
| | STOP | 210 |
| | TRON/TROFF | 212 |
| 4-3 | FUNCIONES NUMERICAS | 214 |
| | SIN | 214 |
| | COS | 217 |
| | TAN | 218 |
| | ASN, ACS, ATN | 219 |
| | HYP SIN/HYP COS/HYP TAN | 221 |
| | HYP ASN/HYP ACS/HYP ATN | 221 |
| | SQR | 222 |
| | LOG, LGT | 223 |
| | EXP | 226 |
| | ABS | 228 |
| | INT | 230 |
| | FRAC | 232 |
| | SGN | 234 |
| | ROUND | 236 |
| | PI | 238 |
| | RND | 239 |
| | DEG | 241 |
| | PEEK | 242 |

INDICE

| | | |
|-----|-------------------------------|-----|
| 4-4 | FUNCIONES DE CARACTERES | 243 |
| | ASC | 243 |
| | CHR\$ | 245 |
| | VAL | 247 |
| | STR\$ | 250 |
| | LEFT\$ | 252 |
| | RIGHT\$ | 253 |
| | MIDS | 254 |
| | LEN | 256 |
| | INKEY\$ | 257 |
| | DMSS | 259 |
| | HEXS | 260 |
| 4-5 | FUNCIONES DE VISUALIZACION | 261 |
| | TAB | 261 |
| | USING | 263 |
| | POINT | 266 |
| 4-6 | MANDOS/FUNCIONES ESTADISTICAS | 268 |
| | STAT | 268 |
| | STAT CLEAR | 269 |
| | STAT LIST/STAT LLIST | 269 |
| | CNT | 270 |
| | COR | 270 |
| | SUMX/SUMY/SUMX2/SUMY2/SUMXY | 271 |
| | MEANX/MEANY | 272 |
| | SDX/SDY/SDXN/SDYN | 273 |
| | EOX/EOY | 274 |
| | LRA/LRB | 274 |
| 4-7 | MISCALEANOS | 275 |
| | &H | 275 |

CAPITULO 5 LISTA DE PROGRAMAS

| | |
|---|-----|
| ADMINISTRACION DE ACCIONES Y PRECIOS ADECUADOS DE COMPRA/VENTA | 278 |
| DIRECTORIO TELEFONICO | 287 |
| TOTAL CRUZADO | 294 |
| PROGRAMA PARA TRAZAR GRAFICAS | 303 |

CAPITULO 6 MATERIAL DE REFERENCIA

| | |
|---|-----|
| 6-1 TABLA DE MANDOS DE LA UNIDAD PB-770 | 312 |
| 6-2 TABLE DE MENSAJES DE ERROR | 324 |
| 6-3 TABLA DE CODIGOS DE CARACTERES | 328 |
| ESPECIFICACIONES | 329 |

GENERALIDADES

La PB-770 es una computadora personal que posee una gran cantidad de características de suma utilidad. La comprensión de dichas características no puede materializarse de un día para el otro.

Seguir cada procedimiento paso por paso es la mejor manera de familiarizarse con cada función. Vd. irá mejorando en proporción directa al tiempo que dedique al uso de su computadora.

Bajo este concepto, este manual muestra lo que es una computadora personal de manera muy sencilla, con ejemplos prácticos del lenguaje de programación BASIC y del manejo del teclado.

Primeramente, deben tenerse en cuenta ciertas precauciones que se tomarán durante el uso de la PB-770. El Capítulo 1 describe sus características y los procedimientos básicos de uso. El Capítulo 2 cubre lo que sirve para almacenar información y hacer cálculos. El Capítulo 3 cubre lo que sirve para almacenar información y hacer cálculos. El Capítulo 3 explica cómo se puede preparar un programa en BASIC para el almacenamiento y recuperación de información en grandes cantidades dentro de la PB-770, mediante el método denominado "programación con matrices". También enseña al novato cómo programar gráficos en la pantalla de gran tamaño de la PB-770 y cómo hacer uso de la impresora con interface para cassette.

El Capítulo 4 es para quienes ya conocen el lenguaje BASIC. En él se explican detalladamente el uso de la gran cantidad de mandos y funciones disponibles en la PB-770. Para la principiante, no es recomendable comenzar por este capítulo, ya que es necesario que aprenda primeramente la estructura de este lenguaje de programación.

El Capítulo 5 presenta programas prácticos para usar en la PB-770. De fácil aplicación, estos programas pueden modificarse sencillamente para satisfacer diferentes usos.

CAPITULO 1

GUIA GENERAL

1-1 ANTES DEL USO

La PB-770 es el resultado de la avanzada tecnología electrónica y los estrictos procesos de prueba y control de calidad de CASIO.

Para asegurar una larga vida y evitar fallas rogamos tenga en cuenta las siguientes consideraciones.

■ Importante

- Esta computadora se compone de piezas electrónicas de alta precisión. No intenta desarmarla o repararla por su cuenta. Evite que se doble o se caiga. No la lleve en el bolsillo de un pantalón.
- Use solamente la impresora opcional con interface para grabadora de cassette FA-10, la impresora con grabadora de cassette e interface para grabadora de cassette FA-11 o la interface para impresora (CENTRONICS universal) con interface para grabadora de cassette. No conecte nunca otro equipo periférico al conector de esta unidad.
- Evite el uso en ambientes con temperaturas extremas. No la guarde en automóviles, cerca de calefactores u otros sitios donde quede expuesta a temperaturas altas. Evite también sitios donde haya mucha humedad o polvo. La respuesta de la pantalla puede mermar o desvanecerse completamente en temperaturas muy bajas. Ello no significa que la unidad está averiada.
- Limpie la unidad frotándola con una paño suave, seco o levemente humedecido con detergente neutral. Nunca use líquidos diluyentes.
- Antes de solicitar la reparación de su unidad al personal técnico especializado, verifique la alimentación y asegúrese de que no haya cometido errores de programación.
- En caso de que su unidad deba repararse, sírvase dirigirse a su distribuidor más cercano.

1-2 CONFIGURACION Y CONEXIONES DEL SISTEMA

FA-10: Impresora con interface para grabadora de cassette

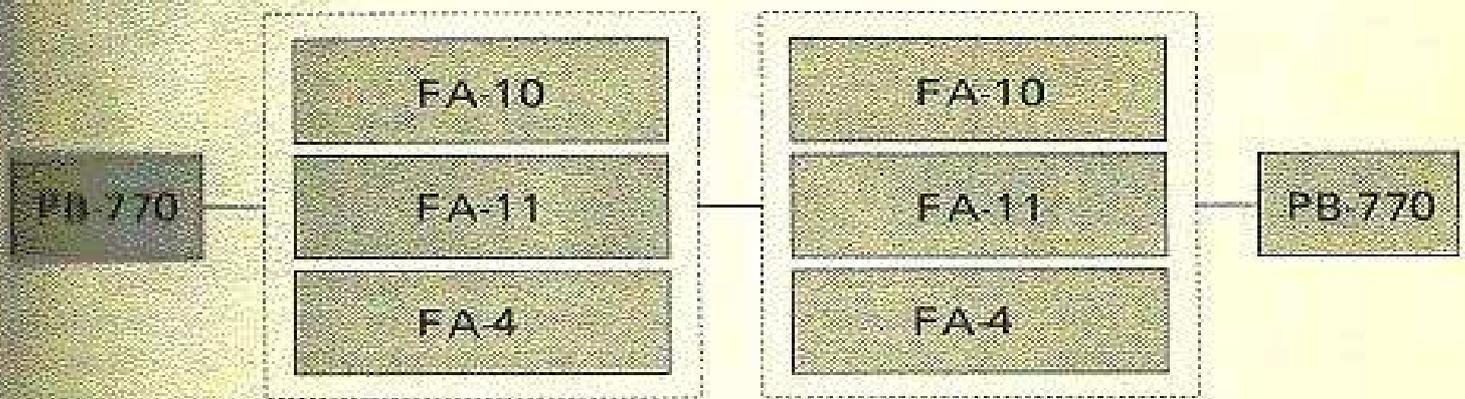
CM-1: Grabadora de microcassette

FA-11: Impresora con grabadora de cassette común e interface para grabadora de cassette

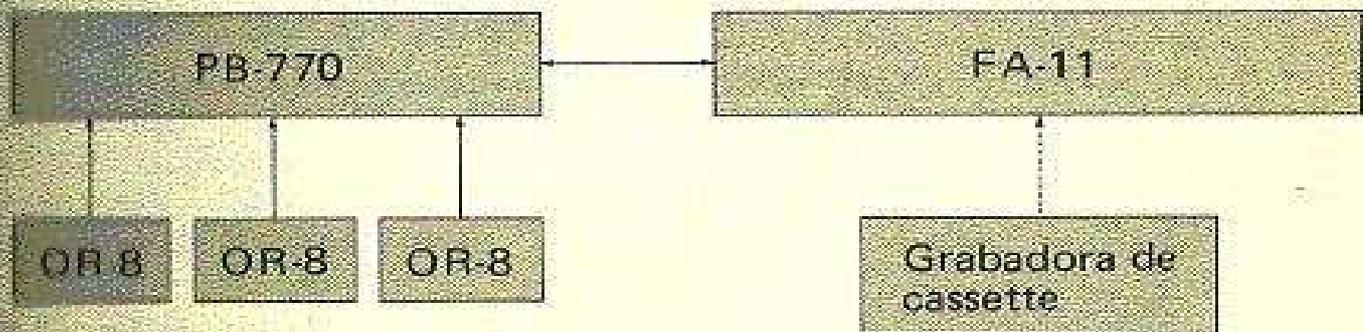
FA-4: Interfaces para impresora (CENTRONICS universal) y para grabadora de cassette

OR-8: Módulo de expansión de la RAM (8k bytes)

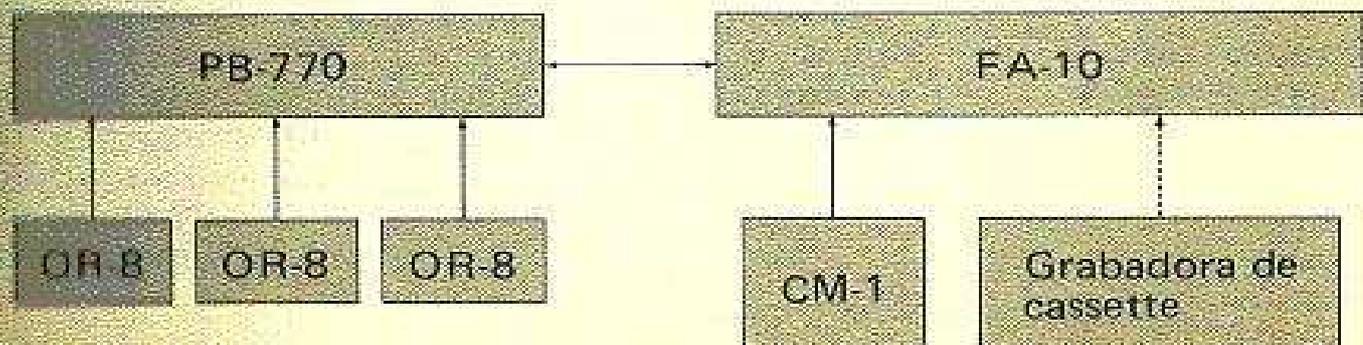
• Conexión con otra PB-770 (para transferencia de datos/programas)



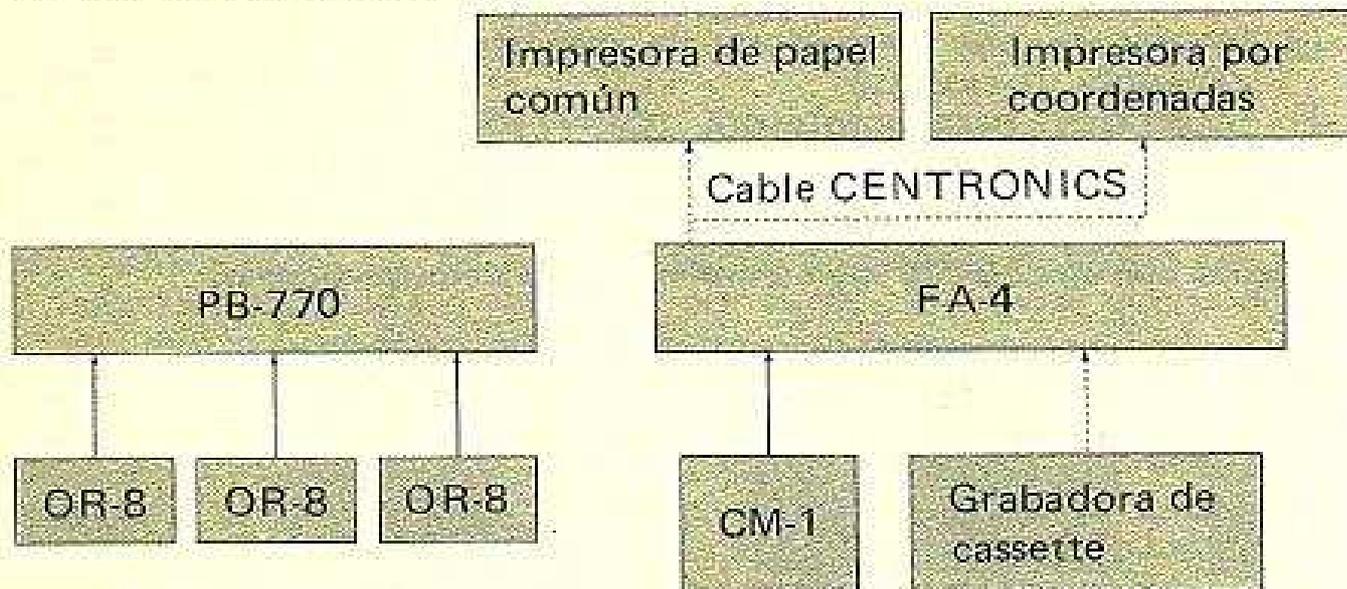
• Conexión de una unidad FA-11



• Conexión de una unidad FA-10



■ Conexión de una unidad FA-4



1-3 MANTENIMIENTO DE LAS PILAS

■ Colocación de las pilas

Apague la PB-770. Déla vuelta y deslice la tapa del compartimiento para las pilas para abrirlo (Figura 1). Coloque cuatro pilas tamaño AA. Tenga en cuenta su polaridad, ya que las pilas pueden sulfatarse si se las coloca incorrectamente. Colóquelas de tal modo que el negativo (-) quede del lado de los resortes que se encuentra en el compartimiento para las pilas (Figura 2).

El uso de pilas nuevas con otras ya agotadas hará que la duración de las mismas sea más corta que normalmente. Por lo tanto, cambie todas las pilas de una vez.

La unidad podrá dañarse y funcionar incorrectamente si las pilas se sulfatan. Siempre que deje de usar la unidad por un tiempo prolongado, quite las pilas de la misma.

■ Cambio de las pilas

Cambie las pilas siempre que merme la intensidad del sonido emitido por el mando BEEP, o siempre que se apague la pantalla (vémitase a la página 328, para las especificaciones de las baterías).

■ Configuración de la alimentación

La PB-770 dispone de dos fuentes de alimentación: una principal y otra secundaria para proteger el contenido de la memoria RAM (Figura 3). La información y programas almacenados quedan así protegidos al cambiarse una u otra de las pilas de alimentación. La información sólo se pierde si se quitan o averían ambas fuentes en forma simultánea.

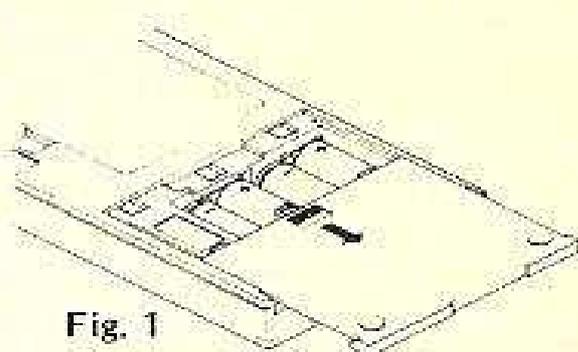


Fig. 1

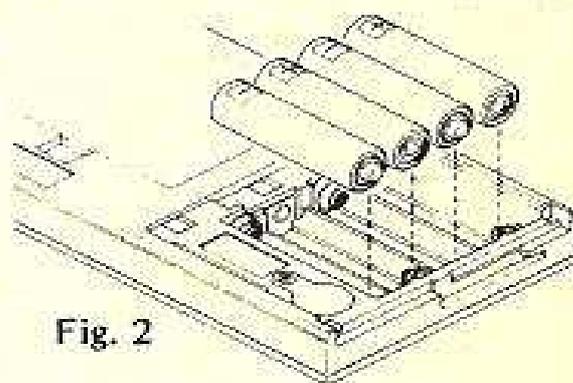


Fig. 2

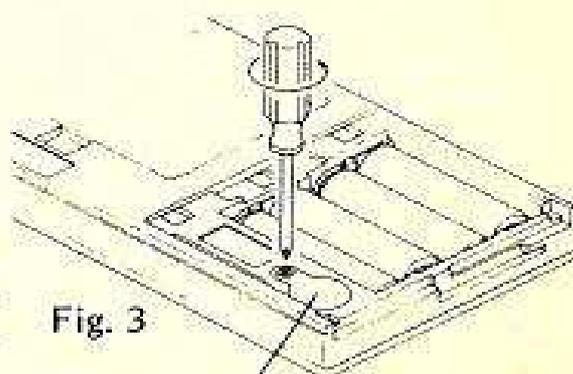


Fig. 3

Pila para protección de la memoria RAM

- * Para evitar que la unidad se dañe por la sulfatación de las pilas, cambie sin falta las pilas principales y la de protección de la RAM cada dos años.
- * Aun si se quitan las pilas principales, las secundarias protegen la memoria RAM por unos 4 meses cuando su capacidad es de 8KB y por unos 1 meses cuando su capacidad es de 32KB.
- * Recuerde entrar NEW ALL  siempre que cambie cualquiera de las dos pilas.
- * **Guarde las pilas fuera del alcance de los niños. En caso de ingestión, consulte inmediatamente a un médico.**

■ Apagado automático

Para conservar la carga de las pilas, la unidad se apaga automáticamente unos 6 minutos después de la última operación del teclado (excepto durante la ejecución de un programa). La unidad puede volver a encenderse presionando la tecla  o apagándola una vez y encendiéndola nuevamente.

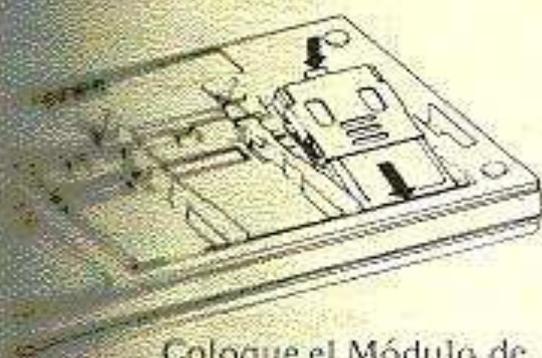
■ Función de detección de baja tensión

La función de detección de baja tensión de la PB-770 protege el contenido de la RAM ante casos en que la tensión de las pilas principales caiga por debajo de determinado nivel.

Cuando la tensión cae por debajo de dicho nivel, se oscurece toda la pantalla o, en el caso de la ejecución de un programa, lo visualizado en la pantalla comienza a parpadear, y la unidad deja de funcionar. En tal caso, deben cambiarse las pilas inmediatamente. El contenido de la RAM puede cambiar si se enciende la unidad con las pilas agotadas.

1-4 MODULO DE EXPANSION DE LA MEMORIA RAM (OPCIONAL)

Figura 1. Apague la unidad.



Coloque el Módulo de Expansión RAM del modo arriba ilustrado.

Figura 2. Deslice el soporte en la dirección que indica la flecha.

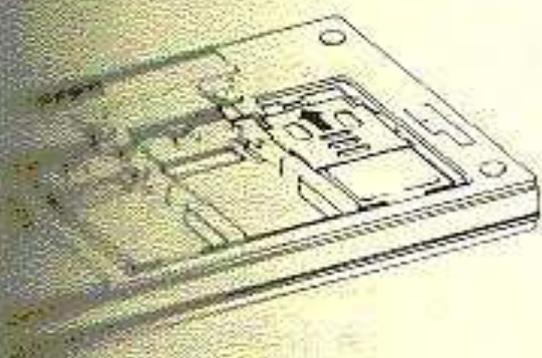
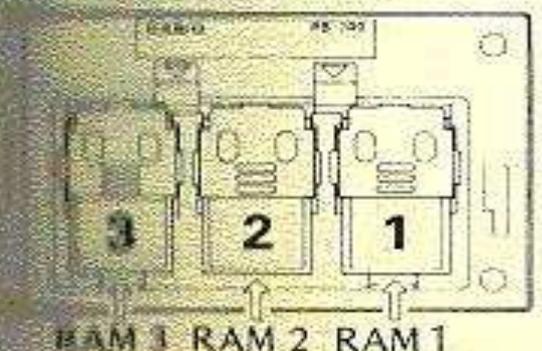


Figura 3. Orden de instalación de Módulos de Expansión RAM múltiples.



La PB-770 viene originalmente con 8K bytes de RAM. Esta capacidad puede ampliarse hasta un máximo de 32K bytes mediante la instalación de módulos de expansión RAM modelo OR-8.

Por cada módulo que se instala, se agregan 8K bytes a dicha memoria. El procedimiento de instalación es el siguiente:

- (1) Apague la PB-770.
- (2) Dé vuelta la PB-770 y quite la tapa del compartimiento para los módulos RAM presionando los dos retenes.
- (3) Tome el módulo OR-8 por sus laterales y colóquelo en la primera posición de la derecha que esté desocupada (Figura 1). Tenga cuidado de no tocar los terminales del módulo, ya que es sumamente sensible a la electricidad estática.
- (4) Haga levemente presión sobre el soporte y deslícelo hasta la posición donde queda trabado (Figura 2).
- (5) Instale el número de módulos RAM que necesite. Luego, vuelva a colocar la tapa del compartimiento en su sitio original y encienda la PB-770.
- (6) Entre NEW ALL . Verifique la capacidad de la memoria RAM entrando   .
- (7) Instale los Módulos de Expansión RAM siguiendo el orden ilustrado en la Figura 3. La PB-770 no funcionará correctamente si se instala un Módulo RAM a la izquierda de un compartimiento vacío.

1-5 NOMENCLATURA Y FUNCION DE CADA SECCION

- **Control de contraste** Ajustelo a gusto personal. de la pantalla
- **Teclas alfabéticas** Se utilizan para visualizar letras mayúsculas. También se pueden visualizar letras minúsculas, presionando la tecla que corresponda a la letra simultáneamente con la tecla **[CAPS]**.
Los mandos y símbolos impresos por encima de cada tecla pueden entrarse pulsando la tecla pertinente mientras se presiona la tecla **[SHIFT]** (mandos de una tecla). Las funciones impresas debajo de cada tecla pueden visualizarse pulsando la tecla correspondiente mientras se presiona la tecla **[F]** (funciones de una tecla).

Ejemplo.

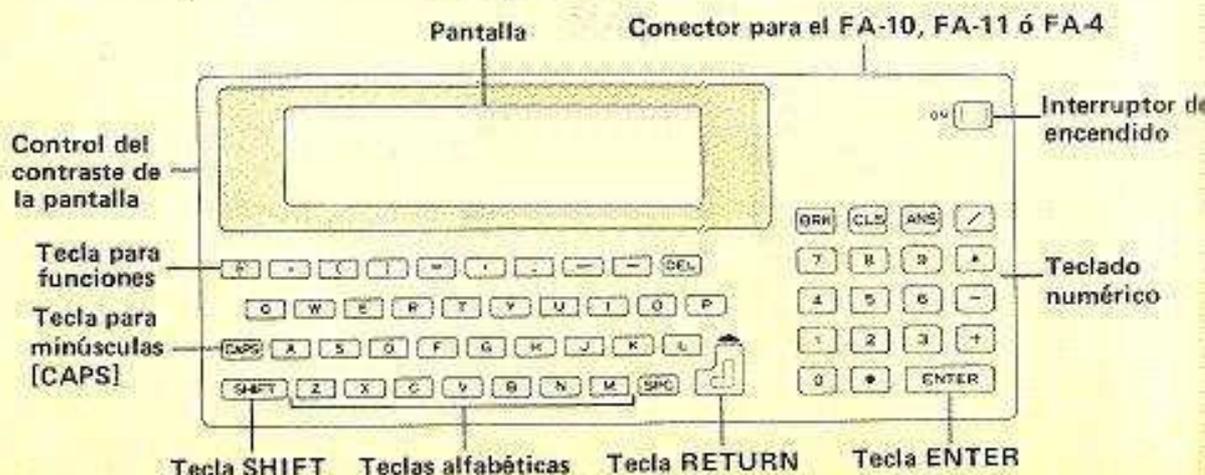
SYSTEM Modo SHIFT
[S] Modo directo
 INKEY\$ Modo de funciones

* Para visualizar símbolos y caracteres que no pueden entrarse directamente desde el teclado, haga uso de la función CHR\$ (ver la página 245).

- **Tecla RETURN** () . . . Se utiliza para ejecutar mandos y para la entrada de programas en BASIC. Ejecuta también la entrada y salida de datos.
- **Tecla ENTER** (**[ENTER]**) Se utiliza para ejecutar instrucciones en cálculos manuales.

Ejemplo: **[1] [2] [3] [4] [ENTER]** → 3

- En el apartado 2-5, "Funciones de las Teclas Especiales y de Compaginación", se describe en detalle el uso de cada tecla.



16 PROGRAMA DE PRUEBA

Si aquí un programa de demostración de las funciones de la PB-770, tiene problemas con la entrada del programa, revise los Capítulos 15 y 16.

Por medio de este programa, podrá ver varias funciones de control de la pantalla, el funcionamiento del mando BEEP y las velocidades de ejecución de los mandos para gráficos.

```
10 CLS
20 LOCATE 3,1:PRINT "PB-770 TESTING"
30 FOR A=31 TO 0 STEP -1
40 DRAW(0,A)-(159,A)
50 NEXT A
60 LOCATE 5,1:PRINT "BEEP SOUND"
70 FOR A=0 TO 9
80 BEEP 1:BEEP 1:BEEP 0
90 NEXT A
100 CLS
110 FOR A=33 TO 255
120 PRINT CHR$(A):
130 NEXT A
140 FOR A=0 TO 20
150 NEXT A
160 CLS
170 FOR A=1 TO 16
180 DRAW(A+2,16-A)-(A*3,16-A)-(A*3,15+
    A)-(A+2,15+A)-(A+2,16-A)
190 NEXT A
200 FOR A=0 TO 200
210 NEXT A
```


CAPITULO 2

OPERACION DEL TECLADO Y LA PANTALLA

2-1 FUNCIONES DE LAS TECLAS EN EL MODO DIRECTO

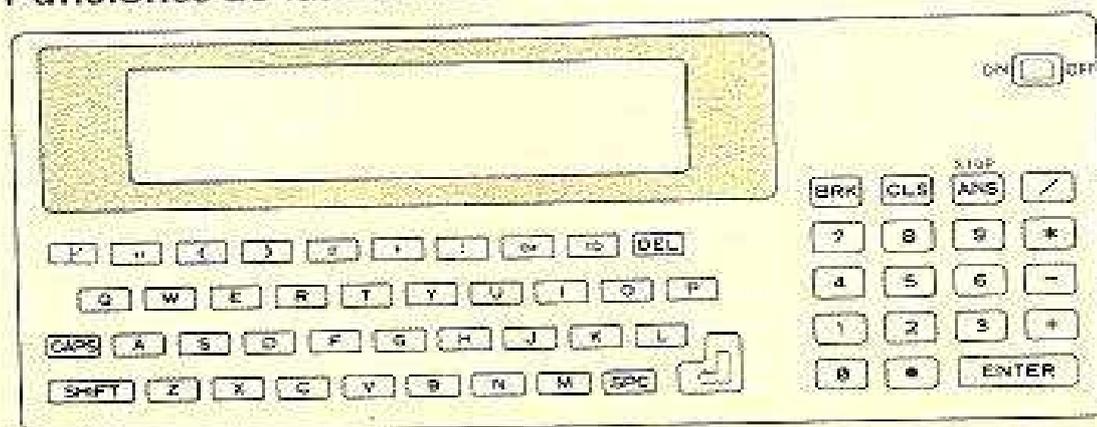
Siempre que se pulse cualquier tecla en el modo directo, aparece en la pantalla el carácter o se da entrada la función rotulada sobre la tecla misma.

| | |
|---|--|
| A - Z | Letras mayúsculas |
| SPC | Espacio (en blanco) |
| " , = , (,) , * , : | Símbolos |
| 0 - 9 | Números |
| . | Punto (y no coma) decimal |
| + , - , * , / | Símbolos aritméticos |
| CLS | Borrado de la pantalla |
| DEL | Supresión de un carácter |
| BRK | Interrupción de la ejecución |
| ENTER | Ejecución de cálculos manuales |
| ↵ | Entrada y ejecución de programas |
| ← , → | Deplazamiento del "cursor" (izquierda/derecha) |
| SHIFT | Designación del modo SHIFT |
| CAPS | Designación del modo CAPS |
| F | Designación del modo de funciones |
| ANS | Visualización del resultado del último cálculo |

■ Número de caracteres en una sentencia

79 es el número máximo de caracteres que pueden entrarse en una fórmula para cálculos manuales o en una línea durante la programación en BASIC.

Funciones de las teclas en el modo DIRECTO



9.2 FUNCIONES DE LAS TECLAS EN EL MODO SHIFT

Cuando se presiona la tecla **SHIFT** junto con cualquier otra tecla, se da entrada a los caracteres, símbolos y mandos rotulados en marrón por encima de cada tecla. Hace posible el uso de 26 diferentes mandos de una tecla.

SHIFT, **(|)**, **(<)**, **(>)**, **(\$)**, **(;)**, **(¥)**, **(?)**, **(!)**, **(^)**, **(%)** Símbolos

SHIFT Inserción de caracteres y símbolos

SHIFT **(P)** Designación del área de programa

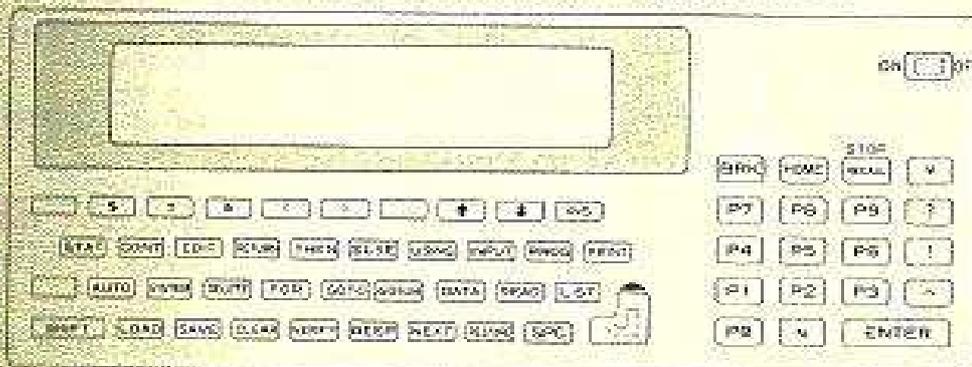
SHIFT Visualización de la línea anterior en el modo EDIT (página 132).

SHIFT, **(↑)** Desplazamiento del cursor (arriba/abajo)

SHIFT Desplazamiento del cursor (al comienzo de la sentencia)

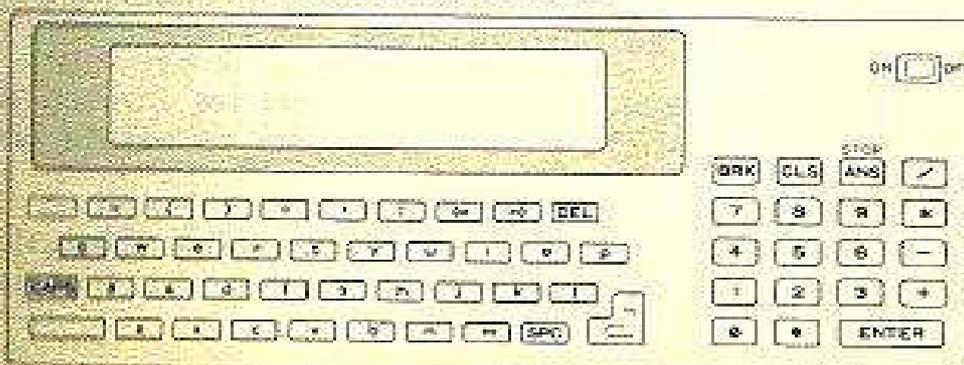
SHIFT Función de recuperación

Funciones de las teclas en el modo SHIFT



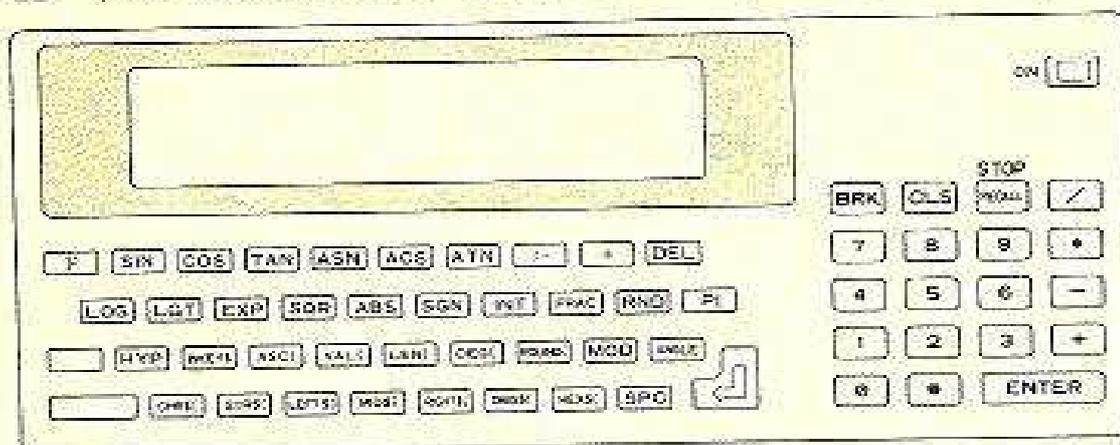
9.3 TECLAS CAPS

Cuando cualquier tecla junto con la tecla **CAPS**, se puede entrar la letra correspondiente a dicha tecla.



2-4 PARA FUNCIONES

Cuando se pulsa cualquier tecla al mismo tiempo que se está presionando la tecla **F**, se da entrada a la función impresa por debajo de la tecla pulsada.



2-5 FUNCIONES DE LAS TECLAS ESPECIALES Y DE COMPAGINACION

- (1) **SPC** (SPACE) Espacio en todos los modos.
- (2) **HOME** Borra la pantalla y lleva el cursor al extremo superior izquierdo de la misma.
CLS
La entrada **SHIFT HOME** lleva el cursor al comienzo de la sentencia sin alterar lo visualizado.
- (3) **INS** Borra el carácter o símbolo en la posición del cursor y desplaza hacia la izquierda los caracteres o símbolos a la derecha del cursor.
DEL
SHIFT INS abre un espacio desplazando hacia la derecha los caracteres o símbolos a la derecha del cursor.
Siempre que se mantienen pulsadas las teclas **DEL** o **SHIFT DEL** la función de repetición permite borrar caracteres o intercalar espacios, respectivamente, en forma continua.
- (4) **ON** Interrumpe la ejecución del cálculo o programa. Se utiliza también para encender la unidad después de activada la función de apagado automático.
BRK
- (5) **↑**, **↓** Sirven para desplazar el cursor en las cuatro direcciones posibles. La función de repetición es válida sólo para el desplazamiento lateral.
←, **→**
- (6) **ENTER** Para la ejecución de cálculos manuales. Cumple la misma función que la tecla **↵** durante la condición de espera de

entrada desde el teclado por la sentencia INPUT. Las sentencias comunes y mandos/sentencias de asignación no pueden ejecutarse por medio de esta tecla.

Sirve para almacenar el resultado del cálculo manual previo. Almacena también el valor numérico visualizado por la sentencia PRINT o impreso por la sentencia LPRINT.

Ejemplo: $3.4 * 5$ [ENTER] → 17
 $5.8 * 3 -$ [ANS] [ENTER] → 0.4

La pulsación de [SHIFT] [RECALL] activa una función de recuperación, la cual visualiza ya sea la última fórmula ejecutada mediante la tecla [ENTER] o la última sentencia de programa entrada usando la tecla [↵].

Ejemplo: $100 * 5$ [ENTER] → 500
 [SHIFT] [RECALL] → $100 * 5$

La pulsación de [STOP] [ABS] suspende la ejecución de un programa. La misma puede reanudarse mediante el mando CONT.

Sirve para la entrada de información durante la ejecución de un programa o para entrar mandos del mismo. No funciona durante los cálculos manuales.

En el modo de compaginación (EDIT), la secuencia [SHIFT] [↵] visualiza la línea de programa anterior (ver la página 132).

[SHIFT] [P9] → [SHIFT] [P9] . . . Sirve para especificar el área de programa y ejecutarlo a partir de su línea inicial.

[SHIFT] . . . (SHIFT) La pulsación de esta tecla especifica el modo de mandos de una tecla. La unidad vuelve automáticamente al modo directo al soltarse esta tecla.

[SHIFT] . . . (Letras minúsculas) La pulsación de esta tecla especifica el modo "CAPS". La unidad vuelve automáticamente al modo directo al soltarse esta tecla.

[SHIFT] . . . (Funciones) La pulsación de esta tecla especifica el modo de funciones. La unidad vuelve automáticamente al modo directo al soltarse esta tecla.

[SHIFT] . . . La pulsación de esta tecla permite visualizar en la pantalla la línea en cual hay un error (con el fin de corregir la misma) inmediatamente después que haya ocurrido un error durante la ejecución del programa en cuestión.

2-6 FUNCIONES DE CALCULOS

■ Precisión y funciones de cálculo

Los cálculos internos se llevan a cabo en su totalidad con una mantisa de 12 dígitos (más 2 dígitos para el exponente). El empleo de operaciones decimales permite llevar a cabo cálculos de gran precisión.

Los cálculos manuales se ejecutan mediante la tecla **ENTER**. Los resultados aparecen en la pantalla con una mantisa de 10 dígitos (más 2 dígitos para el exponente). Por tal motivo, el undécimo dígito de la mantisa se redondea.

■ Operadores aritméticos

| | |
|-----------|---|
| \wedge | Potenciación |
| $+$, $-$ | Suma, resta |
| $*$, $/$ | Multiplicación, división |
| MOD | Cálculos con resto (descarta la parte fraccionaria de una operación siempre que la haya). |

La gama de cálculo es la siguiente:

- (1) Toda división por 0 (cero) genera un error MA.
- (2) Siempre que el resultado excede la gama de cálculo, aparece un error OV.
- (3) Límites para las potencias

$$0 \wedge 0 \longrightarrow \text{Error MA}$$

$$(\pm x) \wedge 0 \longrightarrow 1$$

$$0 \wedge y \longrightarrow 0$$

$$0 \wedge (-y) \longrightarrow \text{Error MA}$$

$$(-x) \wedge (\pm y) \longrightarrow \text{Se acepta sólo cuando } y \text{ es un entero. De lo contrario, resulta en un error MA.}$$

* Donde $x > 0$, $y > 0$.

■ Prioridad de cálculo

Los cálculos se ejecutan bajo la siguiente prioridad:

1. Operaciones entre paréntesis
2. Funciones
3. Potencias (\wedge)
4. Signos positivo (+) y negativo (-)
5. Multiplicación y división
6. MOD
7. Suma y resta

Ejemplos

Ejemplo

| | | |
|---|------------------------|--------------------------------|
| 1 | $\frac{X+Y}{2}$ | $\rightarrow (X+Y)/2$ |
| 2 | $X^2+2XY+Y^2$ | $\rightarrow X^2+2*X*Y+Y^2$ |
| 3 | $-Y^2$ | $\rightarrow -Y^2$ |
| 4 | $(-Y)^2$ | $\rightarrow (-Y)^2$ |
| 5 | $(X^Y)^2$ | $\rightarrow X^Y^2$ |
| 6 | X^{Y^2} | $\rightarrow X^{(Y^2)}$ |
| 7 | Resto de $\frac{X}{Y}$ | $\rightarrow X \text{ MOD } Y$ |

Ejemplos

| | | |
|----|-----------------------|-------------------------------|
| 1 | 0.5^0 | $\rightarrow 1$ |
| 2 | -0.5^0 | $\rightarrow -1$ |
| 3 | $(-0.5)^0$ | $\rightarrow 1$ |
| 4 | 0.5^2 | $\rightarrow 0.25$ |
| 5 | 0.5^{-2} | $\rightarrow 4$ |
| 6 | $(-0.5)^{-2}$ | $\rightarrow 4$ |
| 7 | $0.5^{0.5}$ | $\rightarrow 0.7071067812$ |
| 8 | $(-0.5)^{0.5}$ | $\rightarrow \text{Error MA}$ |
| 9 | $2^{-0.5}$ | $\rightarrow 0.7071067812$ |
| 10 | $(-2)^{-0.5}$ | $\rightarrow \text{Error MA}$ |
| 11 | $10 \text{ MOD } 6$ | $\rightarrow 4$ |
| 12 | $-10 \text{ MOD } 6$ | $\rightarrow -4$ |
| 13 | $10 \text{ MOD } -6$ | $\rightarrow 4$ |
| 14 | $-10 \text{ MOD } -6$ | $\rightarrow -4$ |

■ Operadores comparativos

Los operadores comparativos pueden usarse sólo en la sentencia IF (página 186).

| | |
|--------|-------------------|
| = | Igual |
| <>, >< | Desigual |
| < | Menor que |
| > | Mayor que |
| =>, >= | Mayor o igual que |
| =<, <= | Menor o igual que |

Ejemplos: $A + B <> 0$. . . El resultado de $A + B$ no es igual a 0 (cero).
 $A\$ <> "Y"$. . . El contenido de $A\$$ no es igual a "Y".
 $A\$ = CHR\$(84) + CHR\$(79) + CHR\(77) . . . $A\$$ es igual a "TOM".
 $CHR\$(67) > CHR\(N) . . . $CHR\$(67)$, el cual corresponde a la letra G, es mayor que $CHR\$(N)$ según la Tabla de Códigos para Caracteres (página 328).

■ Operaciones con variables

El contenido de las variables puede verificarse mediante el use de la tecla **ENTER**.

Ejemplo: $A \text{ [ENTER]} \rightarrow 0$

Se describen a continuación los dos tipos de variables numéricas disponibles, según su grado de precisión:

Precisión simple: Se descarta todo a partir del décimotercer dígito de la mantisa hacia la derecha (12 dígitos). La precisión simple es lo normal en los cálculos.

Media precisión: Se descarta en la mantisa del sexto dígito en adelante (5 cifras). Se define especificando variables de matriz (!). Puede especificarse sólo en variables de matriz (Programación con Matrices). (Página 65).

Cómputos de precisión media

El almacenamiento de valores de precisión media requieren tan sólo 4 bytes en contraste con los 8 bytes necesarios para la precisión simple. En la mayoría de los casos, excepto en cálculos de ingeniería o científicos, 5 dígitos son suficientes. Si el límite de 5 dígitos es suficiente para sus cálculos (resultados de exámenes, porcentajes, número de artículos, precios, cantidades, etc.), se ahorra la mitad de la RAM que, en cambio, sería necesaria si se utilizara precisión simple.

2-7 VARIABLES

■ Tipos de variables

En la PB-770 se pueden usar los siguientes tipos de variables:

(1) Numéricas

Variables fijas (hasta 12 dígitos)

Variables registradas (hasta 12 dígitos)

Variables de matriz (5 dígitos con precisión media y 12 dígitos con precisión simple).

* Las cifras mencionadas corresponden al número de dígitos para los cálculos internos.

(2) Variables de caracteres

Variables de caracteres fijas (hasta 7 caracteres)

Variables de caracteres registradas (hasta 16 caracteres)

Variables de caracteres matriciales. Se pueden especificar de 1 hasta 79 caracteres. Si no se especifica nada, se toman automáticamente 16 caracteres.

Ejemplos.

DIM A\$(9,9) ... Cada secuencia puede tener de 1 hasta 16 caracteres.

DIM A\$(9,9)*50... Cada secuencia puede tener de 1 a 50 caracteres.

■ Variables fijas (A a Z, A\$ a Z\$)

La memoria donde se almacenan valores numéricos o caracteres dispone de 26 variables fijas, las cuales se especifican con las letras A a la Z (con el signo \$ en caso de variables de caracteres). No se pueden usar al mismo tiempo variables numéricas y de caracteres fijas con la misma letra. De lo contrario, se genera un error UV.

Uso incorrecto: 10 PRINT A ; A\$ → Error UV

■ Variables registradas

Pueden usarse, además de las variables fijas, variables compuestas por dos letras mayúsculas o por una letra mayúscula y un número.

Siempre que se defina una variable con tres o más caracteres, aparece un error SN durante la ejecución del programa.

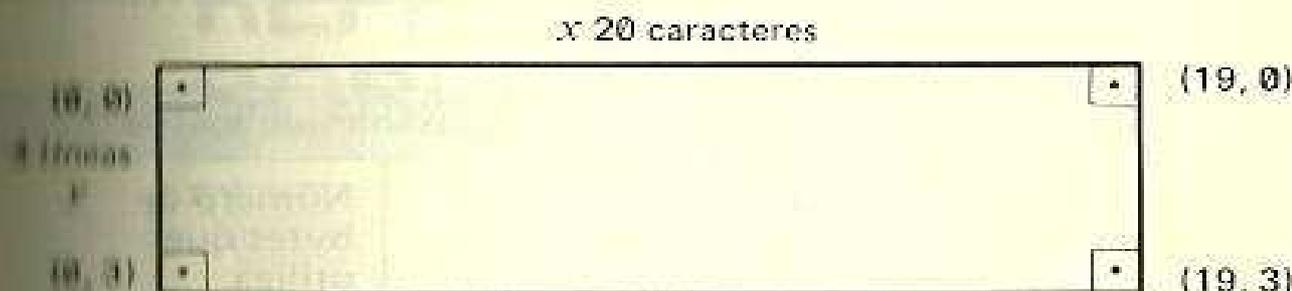
Ejemplos: AB, X1, Y1, X2, Y2, AZ\$, AA\$, B1\$, Z9\$

- El nombre debe comenzar siempre con una letra mayúscula.
- No se pueden usar ciertas palabras reservadas (IF, TO, PI, etc.).
- En las variables numéricas registradas (AB, X1, etc.) se pueden almacenar una mantisa de 12 cifras más un exponente de 2 dígitos.
- Las variables de caracteres registradas tiene capacidad para un máximo de 16 caracteres.
- Se pueden utilizar hasta 40 variables registradas, incluyendo aquellas de tipo matricial. Siempre que se exceda este límite, se genera un error VA, el cual interrumpe la ejecución. Dicha variable debe borrarse entonces mediante el mando ERASE o CLEAR.
- Las variables registradas pueden visualizarse por medio del mando LISTV. Las variables numéricas registradas utilizan 8 bytes, mientras que las de caracteres registradas hacen uso de 17 bytes.

3.8 PANTALLA

■ Coordenadas para los caracteres

La pantalla de cristal líquido (LCD) acomoda 4 líneas de 20 caracteres por una. Cada posición se expresa por medio de la sentencia LOCATE de acuerdo a las siguientes coordenadas:

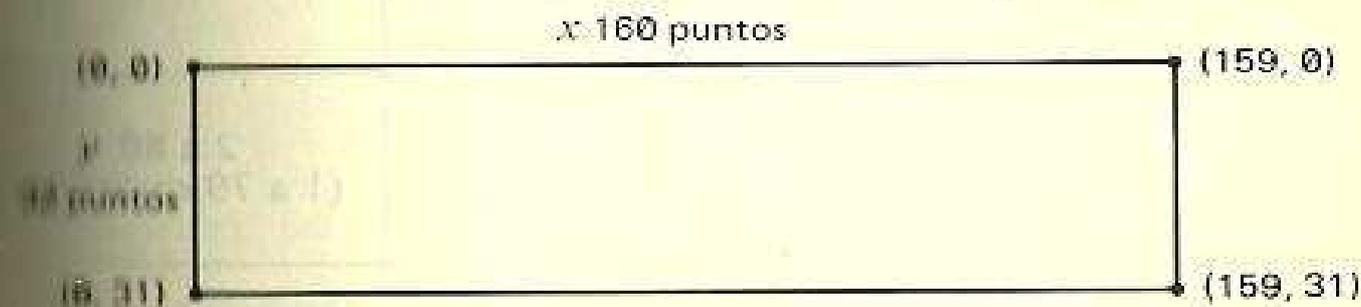


LOCATE (X, Y) Ver LOCATE

Por medio de las coordenadas mencionadas, se pueden visualizar los 20 caracteres que aparecen en la tabla de códigos de la página 328.

■ Coordenadas gráficas

En la pantalla se pueden especificar puntos, en base a las coordenadas a continuación. Para ello, se utilizan las sentencias DRAW y DRAWC, por medio de las cuales se pueden trazar y borrar puntos y líneas rectas.



DRAW (X, Y) Ver DRAW y DRAWC

La función POINT permite verificar la condición de cada punto (encendido o apagado).

POINT (X, Y) Ver POINT

2-9 NUMERO DE BYTES USADOS EN LAS VARIABLES

La capacidad de la RAM disminuye cada vez que en un programa se asigna información a una variable (excepto a las variables fijas). En el cuadro a continuación se muestra el número de bytes por cada tipo de variable. Las variables fijas de la A hasta la Z almacenan la información que se les asigna separadamente, por lo que no afectan directamente la capacidad de la RAM.

| Variables registradas | | Variables de matriz | |
|------------------------|-----------------------------|--|-------------------------------|
| Variable | Número de bytes que utiliza | Variable | Número de bytes que utiliza |
| Variable numérica | 8 | Variable numérica (precisión media) | 4 |
| | | Variable numérica (precisión simple) | 8 |
| Variable de caracteres | 17 | Variable de caracteres (longitud fija) | 17 |
| | | Variable de caracteres (longitud definida) | 2 a 80 (1 a 79 caracteres) |

CAPITULO 3

REFERENCIAS SOBRE "BASIC"

3-1 INTRODUCCION AL LENGUAJE BASIC

Seguramente ya habrá escuchado la palabra BASIC en algún sitio o momento. Este lenguaje de programación es uno de los más sencillos de la actualidad. Su denominación tiene su origen en una sigla del inglés, cuyo equivalente en el español es "Código de Instrucciones Simbólicas Multipropósito para Principiantes".

Lo más atractivo de este lenguaje es que, a pesar de su sencilla sintaxis con mandos cuyos nombres se originan del idioma inglés, permite la preparación de programas sofisticados.

BASIC fue desarrollado en una universidad de los EE.UU. en 1964, para su uso en computadoras de gran tamaño. Desde entonces, se ha transformado en uno de los lenguajes de programación más populares y más utilizados.

En este capítulo, veremos algunos conceptos fundamentales del BASIC. La comprensión de estos fundamentos le permitirán en el futuro desarrollar y escribir programas propios.

3-2 USO DE LAS TECLAS

Si bien la PB-770 tiene gran capacidad para el proceso de información y puede realizar complicados cálculos numéricos, sirve también para llevar a cabo operaciones en forma manual, sin la necesidad de un programa.

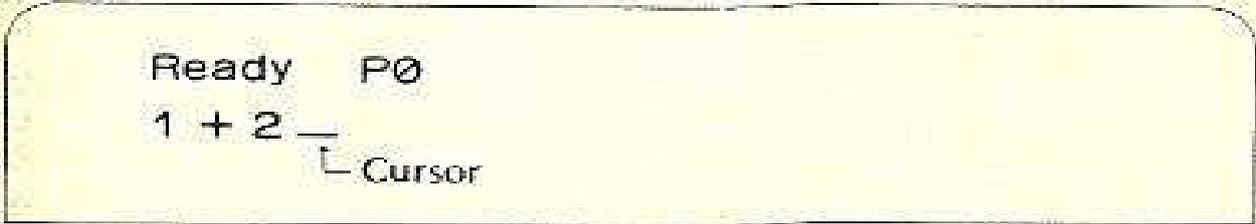
Para ir acostumbrándose a la PB-770, comencemos con una simple operación. Encienda su computadora y verifique la visualización de lo siguiente:

Ready P0 (Indica que se especificó el área de programa N° 0)

El teclado numérico que se encuentra a la derecha cumple, en la PB-770, casi la misma función que en una calculadora, sólo que en él no se incluyen las teclas \times , \div y \equiv . Si bien se encuentra la tecla \equiv en el teclado numérico, su función no es la misma que en una calculadora común.

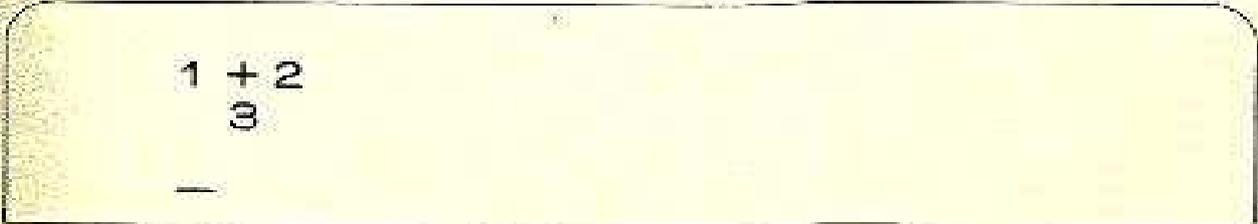
En su lugar, se utiliza la tecla **ENTER**, mientras que las teclas \ast y \div son para multiplicación y división respectivamente.

Intentemos la suma $1+2$. Aparece lo siguiente si entra $1+2$?



Ready P0
1 + 2 =
└ Cursor

Si cometiera algún error, desplace el cursor a la posición donde cometió la equivocación por medio de las teclas \leftarrow y \rightarrow y entre entonces el valor correcto. Pulse a continuación la tecla **ENTER**. Lo siguiente deberá aparecer en la pantalla:



1 + 2
3
—

Además de las 4 funciones aritméticas básicas, con la PB-770 se pueden obtener, entre otras, potencias, funciones trigonométricas, trigonométricas inversas y logarítmicas.

[Expresión numérica]

[Formato de entrada]

$$5 \times 6 \div 2$$

$$\rightarrow 5 * 6 / 2 \quad \boxed{\text{ENTER}}$$

$$6.5^2$$

$$\rightarrow 6.5 \wedge 2 \quad \boxed{\text{ENTER}}$$

$$\frac{\text{SIN } 30^\circ + \text{COS } 60^\circ}{\text{TAN } 45^\circ}$$

$$\rightarrow (\text{SIN } 30 + \text{COS } 60) / \text{TAN } 45 \quad \boxed{\text{ENTER}}$$

3-3 VARIABLES Y SU ASIGNACION

Probemos ahora otro cálculo.

$$500000 \times (1 + 0.07)^{10}, \quad 800000 \times (1 + 0.07)^{10}$$

En estas expresiones, se calcula el interés compuesto a 10 años y se lo suma al capital. ¿Cuál es la forma más simple de realizar estos cálculos? Es muy común que una expresión de cálculo se use más de una vez, por lo cual se almacena en la memoria para ser utilizada cuando sea necesario.

Así,

Entre $A = (1 + 0.07)^{10}$ y entre las dos expresiones anteriores como

$$A * 500000 \text{ [ENTER]}, \quad A * 800000 \text{ [ENTER]}$$

y así se facilita la realización de la operación.

El valor de $(1 + 0.07)^{10}$ se almacena en A. Esta a es lo que se denomina una variable dentro de programa.

Para asignar cierto valor numérico a la variable A, se sigue el siguiente procedimiento:

$$A = 176 \quad \text{(Asignación de 176 en A)}$$

(Lado izquierdo) (Lado derecho)

Se asigna al lado izquierdo del signo de igualdad lo que está al lado derecho del mismo. Así, en este ejemplo, se asigna el número 176 a la variable A usando para ello el signo "=". Confirme la asignación entrando A para así visualizar el contenido de A.

¿Aparece en pantalla el número 176?

Este es un principio muy importante del lenguaje BASIC.

"=" sirve para asignar y no cumple la función de igualdad matemática (excepto en la sentencia IF). Por ejemplo:

$$A = A + 1$$

significa que se asigna a A la suma de $A + 1$. Si el valor de la variable A es 176, se supone que se asigna aquí un valor igual a 177. Confírmelo entrando A .

¿Aparece en la pantalla lo siguiente?

```
A 177
— (Cursor)
```

■  y **ENTER**

Algo también importante para el uso de la PB-700 es la comprensión de la diferencia entre las teclas **ENTER** y . Como podrá haberse observado, la tecla  se utilizó para asignar a A el número 176, mientras que **ENTER** se utilizó para visualizar el contenido de la variable A.

La tecla **ENTER** se utiliza del mismo modo que las teclas en una calculadora para visualizar la respuesta, lo que se denomina en una computadora cálculos manuales. La tecla , por su lado, se utiliza para entrar las instrucciones de un programa BASIC. Por ejemplo, se utiliza para la entrada de programas, para corregir una parte de los mismos o para ejecutar un mando BASIC.

1.4 USO DE LAS VARIABLES

Para la especificación de variables, se utilizan letras (de la A hasta la Z) mayúsculas solas o combinadas con otra letra u otro número (como ser, AA ó N1). Las variables a las cuales se ha asignado un valor numérico pueden usarse libremente en expresiones matemáticas.

Practiquemos las siguientes entradas:

A = 36 

B = 12 

En este caso, se asigna 36 a A y 12 a B. Entre luego

A + B **ENTER**

se utiliza para obtener el resultado de la suma de A y B. Si el resultado visualizado es igual a 48, prosiga con los cálculos a continuación.

A - B **ENTER** → 24

A * B **ENTER** → 432

(A + A) * LOG B **ENTER** → 99.39626599

El dominio del uso de las variables permite una gran versatilidad en la programación. Con el tiempo, se dará cuenta que los sencillos ejemplos aquí presentados son muy limitados. De allí que se recurre a la programación. La preparación de programas no es muy complicada si se comprende la utilización de las variables como se debe.

1.5 ENTRADA DE PROGRAMAS

Antes que nada, echemos un vistazo al procedimiento utilizado para entrar programas. Si observamos el teclado alfabético de la PB-770, veremos que su distribución es igual que la de una máquina de escribir común. Lleve a cabo las siguientes entradas:

SHIFT **PROG** **0**  → Especificación del área P0.

N **E** **W**  → Borrado del programa almacenado en P0.

CLS → Borrado de la pantalla.

[Programa]

[Entrada desde el teclado]

10 CLEAR

1 0 SHIFT CLEAR ↵ →

indica que deben pulsarse la dos teclas juntas.

20 A=A+1

2 0 A = A + 1 ↵

30 LOCATE 7,2

3 0 L O C A T E 7 . 2 ↵

40 PRINT A

4 0 SHIFT PRINT A ↵

50 GOTO 20

5 0 SHIFT GOTO 2 0 ↵

Entre **CLS RUN** ↵ ó **SHIFT P2** ↵. Si la entrada del programa fue correcta, deberán aparecer números en el centro de la pantalla a alta velocidad. Si se visualizara "SN error P0 - n° de línea", significa que se cometió algún error en la entrada del programa. Corríjalo por medio del proceso de depuración.

SHIFT EDIT N° de línea ↵

Por medio de esta entrada se visualiza el número de línea especificado. Desplace el cursor hasta la posición que ha de corregirse, corrija lo que sea necesario y presione la tecla **↵**. Aparecerá entonces la siguiente línea. Una vez finalizada la corrección, presione la tecla **END** y entre **RUN** ↵ nuevamente. Debe notarse que el modo EDIT sirve sólo para la corrección de programas.

■ Areas de programa

La PB-770 posee un total de 10 áreas de programa, de P0 a P9. Cada área de programa es totalmente independiente, por lo cual la unidad puede alojar hasta 10 programas al mismo tiempo, los cuales pueden utilizarse y ejecutarse sin que uno interfiera al otro. Al encenderse la PB-770, aparece en la pantalla el mensaje "Ready P0", para indicar que el área programa en curso es P0. La tecla **END** puede pulsarse en cualquier momento para visualizar en la pantalla el área de programa en curso. El procedimiento a continuación permite cambiar la especificación del área de programa.

PROG 1 ↵ ó **SHIFT PROG** 1 ↵.

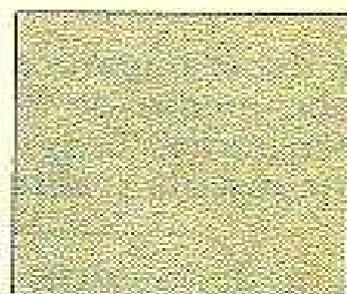
3-6 PROGRAMACION EN BASIC (1)

Problemos un programa en BASIC.

Preparemos un programa para determinar el área del cuadrado de la derecha siguiendo el orden a continuación:

- (1) Entrada de la longitud del lado A.
- (2) Multiplicación del valor entrado por el mismo valor.
- (3) Visualización del resultado.
- (4) Retorno al punto (1).

← A →



```

10 INPUT A ..... ①
20 B = A * A ..... ②
30 PRINT B ..... ③
40 GOTO 10 ..... ④
    
```

Entre, entonces, este programa mediante el siguiente procedimiento. Asegúrese de presionar las teclas correctas.

Encendido de la PB-770 . . . Ready P0 en pantalla.

NEW es una tecla ubicada en el inferior derecho del teclado.

10 A Pulsación simultánea de y .

20 B=A*A se encuentra en el teclado numérico.

30 B Pulsación simultánea de y .

40 10 Pulsación simultánea de y .

Ejecutemos este programa mediante el siguiente procedimiento:

. puede también usarse.

Después de esta entrada, aparece en la pantalla ?_ (_ es lo que se denomina cursor). Entre a continuación:

8.5 

y deberá aparecer la siguiente presentación. En caso contrario, verifique por medio de    la presencia de algún error en la entrada del programa. Tenga especial cuidado con la diferencia entre el cero y la "O" mayúscula y entre el uno y la letra "I".

```

RUN
? 8.5 . . . . . Valor de un lado
  72.25 . . . . . El área es igual a 72,25
? _ (Cursor) . . . . . ¿Cuál es el valor de un lado?
    
```

Analicemos el programa una vez finalizada su ejecución.

```

10 INPUT A
    (mando de entrada) (en la variable A)
    
```

El "10" es el número de línea que indica el orden de ejecución del programa. En este caso, los números de línea aumentan de diez en diez. El mando INPUT solicita la entrada de información desde el teclado, para lo cual visualiza en la pantalla el signo "?". Una vez realizada la entrada, el mando almacena el valor entrado en la variable a continuación del mando (en este caso A).

```

20 B = A * A . . . . . Asigna a B el resultado de A*A
    
```

La línea 20 calcula el área. El valor numérico entrado se multiplica por sí mismo y dicho resultado se asigna a la variable B.

```

30 PRINT B
    (Visualización) (B)
    
```

La línea 30 visualiza el área. PRINT se encarga de ello. Así, en esta línea se lleva a la pantalla el contenido de la variable B.

```

40 GOTO 10
    (Salto) (a la línea 10)
    
```

GOTO es un mando que ordena el salto del programa a la línea con el número que le sigue.

Este programa da una base para la comprensión de los mandos INPUT, PRINT y GOTO. No obstante, este programa es imperfecto, ya que, al ejecutárselo, uno no sabría que información entrar al aparecer en la pantalla el signo "?". Mejoremos entonces el programa mediante el siguiente procedimiento:

 Ready P0 en pantalla.
  Pulsación simultánea de  y .

Una vez visualizada la línea 10, presione la tecla  para desplazar el cursor hasta la posición de A y entre        . Esta modificación resulta en lo siguiente:

```
10 INPUT "A=" ; A
```

Visualiza el mensaje comprendido entre las comillas (" ").

Siempre que se inserte un mensaje, es necesario separarlo de la variable por medio de un punto y coma (tenga cuidado de no entrar en cambio dos puntos).

Luego se visualiza la línea 20. Como en esta línea no es necesario llevar a cabo corrección alguna, pulse la tecla . Aparecerá así la línea 30. Desplace el cursor hasta la posición de B y entre         . La modificación resultará en lo siguiente:

```
30 PRINT "AREA" ; B
```

Se visualiza el mensaje entre comillas.

B aparece a continuación del mensaje.
Un espacio.

A continuación se visualiza la línea 40, en la cual no se necesita modificación alguna, por lo que se pulsa la tecla . Ejecute entonces el programa modificado.

     Puede usarse también  .

De este modo se ha mejorado considerablemente el programa anterior. Su ejecución visualizará "A=?" cuando solicite la entrada de información y "AREA 72.25" al presentar el resultado.

3-7 PROGRAMACION EN BASIC (2)

Veamos otro programa que nos permitirá familiarizarnos con el lenguaje BASIC. En este programa, se obtienen los múltiplos de un valor especificado entre 0 y 200.

```
10 REM MULTIPLE ..... Pulse al final de cada línea
20 A=0
30 INPUT "NUMBER"; N
40 A=A+1
50 B=N*A
60 IF B>200 THEN 100
70 PRINT B;
80 INPUT " OK"; C$
90 GOTO 40
100 END
```

Una vez finalizada la entrada del programa, pulse la tecla **ENT** mientras presiona **SHIFT**. A continuación de ello, la pulsación de la tecla **↵** hace que aparezca en la pantalla la primera línea del programa. En caso de que en ella haya un error, corríjalo desplazando el cursor hasta la posición que corresponda.

| | | | | |
|-----|-----------------|------------|-----------------|--------------------|
| [1] | <u>10</u> | <u>REM</u> | <u>MULTIPLE</u> | Ver la página 200. |
| | Número de línea | Mando | Comentario | |

Los programas en BASIC se componen de numerosas líneas conformadas por un número de línea para cada una de ellas, instrucciones (de programa) y variables o expresiones que incorporan variables.

En la línea 10 aparece la sentencia REM, la cual sirve para incorporar en la lista del programa los comentarios que se deseen, sin afectar la ejecución del programa en sí. En este caso, la sentencia REM indica que éste es un programa para obtener múltiplos. Pulse nuevamente la tecla **↵**.

| | | |
|-----|-----------------|--------------------------|
| [2] | <u>20</u> | <u>A = 0</u> |
| | Número de línea | Variable numérica (fija) |

En la línea 20, se asigna un 0 (cero) a la variable A. Esto es lo que se denomina inicialización.

Ya habrá notado que hasta el momento todas las líneas han sido numeradas con múltiplos de 10. En realidad, en la PB-770 se puede utilizar cualquier número de línea entre 1 y 9999. Sólo que al usar múltiplos de 10, resulta mucho más fácil leer y hacer modificaciones en el programa.

La línea 20 puede expresarse también del siguiente modo.

```
20 LET A = 0
```

Ver la página 195.

LET es una sentencia de asignación que puede omitirse.

Pulse nuevamente la tecla .

```

(1)  30      INPUT      "NUMBER" ; N
      |         |         |         |
      Número  Mando de  Sentencia  Variable
      de línea entrada  con mensaje
  
```

Ver la página 189.

Como la sentencia INPUT solicita una entrada, la ejecución no pasará a la siguiente línea hasta que no se entre un número o letra desde el teclado. El número o letra entrado se asigna a la variable ubicada a continuación de la sentencia y se avanza a la siguiente línea.

Aunque la sentencia de mensaje puede omitirse, es muy conveniente ya que indica al usuario el tipo de información que debe entrar desde el teclado.

El signo "?" aparece o no en la pantalla al ejecutarse el programa según si se coloca un punto y coma o dos puntos, respectivamente, entre la sentencia de mensaje y la variable. Vuelva a pulsar la tecla .

```
(2)  40 A = A + 1
```

La variable A es un contador en el cual se registra el número de veces que se lleva a cabo la ejecución por cada valor entrado. La misma se inicializa (se lleva su valor a 0) en la línea 20. La primera ejecución de la línea 40 hace que el valor de A aumente a 1 (o sea $A = 0 + 1$). Cuando esta línea se ejecuta por segunda vez, el valor de A aumenta a 2 (ya que, al inicio de su ejecución, A ya es igual a 1). O sea que, por cada ejecución de la línea 40, se suma un 1 al valor de A. Sea cual fuere el valor de A al final de programa, se le asigna siempre un 0 cuando se vuelve a ejecutar el programa desde el comienzo (en la línea 20).

| | | | |
|--------------|--------|-------------|---------|
| Ejecución de | 1ª vez | $A = 0 + 1$ | (A = 1) |
| la línea 40: | 2ª vez | $A = 1 + 1$ | (A = 2) |
| | 3ª vez | $A = 2 + 1$ | (A = 3) |
| | | ⋮ | |

Pulse la tecla .

[5] 50 B = N * A

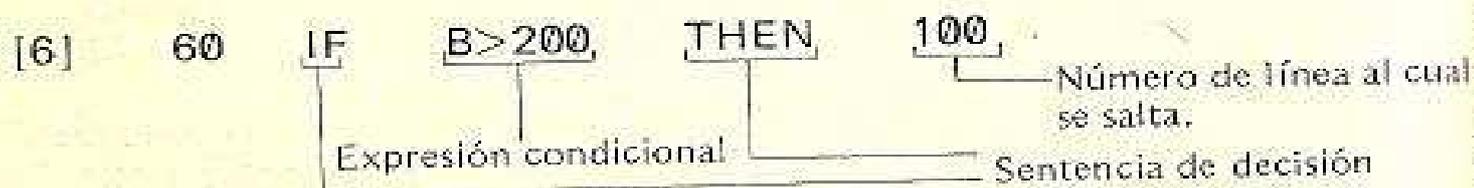
Esta es, de modo similar a la línea 40, una sentencia que asigna el resultado de $N * A$ a la variable B. Al ejecutarse esta línea, A ya tiene un valor igual a 1 por la ejecución de la línea 40. El valor numérico de N se asigna por medio de la entrada desde el teclado por la acción de la sentencia INPUT que se encuentra en la línea 30. Supongamos que se entra en N el número 17,

$$B = 17 * 1 \longrightarrow \text{Se asigna 17 a B.}$$

VER En la segunda ejecución de la línea 50, A tendrá ya un valor igual a 2,

| | | | |
|--------|------------|---|---|
| 2ª vez | B = 17 * 2 | } | El múltiplo de N se asigna continuamente a la variable numérica B, cancelando los valores previos de B. |
| 3ª vez | B = 17 * 3 | | |
| : | | | |

Echemos un vistazo a la línea siguiente. Pulse para ello .



Esta es una sentencia condicional la cual establece que si (IF en inglés) el valor de B es mayor que 200 ($B > 200$) se salte a la línea 100. En otras palabras, siempre que el valor de B sea igual o menor que 200, el programa avanzará a la siguiente línea sin realizar el salto. Al repetirse 12 (17×12) veces la línea 50 y cuando el valor de B llega a 204, se salta a la línea número 100.

IF expresión THEN número de línea **Ver la página 100**

└ Al cumplirse esta expresión, el programa salta al número de línea especificado.

Vuelva a presionar la tecla .

```

(7) 70 PRINT B ;
      |
      |----- Mando de visualización en pantalla
      |
      |----- B ;
              |
              |----- Variable numérica
  
```

Ver la página 198.

PRINT es un mando de visualización. En este programa, este mando visualiza en la pantalla el contenido de la variable B. El punto y coma a continuación de B sirve para que la visualización sea continua así como aparece lo siguiente sin avanzar de línea en la pantalla. De este modo, la visualización solicitada en la línea 80 se realizará directamente después de B, sin cambiar de línea.

```

(8) 80 INPUT "OK" ; C$
      |
      |----- Mando de entrada
      |
      |----- "OK" ;
              |
              |----- Mensaje
              |
              |----- C$
                      |
                      |----- Variable de caracteres
  
```

Ver la página 189.

La sentencia INPUT que aparece en la línea 30 se utiliza nuevamente en esta línea para la entrada de caracteres en una variable. Al ejecutarse la línea 80, aparece en pantalla el mensaje "OK?". Así se podrá asignar desde el teclado un máximo de 7 caracteres a la variable de caracteres C\$. Si se utilizara una variable numérica, como ser C, la computadora sólo aceptaría una entrada numérica; en caso de intentarse la entrada de caracteres en una variable numérica, se genera un error SN.

La función de esta línea es retener momentáneamente la visualización del resultado obtenido en la operación de la línea 70 mediante una sentencia INPUT. En caso de no incluirse esta línea, la repetición continua de la operación de la línea 70, y la visualización de sus resultados, no permitiría ver en la pantalla nada. Pulse la tecla .

```

(9) 90 GOTO 40
      |
      |----- Mando de salto
      |
      |----- 40
              |
              |----- Número de línea donde se realiza el salto
  
```

Ver la página 184.

Esta es un mando que provoca un salto incondicional a la línea 40. Pulse la tecla .

```

(10) 100 END
  
```

Ver la página 170.

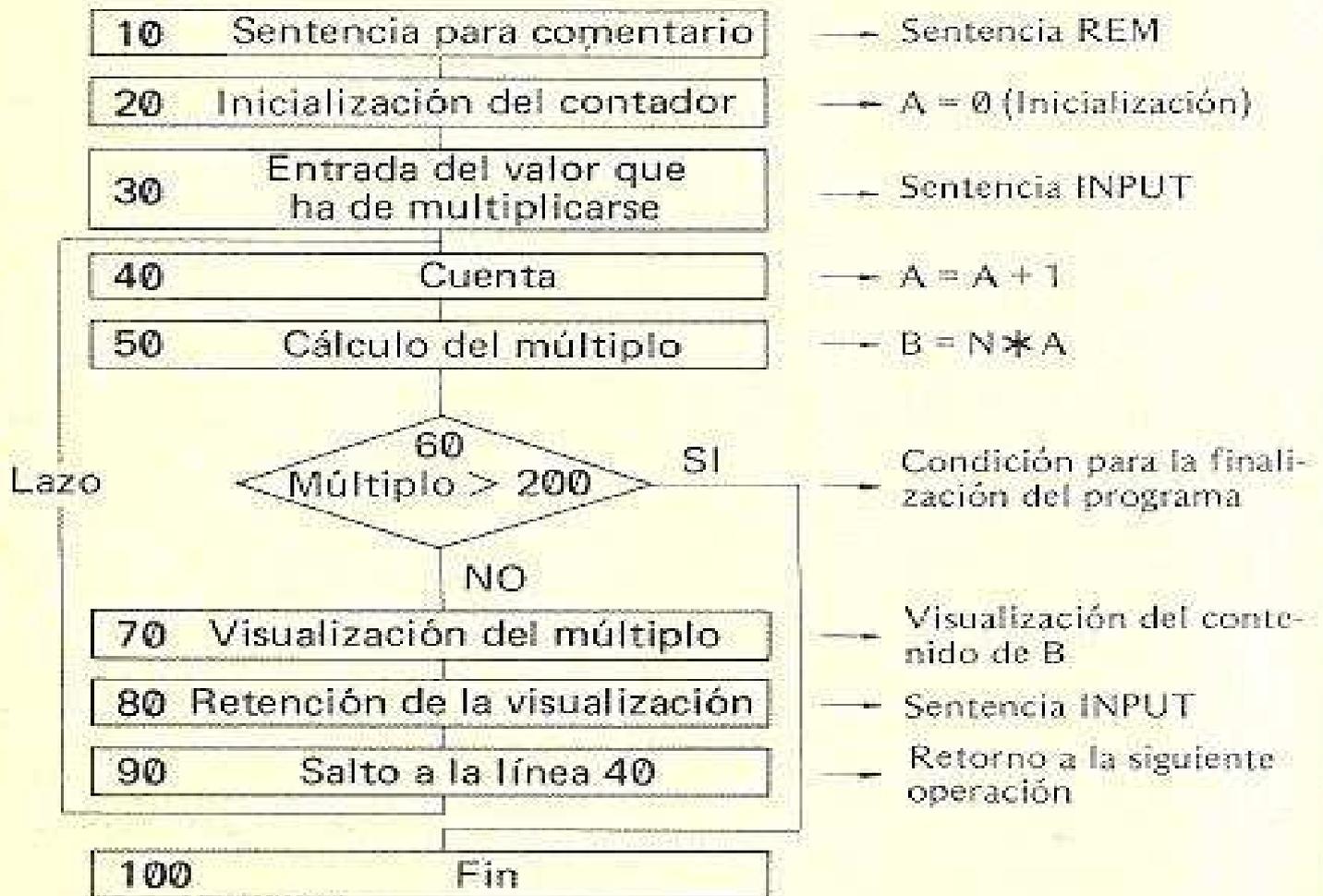
Este es un mando que da por terminado el programa. Este es un mando especial para la finalización de un programa; su ausencia antes de una subrutina, haría que la misma se ejecute innecesariamente (subrutinas, página 180).

El mando END puede insertarse en la línea 60 del siguiente modo. En este caso, no se necesitaría la línea 100.

```

60 IF B>200 THEN END
  
```

Observemos el flujo del programa en el siguiente diagrama:



La línea 90 provoca un salto incondicional (un salto que no requiere condiciones para ejecutarse) nuevamente a la línea 40, por lo que la ejecución continuará indefinidamente entre estas dos líneas. Sólo puede saltarse a la línea 100 cuando se cumple la condición de la sentencia condicional de la línea 60 (cuando el producto del múltiplo por el valor entrado es mayor que 200).

EJERCICIO

- Prepare un programa para calcular la suma acumulativa de una serie de valores numéricos entrados.

Procedimiento

- 1) Inicialización de todas las variables
- 2) Entrada del valor numérico que ha de sumarse
- 3) Suma del número entrado a la suma de los valores anteriores
- 4) Visualización del resultado

Solución

```

10 CLEAR
20 INPUT "DATA = "; A
30 B = B + A
40 PRINT "TOTAL = "; B
50 GOTO 20

```

Justificación

El mando CLEAR de la línea 10 sirve para inicializar todas las variables numéricas y de caracteres. En este caso, sería suficiente con inicializar la variable B (por medio de $B=0$), en la cual se almacena el resultado de la operación. Si el mando GOTO de la línea 50 provocara un salto a la línea 10, se inicializaría la variable y no se podrían acumular los valores numéricos entrados.

3-8 EJECUCION DE PROGRAMAS

Ya hemos verificado el programa entrado.

Ejecutemos este programa.

Pulse las teclas **PRG** y **CLS** . Entre luego el mando de ejecución **R U N** .

Si el programa se encuentra en el área P0, el mismo podrá ejecutarse directamente por medio de **SHIFT** **PRG** (funciona igual que **R U N** ).

Si el programa se ha entrado correctamente, lo siguiente aparecerá en la pantalla:

```
RUN
NUMBER ?_
```

Este mensaje solicita el valor que desea multiplicarse. Haga la siguiente entrada:

17 

La presentación será la siguiente:

```
RUN
NUMBER ? 17
17 OK ?_
```

Esta visualización muestra que 17 es el múltiplo mínimo de 17. A continuación se solicita una confirmación. Presione la tecla  para visualizar el siguiente múltiplo.

```
RUN
NUMBER ? 17
17 OK ?
34 OK ?_
```

Pulse la tecla  para obtener el siguiente múltiplo. De este modo, se obtendrán los múltiplos hasta 187. Al presionar luego la tecla  para obtener el siguiente múltiplo, el resultado no se visualiza ya que se ha satisfecho la condición incluida en la línea 60. El programa finaliza ya que el múltiplo obtenido es mayor que 200. Para volver a ejecutar el programa, presione nuevamente **R U N** .

Modificación del programa

Algunas modificaciones nos ayudarán a familiarizarnos aún más con el uso de los numerosos mandos vistos.

Extendamos el límite del resultado a 100 (línea 60: $B > 300$ en lugar de $B > 200$). La línea 60 puede visualizarse por medio de **SHIFT** **F6** 60 **↵**. Desplace el cursor para cambiar el número de la línea 60. Pulse entonces las teclas **↵** y **DEL** y entre luego **RUN** **↵**.

Vamos a continuación algunas variaciones en el uso de la sentencia PRINT que se encuentra en la línea 70.

El punto y coma a continuación de la variable B en "PRINT B" hace, como ya sabemos, que la visualización sea continua. Quitemos ahora este punto y coma.

SHIFT **F6** 70 **↵**

Desplace el cursor hasta la posición de ";" y pulse la tecla **DEL**. Así desaparece el punto y coma. Presione las teclas **↵** y **BRK**, y ejecute el programa del modo ya visto.

El mensaje "OK?" aparece esta vez por debajo del múltiplo obtenido. Al quitarse el ";", el siguiente carácter no se visualiza a continuación del anterior, sino al comienzo de la siguiente línea.

Secuencia de ejecución**NUMBER? 17**

17 OK?

34 OK?

51 OK?

68 OK?

85 OK?

102 OK?

119 OK?

136 OK?

153 OK?

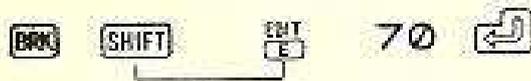
170 OK?

187 OK?

Ready P0

3-9 CONFIGURACION DE LA PANTALLA

Veamos a continuación algunas técnicas para el control de la pantalla mediante la modificación de la línea 70 en el programa para calcular múltiplos.



Debe aparecer lo siguiente:

70 PRINT B

Ver la página 198.

Visualicemos el contenido de la variable contadora A junto con el de la variable B.

[1] 70 PRINT A ; B ;

Lleve el cursor a la posición de B para realizar la modificación y entre   . Así se abren dos espacios previos a B; entre entonces     . No olvide de presionar la tecla  al final. Presione , luego ejecute el programa y confirme la pantalla. Llevemos a cabo la siguiente modificación:

[2] 70 PRINT A, B ;

Para llevar a cabo las modificaciones, presione la tecla , lleve a la pantalla la línea 70 (en el modo EDIT) y haga uso del cursor del modo ya citado previamente. Una vez hechas las correcciones, presione la tecla  y vuelva a ejecutar el programa.

Así se completa la modificación.

Llevemos a cabo la siguiente modificación. Pulse, para ello, la tecla .

[3] 70 PRINT A ; TAB (8) ; B ;

Ejecute el programa luego de realizada la modificación.

La función TAB (8) hace avanzar el cursor el número de posiciones especificado entre paréntesis. Confirme que haya un espacio entre las presentaciones de las variables A y B. Tenga en cuenta que, cuando el contenido de la variable B a visualizarse tiene tres cifras, se alinea el lado izquierdo y el último carácter se desplaza un espacio hacia la derecha.

| | |
|---|---------|
| 3 | 51 OK? |
| 4 | 68 OK? |
| 5 | 85 OK? |
| 6 | 102 OK? |

En este programa, el cual obtiene sólo múltiplos del número entrado, esto no es problema. Sin embargo, cuando se visualizan cantidades o precios, debe entonces alinearse el extremo derecho de las cifras. Para ello, se utilizaría la función USING del siguiente modo:

```
14) 70 PRINT A; TAB (8) ; USING "###" ; B ;
```

Para mayores detalles sobre USING, remítase a la página 263. Ejecute el programa y verifique el resultado.

El mando LOCATE sirve también para controlar la visualización en la pantalla. Usemos este mando en la línea 70.

```
14) 70 CLS : LOCATE 5,2 : PRINT A; B; ...
```

Mando Mando Mando

Se pueden usar en una sola línea más de un mando; ellos deben unirse por medio de dos puntos (:) en lo que se denomina una "sentencia múltiple".

No olvide presionar la tecla  una vez finalizada la modificación. Debe aparecer lo siguiente en la pantalla:

| | | |
|---|----|-----|
| 1 | 17 | OK? |
|---|----|-----|

La presentación se lleva a cabo en el medio de la pantalla. El mando LOCATE tiene el siguiente formato:

LOCATE X, Y

La columna se especifica en la variable X y la línea en Y. Para mayores detalles, remítase a la página 196.

3-10 OPERACIONES REITERATIVAS EN PROGRAMAS

Una "rutina" es una tarea dentro de un programa, la cual puede o no repetirse varias veces. Por ejemplo, supongamos que deseamos encontrar determinada secuencia de caracteres o valor numérico dentro de una gran cantidad de información. O que necesitamos clasificar dicha información en determinado orden. Sea cual fuere el caso, podemos hacer que la computadora compare la información con ciertos valores, con otra información, o que la clasifique totalmente en determinado orden. Los mandos incluidos en esta sección son esenciales para dichas aplicaciones.



Ver la página 150.

Al entrarse la secuencia citada, lo siguiente aparece en la pantalla:

```
P ♥♥23456789. ANGLE 0
      8KB 4720B
Ready P0
—
```

Los números de las áreas de programa que ya han sido utilizadas para almacenar programas o datos han sido reemplazados por el símbolo ♥. En el ejemplo de arriba, podrá observarse que las áreas 0 y 1 contienen ya programas. 8KB es la capacidad total de la RAM. "4720B" indica que en la RAM quedan 4720 bytes disponibles. Este número, por supuesto, aumenta por cada módulo RAM que se instale en la unidad. Luego de "ANGLE" aparece un número, del 0 al 2, el cual indica la unidad angular (ver la página 153). Este valor, el cual es siempre 0 (para grados) al encenderse la computadora, no afecta los cálculos comunes. "Ready P0" indica el área de programa en curso. En este caso, en P0 se puede entrar un programa. Como en este ejemplo las áreas de programa 0 y 1 ya están ocupadas, entraríamos PROG seguida por un valor de 2 a 9 y luego . Entonces, sí, se puede entrar el siguiente programa.

```
10 CLS
20 FOR A = 1 TO 20
30 PRINT CHR$(254);
40 NEXT A
```

Entre lo siguiente para visualizar la primera línea del programa y confirmar que la entrada haya sido correcta.

   → Especifica el modo EDIT

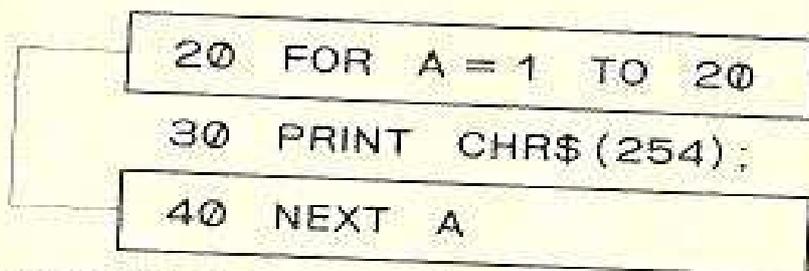
Avance por línea usando .

Una vez que aparezca la indicación "Ready P0", ejecute el programa entrado. Este programa visualiza un cuadrado  20 veces. Veamos los nuevos mandos incorporados en el programa.

(1) 10 CLS

CLS es un mando que borra toda la pantalla y lleva el cursor a la primera posición superior izquierda.

(2)



Las líneas 20 y 40 conforman un único mando. Ello es:

| | |
|-----------------------|--|
| FOR A = 1 TO 20 | Asigna en forma secuencial los números del 1 al 20 a la variable A. |
| | |
| NEXT A | Siempre que A sea menor que 20, se suma 1 a la variable A y se vuelve a FOR. |

Esto es lo que se denomina un lazo FOR-NEXT. Veamos más en detalle como se ejecuta.

- 1) Se asigna 1 a A.
- 2) Se ejecuta la línea 30.
- 3) Se verifica si A es menor que 20 en NEXT A (línea 40)..
- 4) Como A=1, se retorna a la línea 20 donde se asigna 2 a la variable A.
- 5) Se ejecuta la línea 30.
- 6) Idem paso (3).
- 7) Como A=2, se vuelve a la línea 20 y se asigna un 3 a la variable A.
- 8) Cuando el valor de A finalmente llega a 21, se ejecuta la línea que se encuentra a continuación de la sentencia NEXT. En este ejemplo, no hay línea alguna a continuación de NEXT, por lo que se da por terminado el programa y aparece en la pantalla el mensaje "Ready P0".
- 9) Tomemos un vistazo a la línea 30, la cual se repite 20 veces en este programa.

[3] 30 PRINT CHR\$ (254) ; Ver la página 245.
 Visualización CHR\$ (254) └─ misma línea

Todos los caracteres y teclas tienen asignados ciertos códigos que van desde 0 hasta 255. CHR\$ (254) es la función que especifica el código de caracteres 254 () (ver la tabla de la página 328).

Como este carácter no puede entrarse directamente desde el teclado, se lo especifica por medio de la función CHR\$ (254). Esta función es primordial para la especificación de dichos símbolos y caracteres gráficos que no pueden entrarse directamente desde el teclado. "CHR\$(" se entra por medio de  .

EJERCICIO

- Prepare un programa que use el lazo FOR-NEXT para visualizar números enteros desde 1 hasta un límite especificado en forma continua.

Procedimiento

```
FOR A = 1 TO N
  {
NEXT A
```

Solución

```
10 CLS
20 INPUT "NUMBER"; N
30 FOR A=1 TO N
40 PRINT A;
50 NEXT A
60 END
```

Explicación

Para la entrada del número límite en la línea 20, se ha incorporado el mensaje "NUMBER?". El valor numérico entrado se asigna a la variable numérica N, mientras que el número de repeticiones se especifica en la línea 30. Si se entrara en N un valor numérico igual a 15, la línea 40 se ejecutaría dicho número de veces. En la línea 40 se visualiza el contenido de la variable A usada en el lazo FOR-NEXT. Queda entonces claro que por cada ejecución del lazo FOR-NEXT, el valor de A aumenta en incrementos de 1.

3-11 PROGRAMA PARA LA SUMA DEL TOTAL

Cierto tiempo es necesario para familiarizarse con los lazos FOR-NEXT presentados en la sección anterior. Veamos, para ello, el programa a continuación.

Esta vez usaremos el área de programa P4. Designe dicha área mediante el procedimiento ya descrito anteriormente. Entre luego NEW  y proceda con la entrada del programa.

Este programa computa el costo acumulativo de una serie de artículos con precios unitarios diferentes. Se obtiene, también, el subtotal para cada artículo. El número total de artículos diferentes se entra al comienzo del programa.

```
10 CLEAR
20 INPUT "NUMBER OF ARTICLES";N
30 FOR A=1 TO N
40 INPUT "UNIT PRICE";B
50 INPUT "QUANTITY";C
60 PRINT "SUBTOTAL";B*C
70 D=D+B*C
80 NEXT A
90 PRINT "TOTAL";TAB(10);"$";D
100 END
```

El proceso en cada línea del programa es el siguiente:

- 10 Inicialización de las variables
- 20 Solicita la entrada del número de artículos a procesarse (asignación del número a la variable N).
- 30 Solicita el precio unitario y la cantidad el número de veces especificado por N. Después de cada entrada, se visualiza el subtotal, el cual se suma al total.
- 40
- 50
- 60 Visualiza el monto total.
- 100 Fin del programa.

Analicemos en detalle el lazo FOR-NEXT de las líneas 30 hasta 80. Las operaciones que se realizan en el lazo son las siguientes:

- (1) Entrada del precio unitario y su asignación a la variable B.
- (2) Entrada de la cantidad y su asignación a la variable C.
- (3) Multiplicación del precio unitario por la cantidad y visualización del subtotal.
- (4) Suma del subtotal al total.

La línea 90 es un ejemplo de cómo se puede visualizar la información ordenadamente en la pantalla.

```

90 PRINT "TOTAL"; TAB (10) ; "$" ; D
    
```

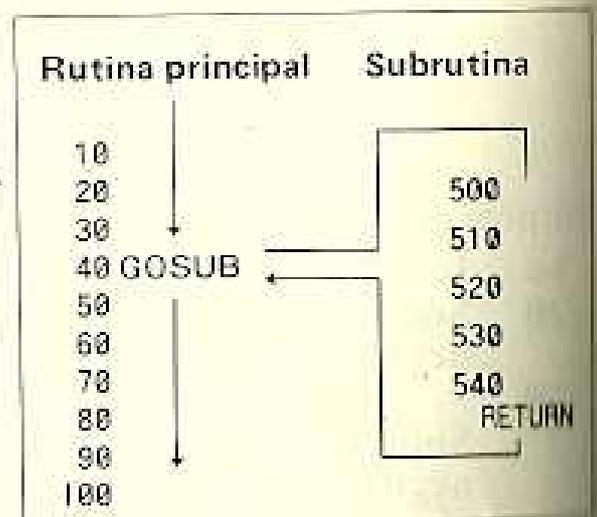
Visualización del total.
10 espacios en blanco.
Visualización del signo \$.
Visualización del monto total.

Reconstruyamos el programa en base al uso de subrutinas. A continuación se ilustra el concepto básico para hacerlo. El mando utilizado para este procedimiento es GOSUB-RETURN.

Las líneas 40 hasta 70 del programa anterior se las numera de 500 a 540, y se inserta el mando GOSUB en la línea 40.

```

    GOSUB 500
    Salto a la subrutina que comienza en la línea 500.
    RETURN
    Retorno al mando a continuación de GOSUB.
    
```



Las líneas 30 a 80 se modifican del siguiente modo:

```

30 FOR A=1 TO N
40 GOSUB 500
50 NEXT A
    
```

Si formamos así un único programa, obtendremos los siguiente:

```
10 CLEAR
20 INPUT "NUMBER OF ARTICLES";N
30 FOR A=1 TO N
40 GOSUB 500
50 NEXT A
60 PRINT "TOTAL";TAB(10);"$";D
70 END
500 PRINT A;TAB(5);"UNIT PRICE";:INPUT
    B
510 PRINT A;TAB(5);"QUANTITY";:INPUT C
520 PRINT "SUBTOTAL";B*C:BEEP 1
530 D=D+B*C
540 RETURN
```

] Lazo FOR-NEXT. Se repite N veces.

] Subrutina

Entre este programa en otra área de programa, y compare la ejecución de ambos.

3-12 VARIABLES DE CARACTERES

Antes de ver cómo se almacenan grandes volúmenes de datos, echemos un vistazo a cómo se procesan los datos de caracteres. Como ya se ha mencionado anteriormente, hay dos tipos principales de variables: las numéricas (como ser A, B, C, ó A1), a las cuales sólo se pueden asignar valores numéricos, y las variables de caracteres (como ser A\$, B\$ ó A1\$), a las cuales se pueden asignar caracteres y símbolos.

Seguramente no habrá quedado muy clara la diferencia entre ambos tipos, ya que afirmamos que a las variables de caracteres pueden asignarse tanto símbolos como caracteres alfabéticos. Debe tenerse en cuenta, no obstante, que una variable numérica expresa una cantidad, mientras que un valor numérico asignado a una variable de caracteres expresa no más que uno o más caracteres.

Por ejemplo:

(Valor numérico)

4 + 3 **ENTER** → 7

(Carácter)

"4" + "3" **ENTER** → 43

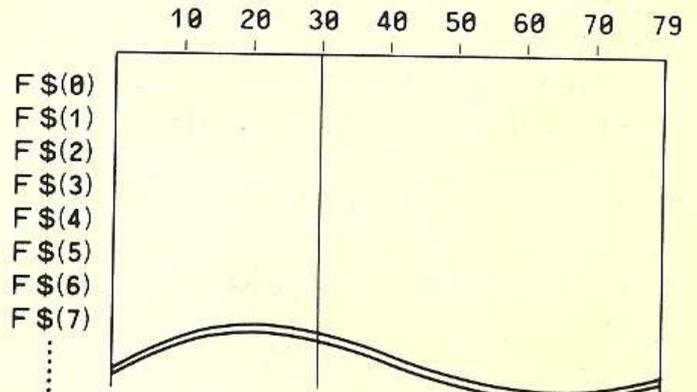
Como podrá observarse a la derecha, los caracteres asignados se encuentran entre comillas, igual que los mensajes de los mandos INPUT y PRINT. La "suma" de dos caracteres da como resultado una serie formada por los dos caracteres "sumados". Los números tratados como caracteres pueden convertirse en valores numéricos mediante el uso de la función VAL (ver la página 247). En la sección siguiente veremos el uso de variables de caracteres en matrices. Cada elemento de una matriz de caracteres aloja, por lo general, hasta 16 caracteres; este límite puede especificarse hasta 79 caracteres por elemento de la matriz.

Las matrices de caracteres se especifican del siguiente modo.

DIM F\$ (50) * 30

Variable de caracteres Cantidad de datos Número máximo de caracteres por variable

Mediante la especificación del número máximo de caracteres por elemento de la matriz se reserva el área de memoria necesaria. Cuando se especifica 30, como arriba, la memoria necesaria se calcula del siguiente modo.



$$50 \times (30 + 1) = 1550 \text{ (Bytes)}$$

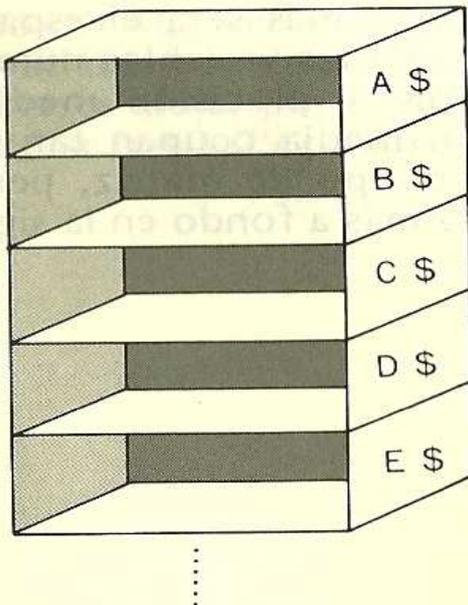
Número de caracteres almacenados + 1

Cuanto menos memoria se utilice por variable, más será el espacio de memoria que quedará disponible. En el caso de las variables numéricas, se puede elegir entre las de precisión simple y precisión media. Las matrices de variables numéricas de precisión media ocupan tan sólo la mitad de la memoria que requiere el mismo tipo de matriz, pero con precisión simple. Esta diferencia se explicará más a fondo en la siguiente sección.

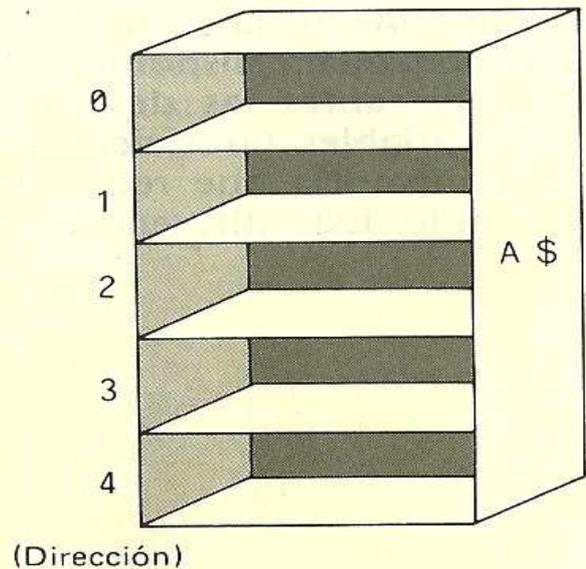
3-13 ¿QUE ES UNA DIMENSION?

En la terminología de la informática, la palabra "dimensión" suele escucharse con frecuencia. Imaginemos una estantería como la que se ilustra en la figura (2). Supongamos que en cada estante se puede almacenar (guardar) un dato (sean valores numéricos o caracteres), a los cuales se puede tener acceso en cualquier momento. En este ejemplo, se han preparado 5 estantes bajo un único nombre (de variable). Si almacenamos el nombre "Smith" en $A\$(0)$, "Johnson" en $A\$(1)$ y "Foster" en $A\$(2)$, podremos recuperar dichos datos con sólo especificar el nombre de la variable (o estantería) y el número de control (o de estante dentro de la estantería especificada). Ello permite almacenar más de un dato bajo el mismo nombre de variable (en este caso $A\$($). Sin las matrices, cada unidad de información necesitaría su variable propia, como se ilustra en la figura (1). Ello dificultaría el tratamiento de la información resultando en programas poco eficientes.

(1) 1 dato por variable



(2) Matriz unidimensional, donde se pueden almacenar muchos datos con un sólo nombre de variable.

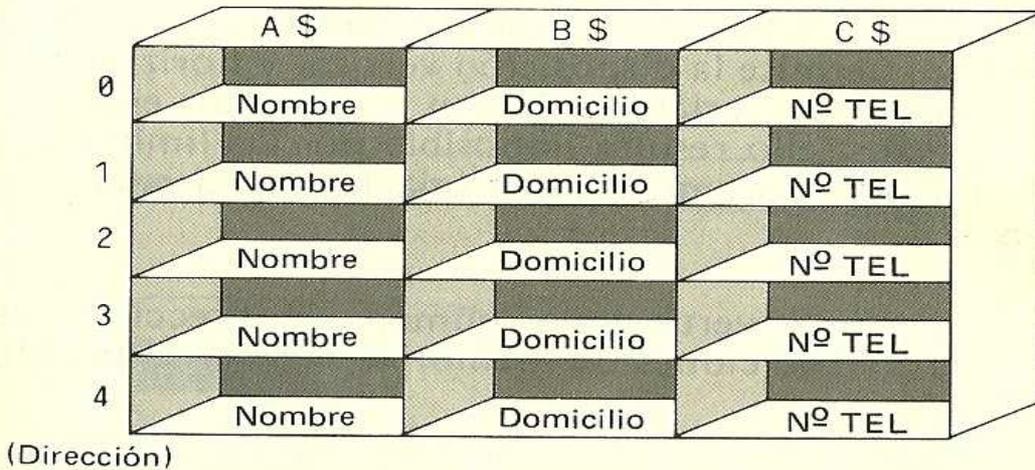


■ **Matrices unidimensionales**

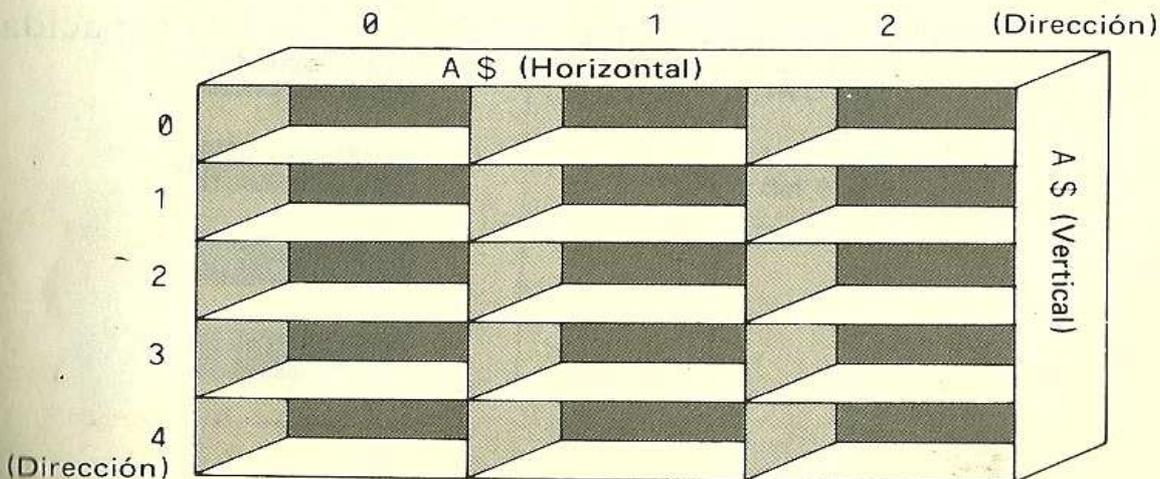
El método ilustrado en la figure (2), conocido por el nombre de matriz unidimensional, es muy práctico para almacenar gran cantidad de información. Si bien en este ejemplo se usaron tan sólo 5 "estantes", por cada matriz se pueden asignar hasta 256 elementos (0 hasta 255).

Antes de usar una matriz, debe reservarse el espacio de memoria necesario para la misma. Todo intento de almacenar datos en una matriz sin haber reservado el espacio necesario para la misma hará que aparezca un error UV.

(3) Libro de domicilios usando matrices unidimensionales



(4) Matriz bidimensional



■ Matrices bidimensionales

En base a lo que ya sabemos sobre las matrices unidimensionales, podríamos preparar una lista de nombres, domicilios y números telefónicos. En la figura (3), vemos cómo se han asignado los nombres a A\$, los domicilios a B\$ y los números de teléfono a C\$. Para averiguar el número telefónico correspondiente a determinado nombre, entraríamos el nombre y localizaríamos los "estantes" correspondientes en B\$ y C\$ en base al número de estante donde se encontró el nombre entrado dentro de la estantería donde se encuentran los nombres, o sea A\$. De este modo, especificando el nombre de la variable y su número de elemento, se puede tener acceso a cada uno de los datos almacenados. Otra forma de lograr el mismo resultado sería la utilización de las denominadas matrices bidimensionales, del modo ilustrado en la figura (4). Este tipo de matrices permite la disposición vertical y horizontal de datos bajo un único nombre de variable. Si bien se podrían especificar 256 filas por 256 columnas, ello resulta imposible por las limitaciones en la capacidad de la memoria. Como la memoria necesaria para una matriz bidimensional es:

Número de direcciones verticales X número de direcciones horizontales X número de posiciones de memorias necesarias por dirección

Una matriz de 256×256 con 16 caracteres por variable necesitaría:

$$256 \times 256 \times (16+1) = 1.114.112 \text{ bytes.}$$

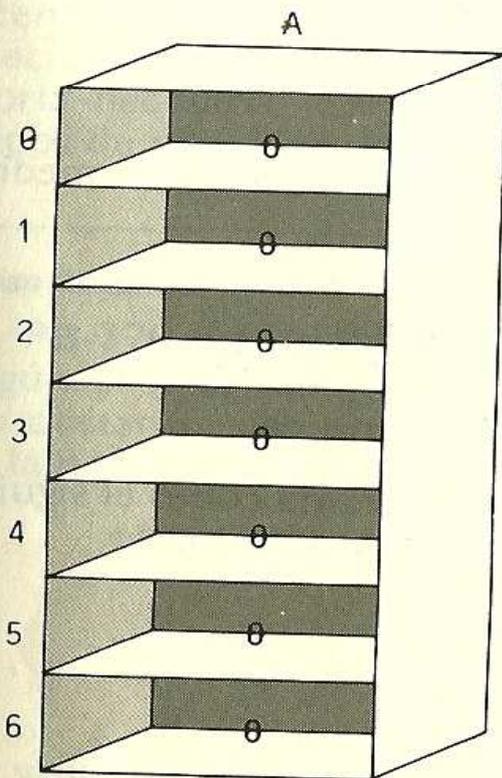
Ello, obviamente, provocaría un error por superación de la capacidad de la memoria.

3-14 MATRICES DE VARIABLES NUMERICAS

Las matrices numéricas se utilizan para almacenar valores numéricos en los "estantes" descritos en la sección anterior. En la sección anterior, se utilizaron los nombres de variables (variable matriciales en el caso de las matrices) A\$, B\$ y C\$. Las variables matriciales numéricas, por su lado, se basan en el mismo principio, sólo que su tratamiento y expresión dentro de los programas son diferentes.

Preparemos primeramente un programa usando una matriz de variables numéricas unidimensional, como la que vimos en la sección anterior. Como esta "estantería" alojará valores numéricos, su nombre debe ser tan sólo una letra cualquiera del alfabeto. Supongamos que usamos el nombre de variable "A", para una matriz con un total de 7 elementos (0 hasta 6).

(1) Configuración y expresión de la variable de matriz A

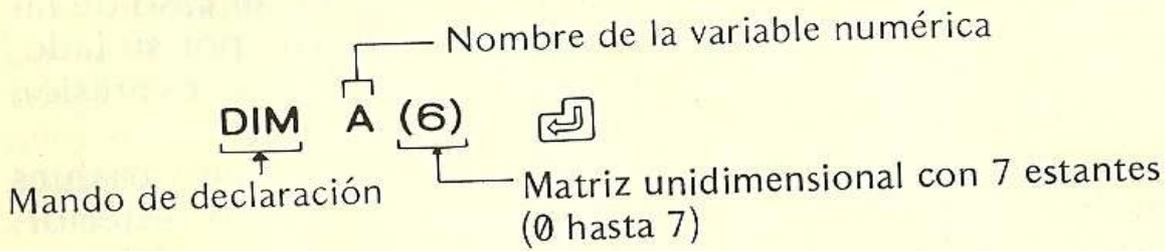


DIM A (6) para preparar los estantes

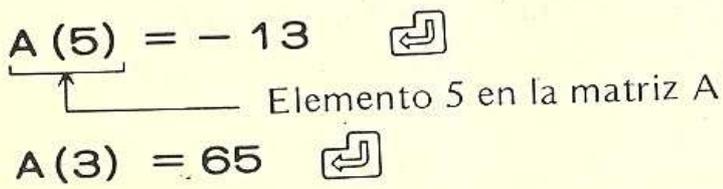
¡NOTA!

Los siete datos se numeran del 0 al 6.

He a continuación la declaración de una matriz de variables de nombre A, que posee desde el estante 0 hasta el estante 6.



Esta especificación establece la matriz A y asigna un cero a todos los elementos de la misma (vacía todos los estantes). En caso de que aparezca un error DD cuando se entra el mando DIM, entre CLEAR ↵ y vuelva a repetir la operación. Almacenemos algunos datos en los estantes.



Verifique los datos asignados a los estantes especificados por medio del siguiente procedimiento:

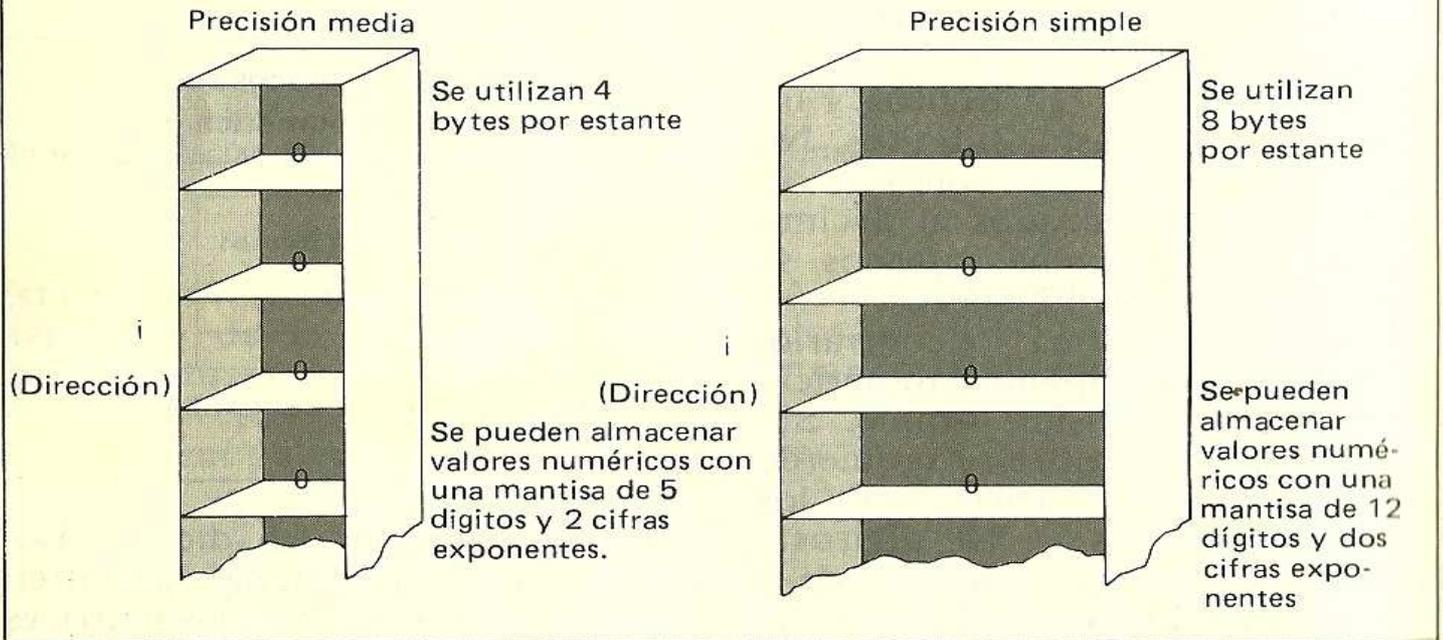
- A (5) **ENTER** ➔ -13 aparece en pantalla
- A (3) **ENTER** ➔ 65 aparece en pantalla

Estas operaciones pueden incorporarse en un programa como el siguiente:

```

5 CLS
10 CLEAR
20 DIM A(6)
30 INPUT "A(5) = "; A(5)
40 INPUT "A(3) = "; A(3)
50 PRINT A(5); A(3)
60 END
    
```


(3) Tamaño de los estantes con precisión media y simple



Las matrices de precisión simple se expresan del siguiente modo.

A (i) Matriz numérica de precisión simple

Las matrices numéricas de precisión media se expresan colocando un signo de admiración a continuación del nombre de la variable.

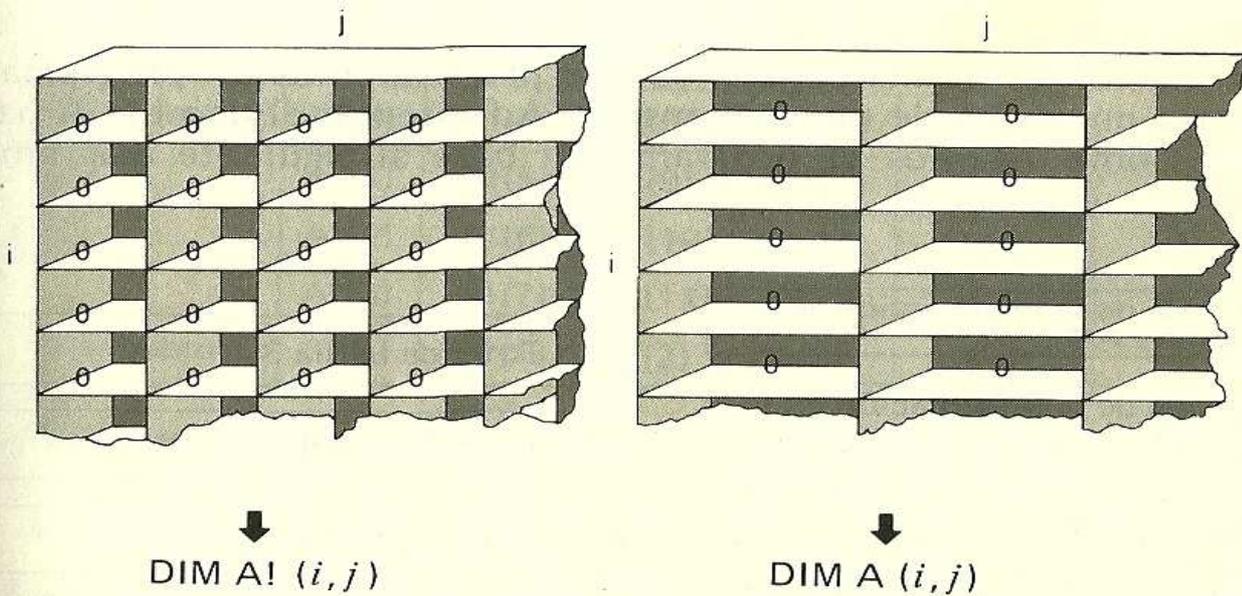
A! (i) Matriz numérica de precisión media

■ **Precisión simple y media en una matriz bidimensional**

Del mismo modo que en las matrices unidimensionales, la precisión de las matrices bidimensionales es simple siempre que no se especifique lo contrario. En la figura (4) se muestra un ejemplo.

DIM A! (i, j) Precisión media
 DIM A (i, j) Precisión simple

(4) **Matrices numéricas bidimensionales de precisión simple y media**

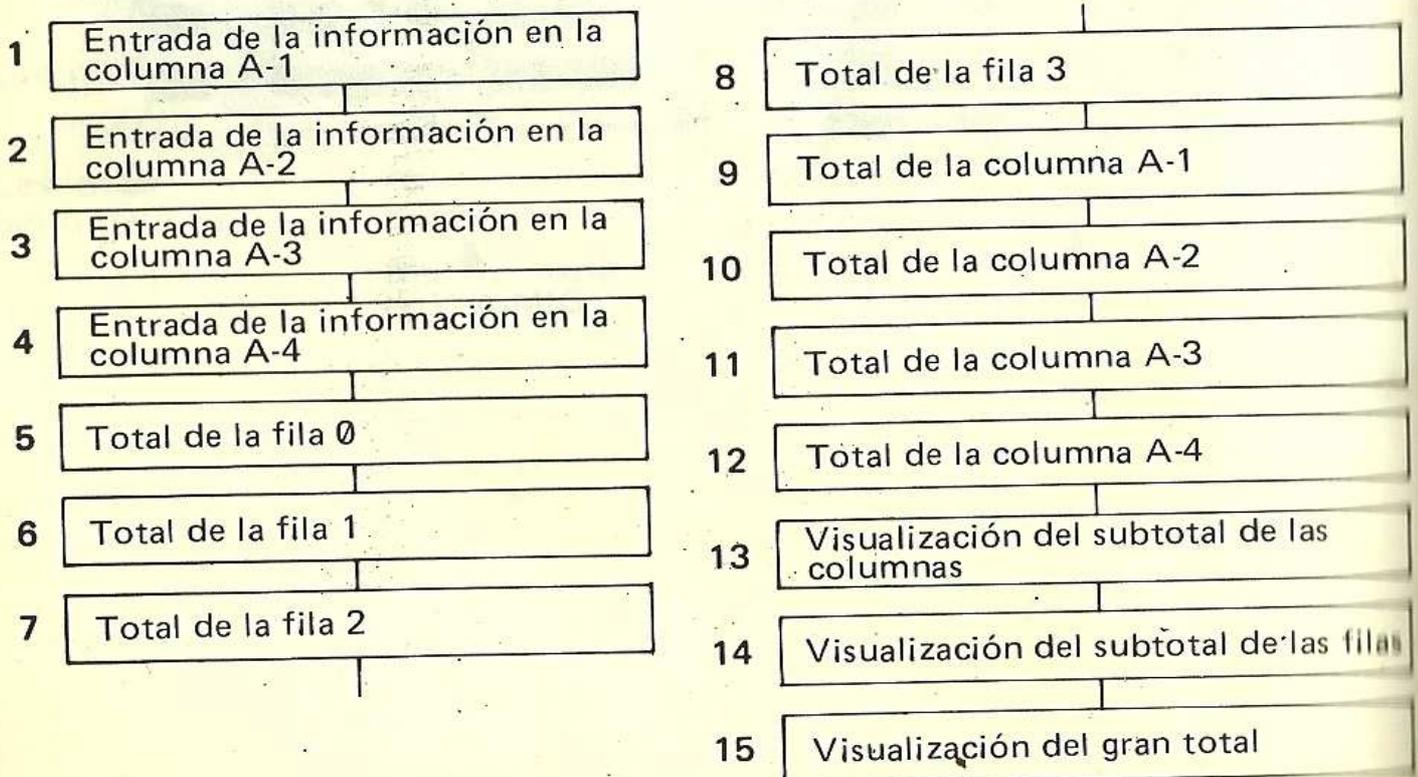


3-15 PROGRAMACION CON MATRICES NUMERICAS

Preparamos un programa que obtenga los subtotales por columna y por fila de una tabla y, al final, el total en ambas direcciones. Se entrará un total de 16 datos, 4 horizontalmente y 4 verticalmente.

| | A-1 | A-2 | A-3 | A-4 | Subtotal |
|----------|-----|-----|-----|-----|----------|
| 0 | 25 | 17 | 3 | 67 | 112 |
| 1 | 19 | 20 | 11 | 58 | 108 |
| 2 | 32 | 15 | 26 | 55 | 128 |
| 3 | 36 | 28 | 29 | 40 | 133 |
| Subtotal | 112 | 80 | 69 | 220 | 962 |

La información es numérica, por lo que prepararemos un programa con una matriz numérica de una dimensión. Adoptemos el nombre A para la variable. Compongamos el programa en base al siguiente diagrama de flujo.



Preparemos una rutina para la entrada de la información en cada columna, tal como se expone en los puntos 1 a 4 del diagrama de flujo.

```

10 CLEAR
20 DIM A!(15)
30 FOR I = 0 TO 3
40 INPUT A!(I)
50 NEXT I
    
```

} Lazo FOR-NEXT

En este programa, se entran datos en los elementos A!(0) hasta A!(3) (columna A-1), pero ¿cómo se entran los datos en las columnas A-2 hasta la A-4 sin cambiar el nombre de la variable? Veamos a continuación qué es lo que estamos tratando de hacer.

| | | | | | |
|----------|-------|-------|--------|--------|----------|
| | → I | | | | Subtotal |
| | A!(0) | A!(4) | A!(8) | A!(12) | B!(0) |
| ↓ | A!(1) | A!(5) | A!(9) | A!(13) | B!(1) |
| J | A!(2) | A!(6) | A!(10) | A!(14) | B!(2) |
| | A!(3) | A!(7) | A!(11) | A!(15) | B!(3) |
| Subtotal | C!(0) | C!(1) | C!(2) | C!(3) | D |

Si observamos la relación de columna a columna llegamos al siguiente patrón.

A!(X) A!(X+4) A!(X+8) A!(X+12)

En base a ello, todo lo que necesitamos es una rutina que sume 4 unidades al contador del lazo cada vez que deseamos pasar a la columna siguiente. Para ello, podemos utilizar el lazo que aparece en las líneas 40 hasta 90, a continuación.

```

10 REM INPUT
20 CLEAR
30 DIM A!(15)
40 FOR J = 0 TO 3
50 FOR I = 0 TO 3
60 PRINT "A-"; J+1; "("; I; ")";
70 INPUT A!(J*4+I)
80 NEXT I
90 NEXT J
    
```

Entrada de datos desplazándose secuencialmente desde la columna A-1.

Entrada de datos desplazándose secuencialmente desde la columna A-1.

Veamos como trabaja. El valor inicial de J es 0. Este valor de J se mantiene durante la tres siguientes pasadas de I. La línea 60 visualiza "A-1 (0)" en la pantalla, ya que tanto J como I son iguales a 0. La línea 70 espera una entrada para A!(0), ya que ~~$3 * 4 + 2 = 14$~~ . (~~$0 * 4 + 0 = 0$~~) En la segunda pasada por I, J sigue siendo igual a 0, pero I aumenta a 1. Por lo tanto, la línea 60 visualiza "A-1 (1)" y la línea 70 espera una entrada para A!(1), ya que $0 * 4 + 1 = 1$. Veamos ahora el case de A!(14). En este caso, J es igual a 3 e I igual a 2. La línea 60 visualiza "A-4(2)" y la línea 70 espera una entrada para A!(14), ya que $3 * 4 + 2 = 14$.

Así se completa la rutina para entrada de la información. Los datos se entran en forma secuencial con el mensaje "A-Nº de columna (vertical)?" La información se almacena en la variable de matriz A!() en la línea 70.

Preparemos a continuación una rutina para obtener el subtotal de las filas.

```

95 DIM B!(3)
100 FOR I=0 TO 3
110 FOR J=0 TO 3
120 B!(I) = B!(I) + A!(J * 4 + I)
130 NEXT J
140 PRINT "B-"; I; TAB(5); B!(I)
150 INPUT "OK"; F$
160 NEXT I
    
```

Cálculo y visualización de los datos de filas

Cálculo y visualización del subtotal desplazándose verticalmente 4 posiciones.

Los datos se almacenan en las variables B!(0) a B!(3) mediante la línea 120 y se visualizan por medio de la línea 140. Para evitar que lo visualizado en pantalla se desplace hacia arriba, la línea 150 interrumpe la presentación hasta que se pulse la tecla .

Preparemos ahora un programa para el subtotal de las columnas.

```

170 DIM C!(3)
180 FOR J=0 TO 3
190 FOR I=0 TO 3
200 C!(J) = C!(J) + A!(J * 4 + I)
210 NEXT I
220 PRINT "C-"; J; TAB(5); C!(J)
230 INPUT "OK"; F$
240 NEXT J
    
```

Cálculo y visualización de los datos de columnas

Cálculo y visualización de subtotal desplazándose verticalmente 4 posiciones.

Este es casi el mismo método que el utilizado para obtener el subtotal de las filas por medio de dos lazos FOR-NEXT con una secuencia de variables diferentes (variables para el control de lazos) I y J.

Calculemos a continuación el total.

```

250 REM TOTAL
260 FOR I=0 TO 3
270 D = D+C I (I)
280 NEXT I
290 PRINT "TOTAL=" ; D
300 END

```

Programación con una matriz bidimensional

El manejo de cada estante donde se almacenan los datos es bastante complicado y dificultoso cuando se usa una matriz de una sola dimensión como en el ejemplo anterior. Preparemos entonces un programa para obtener el mismo resultado, pero esta vez usando una matriz bidimensional, la cual es muy conveniente para entrar datos en ambos sentidos.

1 Inicialización

| | |
|----------------|--|
| 10 ERASE A! | Inicializa la matriz de la variable A! |
| 20 CLS | Borra la pantalla |
| 30 N=4 | Número de posiciones verticales y horizontales |
| 40 DIM A!(N,N) | Declaración de la dimensión (precisión media) |

El número de posiciones en ambos sentidos puede modificarse cambiando el valor de N en la línea 30.

2 Entrada de la información

```

50 FOR I=0 TO N-1
60 FOR J=0 TO N-1
70 PRINT I; "-"; J;
80 INPUT A!(I, J)
90 NEXT J
100 NEXT I
    
```

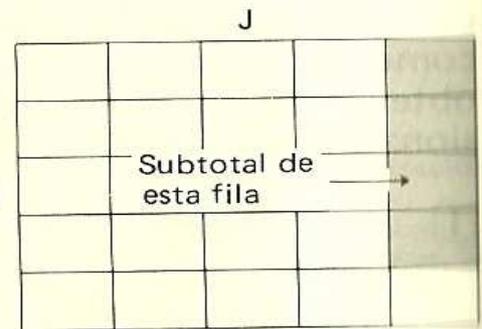
| | → J | | | | Total de las filas |
|-----------------------|-----|-----|-----|-----|--------------------|
| | 28 | 39 | 12 | 54 | 133 |
| I ↓ | 53 | 29 | 55 | 30 | 167 |
| | 28 | 17 | 80 | 53 | 178 |
| | 60 | 31 | 70 | 44 | 205 |
| Total de las columnas | 169 | 116 | 217 | 181 | 683 |

En las líneas 50 y 60 se utiliza N-1, ya que para los totales de las filas y las columnas no se requiere entrada de dato alguno. Obsérvese que los datos se entran en columnas, y no verticalmente como en las matrices unidimensionales.

3 Subtotal de las filas

```

110 FOR I=0 TO N-1
120 FOR J=0 TO N-1
130 A!(I, N)=A!(I, N)+A!(I, J)
140 NEXT J
150 NEXT I
    
```

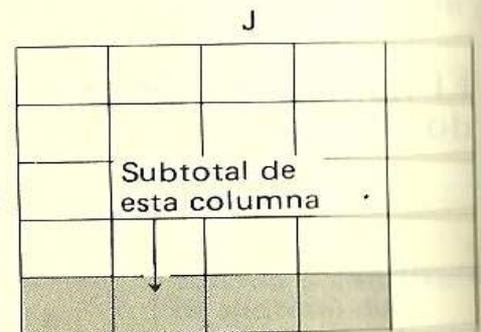


Los subtotales de las columnas se calculan cuatro veces (cuando N=4). Lleve a cabo esta rutina manualmente para corroborar todos los valores asignados a A! ().

4 Subtotal de las columnas

```

160 FOR J=0 TO N
170 FOR I=0 TO N-1
180 A!(N, J)=A!(N, J)+A!(I, J)
190 NEXT I
200 NEXT J
    
```



Los subtotales de las columnas se calculan cuatro veces (cuando N=4). Nuevamente, lleve a cabo esta rutina manualmente para corroborar todos los valores asignados a A! ().

5 Visualización del subtotal de las filas

```

210 FOR I=0 TO N-1
220 PRINT A!(I,N);
230 NEXT I
240 STOP

```



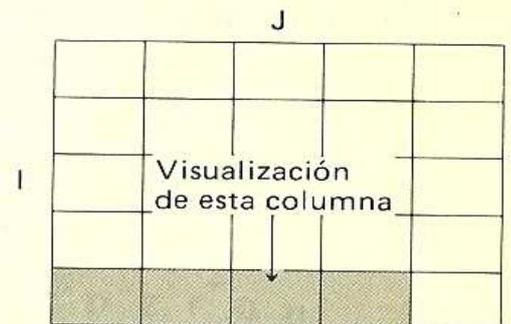
El programa se detiene y permite confirmar cada subtotal horizontal. Luego, la ejecución avanza a la siguiente línea por medio de la pulsación de las teclas **SHIFT** **CONT** **W** **↵**.

6 Visualización del subtotal de las columnas

```

250 FOR J=0 TO N
260 PRINT A!(N,J);
270 NEXT J
280 END

```



Esta rutina visualiza la suma total de los subtotales de todas las columnas. Como el subtotal de la primera columna de la derecha es en realidad el total de los subtotales de las columnas, el resultado es la suma todos los valores entrados en la tabla (o matriz).

Lo más difícil de este programa fue la inclusión de un lazo FOR-NEXT dentro de otro. Esta técnica es muy conveniente; la práctica le ayudará a dominarla. Lo único que debe recordar es que la variable de control aumenta automáticamente por cada ejecución del lazo FOR-NEXT.

7 Ejecución del programa

Al ejecutarse el programa, aparece en la pantalla "0-0?" solicitando la entrada de información. Entre los datos y pulse la tecla . A continuación aparecerá el mensaje "0-1?" solicitando la entrada de la información para la siguiente fila.

Una vez entrada toda la información, se visualizan los subtotales de las filas y el mensaje "STOP P0-240" y se interrumpe el programa. A continuación, la entrada    hace que se visualicen los subtotales de las columnas y el gran total, y que finalice el programa. Intente cambiar los formatos de visualización modificando el programa.

■ Variables de control

En la rutina para los subtotales de las columnas, los valores de las variables de control cambian del siguiente modo.

```

110 FOR I=0 TO N-1
120 FOR J=0 TO N-1
130 A!(I,N)=A!(I,N)+A!(I,J)
    
```

Variables de control

I y J son las variables de control para los lazos FOR-NEXT. Las mismas cambian de 0 hasta 3.

Cuando $I = 0$, $J = 0$ y $N = 4$,
 $A!(0, 4) = A!(0, 4) + A!(0, 0)$.

3-16 VARIABLES MATRICIALES DE CARACTERES

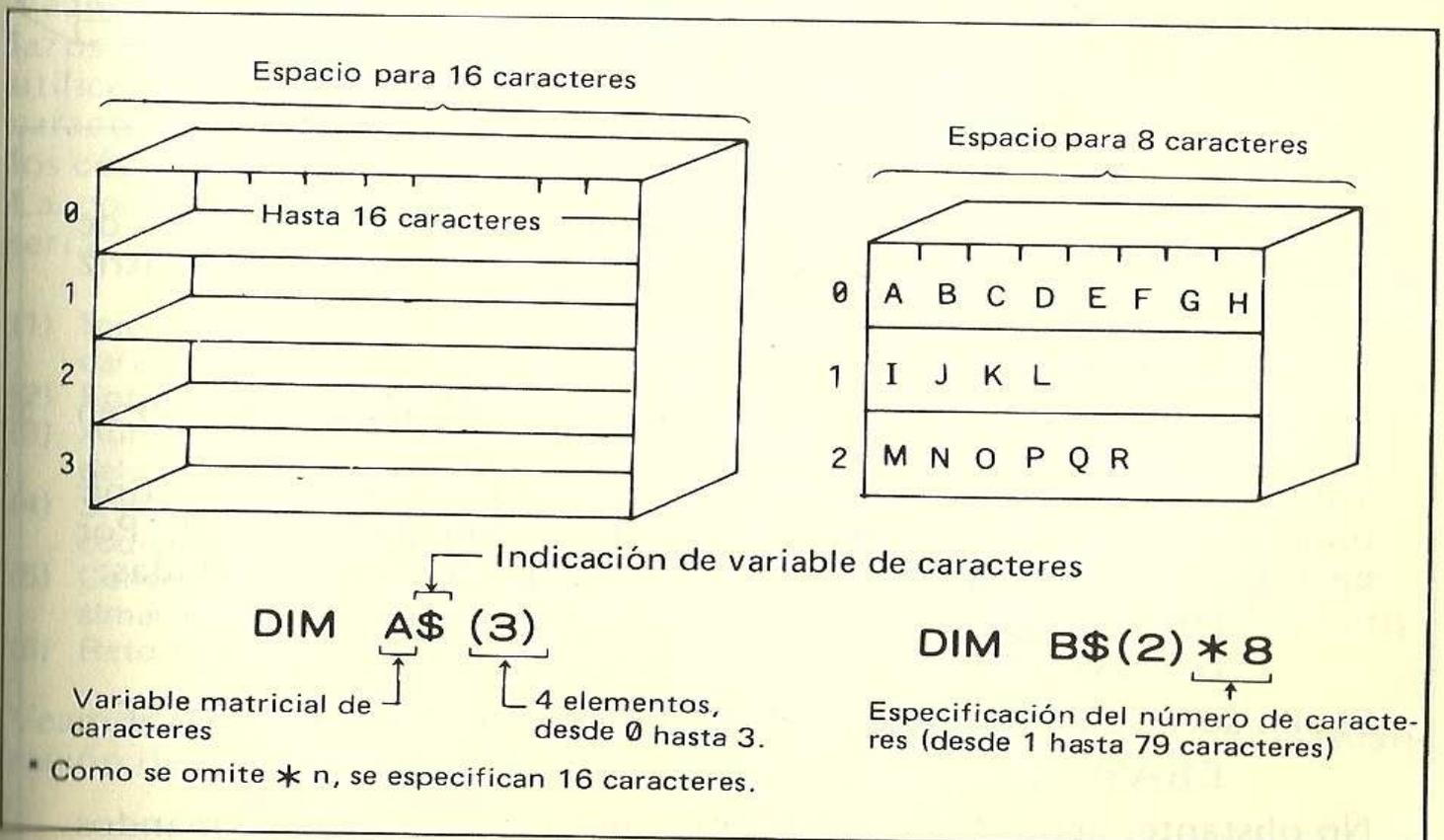
A las matrices unidimensionales y bidimensionales se les puede asignar también caracteres. Como en el caso de las matrices numéricas, con las cuales se puede especificar el tipo de precisión para aprovechar mejor la capacidad de memoria, con las matrices de variables de caracteres se puede especificar la cantidad de caracteres para cada uno de los elementos.

Se desperdiciaría memoria si se asignaran tan sólo 6 caracteres a una variable para la cual se especificaron más caracteres. En contraste, si intenta asignar un número de caracteres mayor que el especificado para la variable, aparecerá un error en la pantalla. Es por ello que resulta importante determinar la cantidad de caracteres que se asignarán a cada uno de los elementos de la matriz, antes de definirse esta última.

La especificación del número de caracteres se lleva a cabo de la siguiente manera.

`DIM A$(i) * n` ($1 \leq n < 79$)

* El número de caracteres especificado es igual a 16 si se omitiera * n.



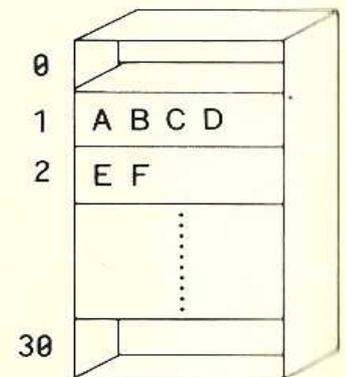
Entremos caracteres en una variable matricial. Lleve a cabo la siguiente operación:

```
CLEAR ↵
DIM H$(30)*5 ↵
H$(1) = "ABCD" ↵
H$(2) = "EF" ↵
```

Al llevarse a cabo estas entradas, los caracteres se almacenan en la variable de matriz del modo ilustrado a la derecha.

Para visualizar lo almacenado, lleve a cabo la siguiente operación:

```
H$(1) [ENTER] ➔ ABCD
H$(2) [ENTER] ➔ EF
```



■ Errores

Aparecerá un error siempre que se intente asignar un número de caracteres mayor que el especificado cuando se definió la matriz mediante la sentencia DIM.

Error ST

En dicho caso, vuelva a entrar la serie de caracteres teniendo en cuenta el límite especificado.

También aparecerá un error (DD en este caso) si se intenta definir una matriz la cual ya ha sido definida con anterioridad. Por ejemplo, este error aparecería si se realizaran las siguientes entradas:

```
DIM A$(20)
DIM A$(30)
```

Si ello sucede, borre la matriz A\$ () mediante el mando ERASE.

```
ERASE A$
```

No obstante, debe tenerse en cuenta que la ejecución de los mandos ERASE y CLEAR borra todos los datos asignados a las variables.

Programación con matrices de caracteres

Ejercicio

Problema

Prepare un programa con 3 matrices en las cuales se puedan almacenar hasta 10 caracteres respectivamente. Los caracteres deberán entrarse por su código. Para pasar de una matriz a la siguiente, deberá entrarse un 0. Deben visualizarse todos los caracteres cuyos códigos fueron entrados una vez finalizada la entrada.

Procedimiento

Como se entrarán 10 caracteres en cada una de las 3 matrices,

```
10 CLEAR
20 DIM N$(2)*10
```

Veamos a continuación la rutina para la entrada de los datos. Prepare tres lazos FOR-NEXT, ya que son tres la matrices que van a utilizarse y utilice otro lazo por medio de una sentencia GOTO para leer los 10 caracteres. Incluya una sentencia INPUT en este lazo para también leer los códigos de los caracteres.

La configuración básica de la rutina para la entrada de la información sería:

- | | | |
|--|-------|---------------------------|
| (1) Inicialización del contador del número de caracteres por variable | | B = 0 |
| (2) Entrada de un código de carácter | | INPUT N |
| (3) Aumento a incrementos de 1 ó del contador del número de caracteres | | B = B + 1 |
| (4) Si el contador es mayor que 10, entra los códigos en la siguiente matriz | | IF B = 10 THEN ~ |
| (5) Conversión de un código en un carácter y su almacenamiento en una variable de matriz | | N\$(I) = NR(I) + CHR\$(N) |
| (6) Retorno a (2) | | GOTO ~ |

Veamos a continuación la rutina para visualización del resultado. Visualización de N\$(0) a N\$(2) en la misma línea.

```
100 FOR I = 0 TO 2
110 PRINT N$(I); "_"; ..... Se entra en I de 0 a 2 en forma
120 NEXT I                               secuencial.
```

Solución

```

10 CLEAR : CLS
20 DIM N$(2)*10
30 FOR I=0 TO 2:B=0
40 PRINT "N$( "; I; ")"; : INPUT " No. "; N
45 IF N>255 THEN 40
50 IF N=0 THEN 90
60 B=B+1; IF B=10 THEN BEEP :GOTO 90
70 N$(I)=N$(I)+CHR$(N)
80 GOTO 40
90 NEXT I
100 FOR I=0 TO 2
110 PRINT N$(I); " ";
120 NEXT I
130 END
    
```

Al ejecutarse este programa, aparece el mensaje "N\$(0) _ _ No.? _".
 Entre, entonces, el código de caracteres. El número entre paréntesis en
 "N\$(I)" aumenta de a 1 por cada entrada 0
 Entre los siguientes códigos y vea lo que sucede.

| | | | | | | | | | | | | | | | | | | | | | | |
|----|--------------------------|--------------|----|--------------------------|--------------|----|--------------------------|--------------|----|--------------------------|--------------|----|--------------------------|--------------|----|--------------------------|--------------|--------------------------|--------------------------|---|--------------------------|--------------|
| 67 | <input type="checkbox"/> | ^C | 65 | <input type="checkbox"/> | ^A | 83 | <input type="checkbox"/> | ^S | 73 | <input type="checkbox"/> | ^I | 79 | <input type="checkbox"/> | ^O | 0 | <input type="checkbox"/> | 67 | <input type="checkbox"/> | ^C | | | |
| 79 | <input type="checkbox"/> | ^A | 77 | <input type="checkbox"/> | ^M | 80 | <input type="checkbox"/> | ^D | 85 | <input type="checkbox"/> | ^N | 84 | <input type="checkbox"/> | ^T | 69 | <input type="checkbox"/> | ^E | | | | | |
| 82 | <input type="checkbox"/> | ^R | 0 | <input type="checkbox"/> | | 80 | <input type="checkbox"/> | ^P | 66 | <input type="checkbox"/> | ^B | 55 | <input type="checkbox"/> | ⁷ | 48 | <input type="checkbox"/> | ⁷ | 48 | <input type="checkbox"/> | 0 | <input type="checkbox"/> | ^D |

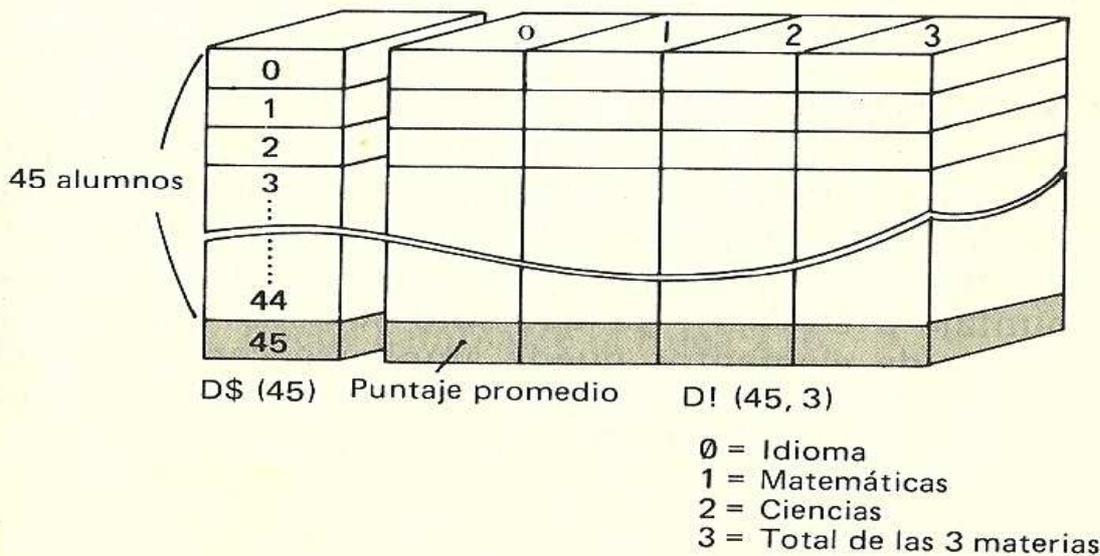
3-17 COMBINACION DE MATRICES NUMERICAS Y DE CARACTERES

En muchos casos, cuando se prepara una tabla para la entrada de información, los datos entrados son números y caracteres combinados. Es por ello que ambos deben procesarse en forma simultanea a modo de un mismo grupo.

Cuando se almacenan, por ejemplo, puntajes de exámenes y nombres de alumnos, o nombres de artículos, sus cantidades y precios, la información incluye tanto caracteres como números.

Veamos un simple ejemplo de cómo procesar los puntajes de alumnos por materias. Primero se necesita una matriz unidimensional de caracteres, para almacenar los nombres. Luego, utilizaremos una matriz numérica bidimensional para almacenar el puntaje total de tres materias (idioma, matemáticas y ciencias) con el nombre de cada alumno.

Se podrá, para ello, adoptar el modelo ilustrado a continuación:



La información se entra con el siguiente formato.

| Nombre | Idioma | Matemáticas | Ciencias | Puntaje total |
|------------------|--------|-------------|----------|---------------|
| A. Y | 50 | 60 | 75 | |
| K. K | 83 | 71 | 70 | |
| S. O | 60 | 63 | 40 | |
| Puntaje promedio | | | | |

1 Entrada del nombre

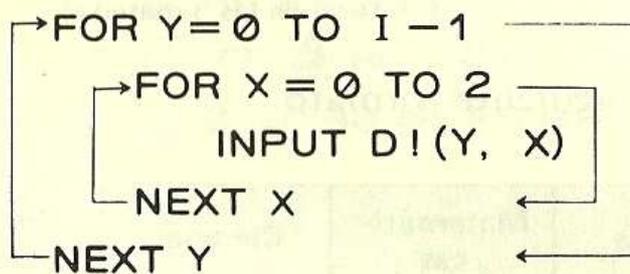
```

10 ERASE D$,D!: CLS
20 DIM D$(45)*10,D!(45,3)
30 I=0
40 INPUT "NAME ";D$(I)
50 IF D$(I)="END" THEN 80
60 I=I+1:IF I=45 THEN 80
70 GOTO 40
    
```

Los nombres se almacenan en D\$(0) hasta D\$(44); el puntaje promedio se almacena en D\$(45). Para salir del lazo, se entra "END". Las líneas 40 hasta 70 son, como podrá observarse, para la entrada de los nombres.

2 Entrada de valores numéricos

He a continuación una rutina que permite la entrada de valores numéricos en una matriz bidimensional. La línea 90 visualiza el nombre del estudiante, almacenado en D\$(Y), mientras que la línea 120 da entrada al puntaje de Idioma en D!(0,0) (cuando x igual a 0). Luego, la línea 130 suma el puntaje a D!(Y,3), ya que sólo se están procesando tres materias. El número de elementos puede ampliarse cambiando el 3 de D!(Y,3) al "número de elementos +1". Los valores de X e Y para D!(Y,X) se determinan en el lazo FOR-NEXT.



Tendría, entonces, el siguiente formato de entrada:

```

80 FOR Y=0 TO I-1
90 PRINT D$(Y)
100 FOR X=0 TO 2
110 PRINT X
120 INPUT " ";D!(Y,X)
130 D!(Y,3)=D!(Y,3)+D!(Y,X)
140 NEXT X
150 NEXT Y

```

→ X

Tabla de puntajes

| ↓ Y | Idioma (0) | Matemáticas (1) | Ciencias (2) | Total |
|-----|---------------|--------------------|-----------------|-------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

3 Puntaje promedio para cada materia

Sea cual fuere la cantidad de nombres que se entren (contados por I), $D!(45,X)$ es el elemento destinado a almacenar el puntaje promedio de cada materia. En la línea 180, se asigna a $D!(45,X)$ los puntos acumulados en cada materia. En la línea 200, se asigna el promedio a D, mientras que en la línea 20 se redondea a un dígito decimal el valor asignado a D, y dicho resultado se vuelve a almacenar en $D!(45,X)$. Se utiliza esta función ya que se procesan resultados de pruebas. Para otras finalidades, puede cambiarse la línea 210.

```

160 FOR X=0 TO 3
170 FOR Y=0 TO I-1
180 D!(45,X)=D!(45,X)+D!(Y,X)
190 NEXT Y
200 D=D!(45,X)/I
210 D!(45,X)=INT(D*10+0.5)/10
220 NEXT X

```

4 Visualización del puntaje total por nombre

Se visualiza también el puntaje total por nombre. Al pulsarse la tecla  en la línea 260, el programa avanza al siguiente nombre. Los puntajes totales para cada nombre se procesan mediante la variable de control de lazo Y. Por lo tanto, lo siguiente se cumple siempre:

```

Nombre . . . . . D$(Y)
Puntaje total . . . . . D!(3, Y)

230 FOR Y=0 TO I-1
240 PRINT D$(Y); " T=";
250 PRINT D!(Y,3)
260 K$=INKEY$: IF K$="" THEN 260
270 NEXT Y
    
```

5 Visualización del puntaje promedio

La visualización del puntaje promedio para cada materia se lleva a cabo visualizando secuencialmente los datos para cada valor de X (de 0 hasta 3) en D!(X, 45).

```

280 FOR X=0 TO 3
290 PRINT "AVE.=";
300 PRINT D!(45, X)
310 K$=INKEY$: IF K$="" THEN 310
320 NEXT X
330 END
    
```

Ejecute este programa y entre algunos datos de prueba. La visualización sería, entonces, del siguiente modo:

| | |
|-------------|---|
| A. Y T= 185 | } El puntaje total por nombre aparece en forma secuencial cada vez que se presiona la tecla  . |
| K. K T= 224 | |
| S. O T= 163 | |

-  AVE. = 64.3 — Puntaje promedio de Idioma (0)
-  AVE. = 64.7 — Puntaje promedio de Matemáticas (1)
-  AVE. = 61.7 — Puntaje promedio de Ciencias (2)
-  AVE. = 190.7 — Promedio acumulativo del total por persona

3-18 FUNCIONES ESTADÍSTICAS

Las funciones para cálculos estadísticos son esenciales en el análisis y proyecciones de información comercial y científica. La PB-770 pone a su disposición las funciones estadísticas citadas a continuación, las cuales simplificarán el cálculo de coeficientes de correlación, valores estimados y otros.

* STAT LIST  visualiza los nombres y los valores de las estadísticas básicas (indicadas en la lista a continuación por un asterisco). Cada pulsación posterior de la tecla  interrumpe y reanuda dicha visualización. STAT LLIST  hace lo propio, pero en lugar de visualizar en la pantalla, imprime la información en la impresora.

| Formato | | Función |
|--------------------------|--------------|---|
| CNT* | n | Número de datos que han de procesarse |
| SUMX* | Σx | Suma de los datos x |
| SUMY* | Σy | Suma de los datos y |
| SUMXY* | Σxy | Suma de los productos de los datos x e y |
| SUMX2* | Σx^2 | Suma de los cuadrados de los datos x |
| SUMY2* | Σy^2 | Suma de los cuadrados de los datos y |
| MEANX | \bar{x} | Media de los datos x |
| MEANY | \bar{y} | Media de los datos y |
| SDX | x_{on-1} | Desviación estándar por muestreo de los datos x $\sqrt{\frac{n \Sigma x^2 - (\Sigma x)^2}{n(n-1)}}$ |
| SDY | y_{on-1} | Desviación estándar por muestreo de los datos y $\sqrt{\frac{n \Sigma y^2 - (\Sigma y)^2}{n(n-1)}}$ |
| SDXN | x_{on} | Desviación estándar total de los datos x $\sqrt{\frac{n \Sigma x^2 - (\Sigma x)^2}{n^2}}$ |
| SDYN | y_{on} | Desviación estándar total de los datos y $\sqrt{\frac{n \Sigma y^2 - (\Sigma y)^2}{n^2}}$ |
| LRA | A | Término constante de la regresión lineal $\frac{\Sigma y - LRB \cdot \Sigma x}{n}$ |
| LRG | B | Coefficiente de regresión lineal $\frac{n \cdot \Sigma xy - \Sigma x \cdot \Sigma y}{n \cdot \Sigma x^2 - (\Sigma x)^2}$ |
| COR | r | Coefficiente de correlación $\frac{n \Sigma xy - \Sigma x \cdot \Sigma y}{\sqrt{[n \Sigma x^2 - (\Sigma x)^2][n \Sigma y^2 - (\Sigma y)^2]}}$ |
| EOX (expresión numérica) | \hat{x} | Valor estimado (de x en base a y) $EOX(y_0) = \frac{y_0 - LRA}{LRB}$ |
| EOY (expresión numérica) | \hat{y} | Valor estimado (de y en base a x) $EOY(x_0) = LRA + x_0 \cdot LRB$ |

STAT CLEAR debe entrarse para borrar el área de memoria para cálculos estadísticos antes de entrar nueva información.

■ **Entrada de datos estadísticos**

• Una variable

• Una variable

Entrada de un sólo dato STAT dato
 Entrada múltiple de un mismo dato STAT dato frecuencia

• Pares de variables

Entrada de un sólo dato STAT x dato y dato

Entrada múltiple de un mismo dato

..... STAT x dato y dato frecuencia

Los datos se entran mediante la tecla ; en cambio, los resultados de desviaciones estándar se obtienen pulsando la tecla **ENTER**. Cualquier equivocación que se cometa provocará la aparición de un error SN.

Ejemplo

En la tabla se representan los embarques de los artículos x e y durante un período de 5 días. Calcule la desviación estándar y determine la varianza de los embarques.

| Fecha \ Artículo | 4 | 5 | 6 | 7 | 8 |
|------------------|---|---|---|---|---|
| x | 2 | 2 | 5 | 8 | 8 |
| y | 1 | 5 | 5 | 5 | 9 |

Operación:

STAT CLEAR

STAT 2 1 STAT 2 5 STAT 5 5

STAT 8 5 STAT 8 9

STAT LIST (Las estadísticas básicas aparecen automáticamente.)

MEANX
 MEANY
 SDXN
 SDYN

| | |
|-------------|---|
| CNT 5 | Número de datos |
| SUMX 25 | Suma de los datos x |
| SUMY 25 | Suma de los datos y |
| SUMXY 149 | Suma de los products de los datos x e y |
| SUMX2 161 | Suma de los cuadrados de los datos x |
| SUMY2 157 | Suma de los cuadrados de los datos y |
| Ready P0 | |
| 5 | Media de los datos x |
| 5 | Media de los datos y |
| 2.683281573 | Desviación estándar total de los datos x |
| 2.529822128 | Desviación estándar total de los datos y |

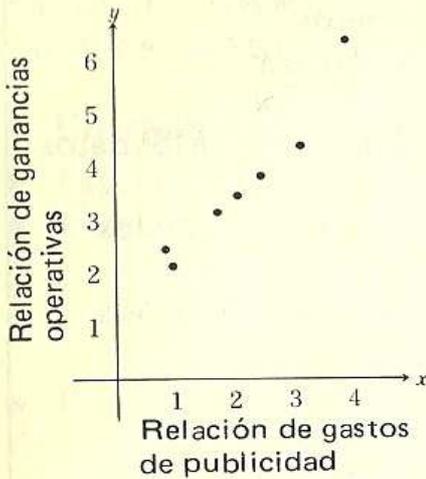
De acuerdo a estos resultados estadísticos, la desviación estándar del artículo x es mayor que la de y , a pesar de que las sumas y medias son iguales para ambos datos. Puede decirse, por lo tanto, que la varianza de los embarques es mayor que x .

Use los datos siguientes para calcular la regresión y determinar el coeficiente de correlación y el valor estimado.

Ejemplo

La tabla a continuación muestra la relación de gastos de publicidad (gastos de publicidad/gastos operativos $\times 100$) y la relación de ganancias operativas (ganancias operativas/ventas $\times 100$) para una cadena de 7 almacenes.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--|-----|-----|-----|-----|-----|-----|-----|
| Relación de gastos de publicidad (x) | 0.8 | 2.1 | 2.5 | 1.8 | 3.1 | 4.0 | 1.0 |
| Relación de ganancias operativas (y) | 2.5 | 3.4 | 3.7 | 3.2 | 4.3 | 6.3 | 2.3 |



Use la tabla para trazar un diagrama de dispersión. Observando el diagrama de dispersión, puede decirse que la ganancia es directamente proporcional a la cantidad destinada a la publicidad. La línea que une los puntos en el diagrama se denomina curva de regresión y, como es casi rectilínea, se la llama regresión lineal. Las curvas de regresión pueden ser logarítmicas, exponenciales o potenciales; la selección del tipo de curva depende de la relación entre los datos x e y . Debe tenerse en cuenta que los límites del coeficiente de correlación (r) son $-1 \leq r \leq 1$. La correlación es positiva cuando $0 \leq r \leq 1$ y negativa cuando $-1 \leq r \leq 0$ y no hay correlación alguna cuando r es igual a 0 .

Procedamos con la entrada de los datos para las 7 almacenes.

Operación:

STAT CLEAR

STAT 0.8 2.5 STAT 2.1 3.4 STAT 2.5 3.7

STAT 1.8 3.2 STAT 3.1 4.3 STAT 4.0 6.3

STAT 1.0 2.3

LRA

LRB

COR

1.174221646 Término constante de la regresión lineal (A)

1.142512973 Coeficiente de regresión lineal (B)

0.9628252383 Coeficiente de correlación (r)

El coeficiente de correlación (r) indica que entre x e y hay una correlación positiva. Averigüemos a continuación cuál sería la relación de gastos de publicidad para obtener una relación de ganancias operativas del 5,7% y cuál la reacción de ganancias operativas si la relación de gastos de publicidad fuere del 4,5%.

EOX5.7

EOY4.5

3.961248986

6.315530022

■ Regresión logarítmica, exponencial y potencial

Estos resultados indican que para una relación de ganancias operativas del 5,7% se requiere una relación de gastos de publicidad de un 3,96%, mientras que de una relación de gastos de publicidad del 4,5% se puede lograr una relación de ganancias operativas del 6,32%.

| Año (x) | Resultados (y) |
|---------|----------------|
| '79 | 5.810 |
| '80 | 5.637 |
| '81 | 6.736 |
| '82 | 7.938 |
| '83 | 8.169 |

● Regresión logarítmica

La fórmula de regresión logarítmica es $y = A + B \cdot \ln x$.

Por cada dato x , se entra su logaritmo; el dato y se entra como está. $\Sigma \ln x$, $\Sigma (\ln x)^2$ y $\Sigma \ln xy$ se obtienen para Σx , Σx^2 y Σxy , respectivamente.

Operación:

STAT CLEAR

STAT LOG54 5810 STAT LOG55 5637

STAT LOG56 6736 STAT LOG57 7938

STAT LOG58 8169

LRA ENTER

LRB ENTER

COR ENTER

COR SHIFT 2 ENTER

-151086.8602

Término constante de la regresión (A)

39240.6409

Coefficiente de regresión (B)

0.9461867989

Coefficiente de correlación (r)

0.8952694585

Coefficiente de decisión (r^2)

● Regresión exponencial

La fórmula de regresión exponencial es $y = A \cdot e^{B \cdot x}$ ($\ln y = \ln A + B \cdot x$).

Por cada dato y se entra su logaritmo; el dato x se entra como está. $\ln A$, $\Sigma \ln y$ y $\Sigma x \cdot \ln y$ se obtienen para A, SUMY y SUMY2, respectivamente.

Operación:

STAT CLEAR

STAT 54 LOG5810 STAT 55 LOG5637

STAT 56 LOG6736 STAT 57 LOG7938

STAT 58 LOG8169

EXP LRA

LRB

COR

21.93154256

Término constante de la regresión (A)

0.102384121

Coefficiente de regresión (B)

0.9442661562

Coefficiente de correlación (r)

● **Regresión potencial**

La fórmula de regresión potencial es $y = A \cdot x^B$ ($\ln y = \ln A + B \cdot \ln x$). Para cada datos x e y se entran sus logaritmos correspondientes. $\ln A$, $\sum \ln x$, $\sum (\ln x)^2$, $\sum \ln y$, $\sum (\ln y)^2$ y $\sum (\ln x \cdot \ln y)$ se obtienen para A , $\sum x$, $\sum x^2$, $\sum y$, $\sum y^2$ y $\sum xy$, respectivamente.

Operación:

STAT CLEAR

STAT LOG54 LOG5810 STAT LOG55 LOG5637

STAT LOG56 LOG6736 STAT LOG57 LOG7938

STAT LOG58 LOG8169

EXP LRA

LRB

COR

6.651154824E-07

5.725355325

0.9433168782

3-19 USO DE LOS CARACTERES GRAFICOS

En la PB-770 pueden usarse los caracteres gráficos que se muestran en la Tabla de Códigos de Caracteres de la página 328.

(1) Función CHR\$

La palabra "MAMA", por ejemplo, puede visualizarse en la pantalla usando el siguiente mando.

```
PRINT "MAMA" 
```

La función CHR\$, también, puede utilizarse para lograr el mismo resultado.

```
PRINT CHR$(77); CHR$(65); CHR$(77); CHR$(65) 
```

Como podrá observarse, la especificación de los códigos de caracteres de cada una de las letras permite, junto con la función CHR\$, presentar las mismas en la pantalla. Los demás caracteres pueden usarse de la siguiente manera.

La tabla a continuación muestra algunos de estos códigos.

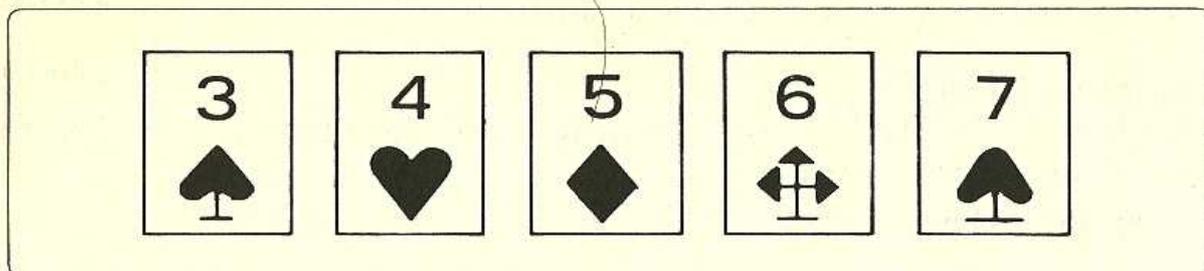
| Código | Pantalla | Código | Pantalla |
|---------|----------|--------|----------|
| 30, 160 | o | 164 | ˘ |
| 161 | ┌ | 165 | . |
| 162 | └ | 176 | — |
| 163 | ┘ | | |

(2) Símbolos gráficos

El programa a continuación visualiza 5 naipes en la pantalla.

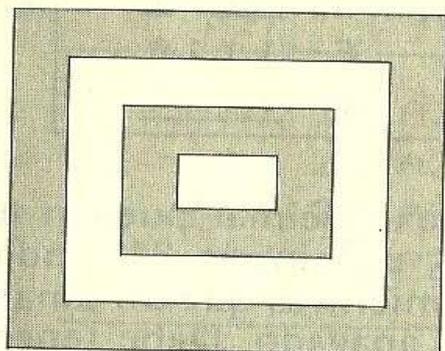
```
10 CLS
20 FOR I=0 TO 4
30 LOCATE I*3,1:PRINT I+3
40 LOCATE I*3+1,2:PRINT CHR$(232+I MOD 4)
50 DRAW(I*24+4,4)-(I*24+20,4)-(I*24+20,26)-(I*24+4,26)-(I*24+4,4)
60 NEXT I
70 IF INKEY$="" THEN 70
80 END
```

Resultado de la ejecución



3-20 VISUALIZACION DE PATRONES GRAFICOS

Además de los caracteres que se encuentran en la Tabla de Códigos de Caracteres, la PB-770 puede visualizar otros patrones gráficos. Probemos visualizar en la pantalla el patrón gráfico mostrado a continuación.



Formato visualizado en pantalla

```
AB$="FF18DB5A5ADB18FF"   
PRINT $AB$ 
```

La ejecución de la operación arriba citada debe hacer aparecer en la pantalla el patrón gráfico en cuestión.

(1) Proceso de visualización

La visualización en la pantalla se produce adjuntando el signo \$ a una variable de caracteres (en este caso AB\$). Esta misma operación puede utilizarse dentro de un programa.

La serie que se asigna a AB\$ se compone de valores hexadecimales (en este sistema numérico, se utilizan los números del 0 al 9 y las letras de la A hasta la F). Esta misma configuración puede utilizarse para obtener diferentes tipos de caracteres gráficos.

(2) Configuración de los patrones

Los gráficos se producen mediante el uso del patrón hexadecimal ilustrado a la derecha. Un patrón se compone de 8 x 8 puntos (de la pantalla), o sea un total de 64 puntos. Como podrá también observarse, los puntos están dispuestos en 16 grupos de 4 puntos cada uno. Estos 16 grupos se numeran de acuerdo al sistema hexadecimal previamente citado.

| | | | | | | | | |
|---|--|--|--|--|--|--|--|---|
| ① | | | | | | | | ② |
| ③ | | | | | | | | ④ |
| ⑤ | | | | | | | | ⑥ |
| ⑦ | | | | | | | | ⑧ |
| ⑨ | | | | | | | | ⑩ |
| ⑪ | | | | | | | | ⑫ |
| ⑬ | | | | | | | | ⑭ |
| ⑮ | | | | | | | | ⑯ |

(3) Designación de cada punto

Los puntos se especifican mediante el uso de valores hexadecimales. Veamos a continuación los equivalentes binarios de los 16 valores hexadecimales.

| ■ Hexadecimal → Binario | | | |
|-------------------------|-------------|-------------|-------------|
| 0 → 0 0 0 0 | 4 → 0 1 0 0 | 8 → 1 0 0 0 | C → 1 1 0 0 |
| 1 → 0 0 0 1 | 5 → 0 1 0 1 | 9 → 1 0 0 1 | D → 1 1 0 1 |
| 2 → 0 0 1 0 | 6 → 0 1 1 0 | A → 1 0 1 0 | E → 1 1 1 0 |
| 3 → 0 0 1 1 | 7 → 0 1 1 1 | B → 1 0 1 1 | F → 1 1 1 1 |

La tabla de arriba muestra cómo los valores hexadecimal pueden expresarse en binarios de 4 dígitos. En la pantalla, un binario "1" indica que el punto que le corresponde está encendido, mientras que un binario "0" indica que el punto que le corresponde está apagado. Debe tenerse en cuenta, no obstante, que la PB-770 cuenta los puntos de izquierda a derecha, o sea, desde 0 hasta 3. Lo que significa, que ESTE SISTEMA DE CUENTA ES A LA INVERSA DEL SISTEMA NUMERICO BINARIO. En la tabla a continuación se muestran los 16 patrones posibles con los grupos de 4 puntos. La especificación de los valores hexadecimales apropiados hará que se produzcan los patrones correspondientes en la pantalla.

| Patrón | Valor hexadecimal |
|--------|-------------------|
| | 0 |
| | 1 |
| | 2 |
| | 3 |
| | 4 |
| | 5 |
| | 6 |
| | 7 |

| Patrón | Valor hexadecimal |
|--------|-------------------|
| | 8 |
| | 9 |
| | A |
| | B |
| | C |
| | D |
| | E |
| | F |

(4) Precauciones

En la definición de patrones deben tenerse en cuenta las siguientes consideraciones.

1. La ausencia del símbolo \$ inmediatamente antes que la variable de caracteres hará que aparezca en la pantalla la serie de valores asignados, en lugar del patrón gráfico que se pretendía crear.
2. La serie de valores que se asignan a la variable debe tener, sin falta, 16 caracteres (ni más, ni menos). En caso contrario, el patrón no aparecerá en la pantalla. El mando LPRINT no tiene efecto alguno sobre estos patrones.

(5) Definición de caracteres como "αβγ"

El programa a continuación ha sido incluido para que Vd. se familiarice con la definición de caracteres. Su ejecución hace que aparezcan los caracteres "αβγ" en la pantalla.

```

10 A1$="000442A21119E600"
20 A2$="C12222A12222E320"
30 A3$="002241C040606040"
40 PRINT $A1$;$A2$;$A3$;

```

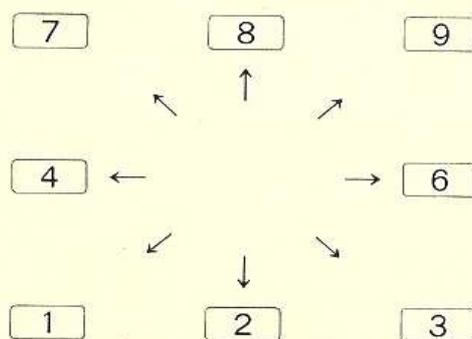
(6) Creación de patrones en la pantalla

Convertir patrones gráficos (manuscritos) en caracteres mediante la definición de valores hexadecimales es un tanto complicado. El siguiente programa permite crear patrones gráficos en la pantalla de la PB-770, para obtener luego la serie de valores hexadecimales que corresponden al patrón trazado.

Uso

Entre el programa de la página 96 en la PB-770. Después de haber confirmado que no se haya cometido error alguno en la entrada del programa, ejecútelo entrando RUN  desde el teclado.

Al hacerlo, aparece en el extremo izquierdo de la pantalla una matriz de 8 x 8 puntos. La indicación "ON" (encendido) u "OFF" (apagado) en la parte derecha de la pantalla indica el estado del punto donde se encuentra ubicado el cursor. Para cambiar el estado del punto en la posición donde se encuentra el cursor (de ON a OFF o viceversa), se utiliza la tecla . El desplazamiento del cursor se controla por medio de las siguientes teclas.



Una señal audible confirma la pulsación de cualquiera de las teclas utilizadas para mover el cursor. Pruebe desplazando el cursor dentro de la matriz y apagando o encendiendo los puntos de la misma. Cree un

patrón gráfico cualquiera. Una vez formado el patrón, presione la tecla **CLS**. Al hacerlo, aparecerá en la pantalla la serie de valores hexadecimales correspondientes al patrón gráfico creado. Tome nota de estos valores y regístrelos en una variable de caracteres. Por último, visualice el patrón creado combinando la variable a la cual asignó los valores obtenidos con la sentencia **PRINT\$**.

```
10 CLS
20 X=2:Y=2:F=0
30 FOR I=0 TO 32 STEP 4
40 DRAW(I,0)-(I,31)
50 IF I=0 THEN 70
60 DRAW(0,I)-(31,I)
70 NEXT I
80 LOCATE 11,3:PRINT "OFF";
90 BEEP
100 K$=INKEY$:L=ASC(K$):N=VAL(K$)
110 IF L=12 THEN 210
120 DRAW(X,Y)
130 DRAWC(X,Y)
140 IF F=1 THEN DRAW(X,Y):DRAW((X-2)/4
    +100,(Y-2)/4):GOTO 160
150 DRAWC((X-2)/4+100,(Y-2)/4)
160 IF K$="" THEN 100 ELSE BEEP
170 LOCATE 11,3
180 IF L=46 THEN IF F=0 THEN F=1:PRINT
    "ON "; ELSE F=0:PRINT "OFF";
190 IF N>0 THEN IF N<10 THEN IF N<>5 T
    HEN GOSUB (N+4)*100
200 GOTO 100
210 AA$="":BEEP
220 FOR K=0 TO 7
230 Z=0
240 FOR I=0 TO 3
250 IF POINT(I+100,K)<>0 THEN Z=Z+2^I
260 NEXT I
270 AA$=AA$+RIGHT$(HEX$(Z),1)
```

```
280 Z=0
290 FOR I=4 TO 7
300 IF POINT(I+100,K)<>0 THEN Z=Z+2^(I
    -4)
310 NEXT I
320 AA$=AA$+RIGHT$(HEX$(Z),1)
330 NEXT K
340 CLS :BEEP :LOCATE 10,0:PRINT $AA$
350 PRINT " $";AA$;" $"
360 PRINT " OK?(Push any key)";
370 K$=INKEY$:IF K$="" THEN 370 ELSE 1
    0
380 END
500 X=X-4:Y=Y+4:IF X<0 THEN X=30
510 IF Y>30 THEN Y=2
520 RETURN
600 Y=Y+4:IF Y>30 THEN Y=2
610 RETURN
*700 X=X+4:Y=Y+4:IF X>30 THEN X=2
→710 IF Y>30 THEN Y=2
720 RETURN
800 X=X-4:IF X<0 THEN X=30
810 RETURN
1000 X=X+4:IF X>30 THEN X=2
1010 RETURN
1100 X=X-4:Y=Y-4:IF X<0 THEN X=30
1110 IF Y<0 THEN Y=30
1120 RETURN
1200 Y=Y-4:IF Y<0 THEN Y=30
1210 RETURN
1300 X=X+4:Y=Y-4:IF X>30 THEN X=2
1310 IF Y<0 THEN Y=30
1320 RETURN
```

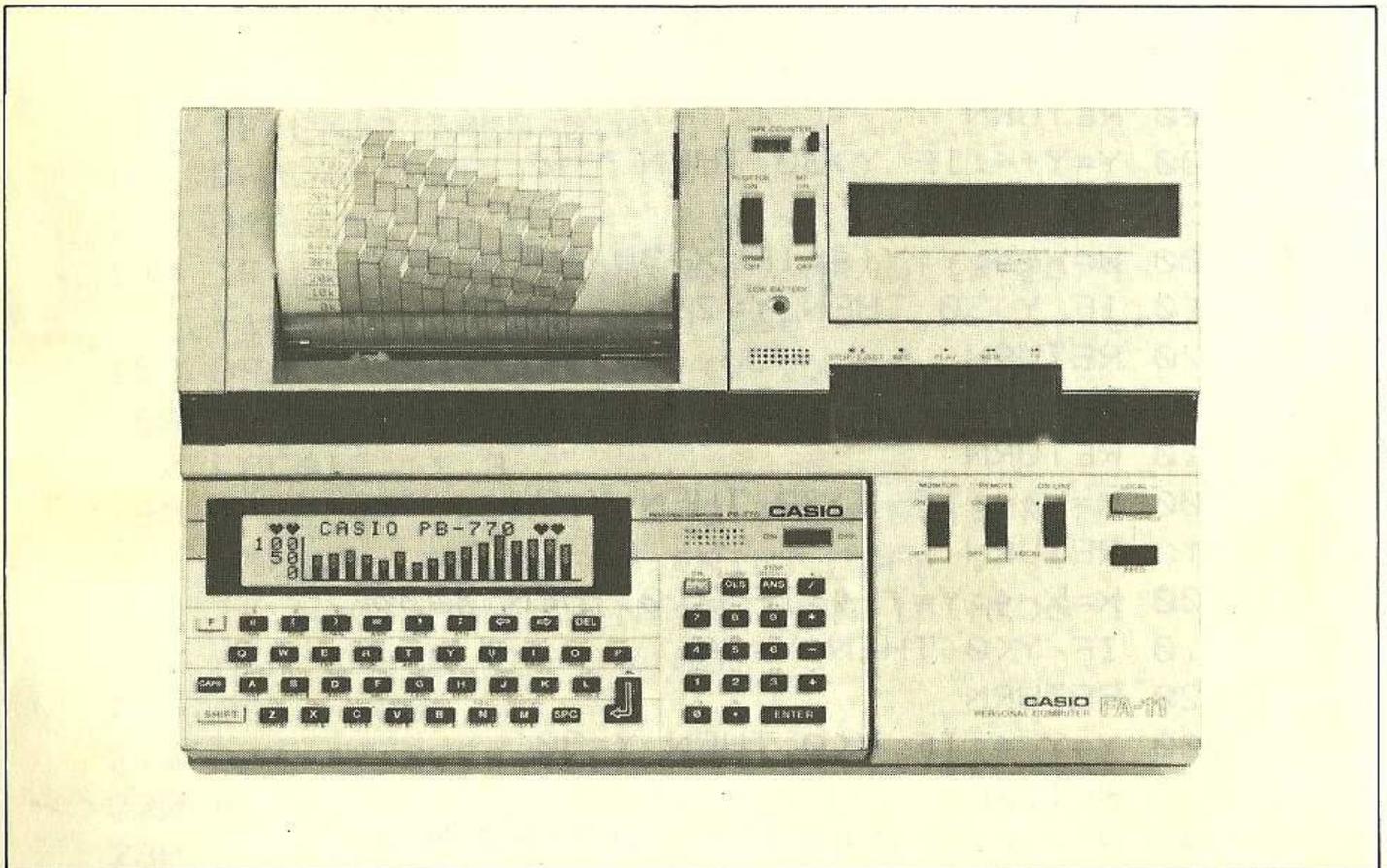
3-21 FUNCIONES GRAFICAS DE LA PB-770

La amplia pantalla de cristal líquido de la PB-770 se compone de 160 por 32 puntos y acomoda 20 caracteres por 4 líneas.

Con la BP-770, Vd. podrá trazar precisos gráficos mediante el uso de simples mandos.

A la PB-770 se puede conectar, además, una impresora por coordenadas de 114mm de anchura con 4 colores e interface para cassette (FA-10). Como en papel de 114mm de anchura se pueden escribir hasta 80 caracteres por línea, esta impresora puede usarse casi del mismo modo que una impresora por coordenadas de mayor escala. La sofisticadas funciones gráficas de la PB-770 le permitirán hacer uso total de la pantalla. Al comienzo podrá parecer confuso, pero con el tiempo y la experiencia se acostumbrará.

Impresora por coordenadas con grabadora de cassette e interface para cassette (FA-11)



3-22 MANDOS GRAFICOS Y COORDENADAS DE LA PANTALLA

Los gráficos se trazan en la pantalla mediante la unión de series de puntos. Por lo tanto, lo único que se requiere para trazar gráficos en la pantalla son las posiciones de los puntos que se desean encender o apagar. La PB-770 hace uso de dos mandos para encender y apagar puntos en las posiciones especificadas de la pantalla.

DRAW Enciende puntos o traza líneas

DRAWC . . . Borra puntos o una líneas

Se provee también la siguiente función gráfica:

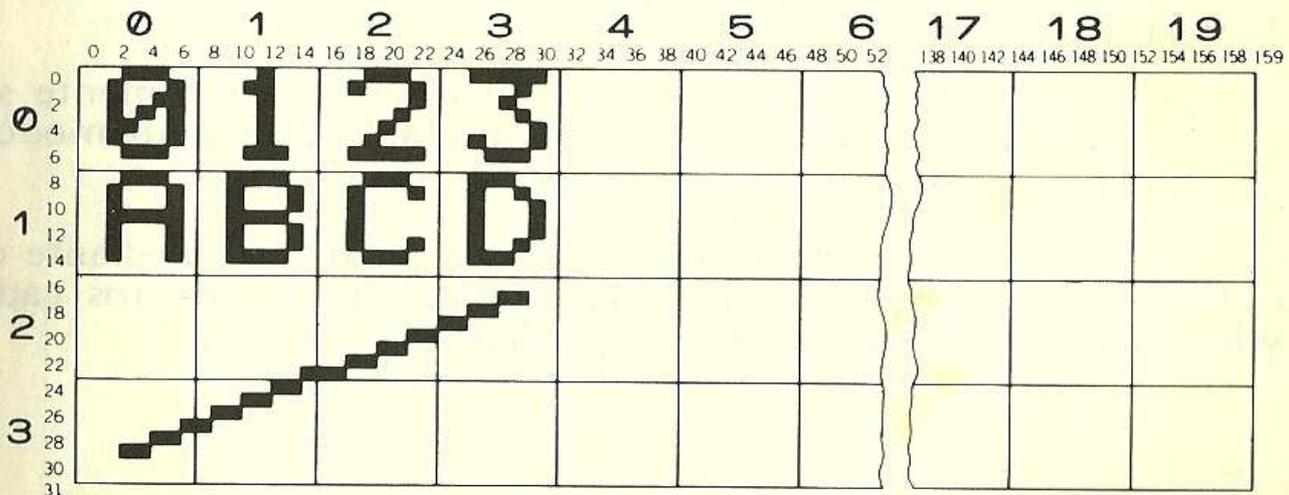
POINT Indica la condición del punto especificado

Antes de tratar el uso de los mandos y funciones gráficas, es necesario considerar las coordenadas de la pantalla. Las posiciones de los puntos se especifican por medio de las coordenadas horizontales 0 a 159 y las coordenadas verticales 0 a 31. Para ubicar los caracteres en la pantalla, se utilizan las coordenadas horizontales 0 a 19 y verticales 0 a 3.

Cada carácter se considera como un patrón gráfico de 8 x 8 puntos. Los puntos y líneas pueden trazarse en cualquier posición de la pantalla por medio de las coordenadas gráficas.

La línea recta de la posición (3, 29) a (29,17) es un simple ejemplo de las posibilidades que ofrecen las funciones gráficas.

Coordenadas de la pantalla



Las coordenadas gráficas consisten de 5.120 puntos, con 160 en la dirección X y 32 en la dirección Y. De tal modo, el extremo superior izquierdo de la pantalla sería (0, 0) y el interior derecho (159, 31). Todo los puntos que conforman la pantalla pueden especificarse por medio de estas coordenadas. Para ello, se utiliza el siguiente mando:

`DRAW (X, Y)`

Para borrar un punto en la posición (X, Y) use

`DRAWC (X, Y)`

Mediante el mismo mando se pueden trazar rectas, especificando ambos extremos de las mismas por medio de $(X_1, Y_1) - (X_2, Y_2)$.

`DRAW (X1, Y1) - (X2, Y2)`

Es posible, también, trazar rectas entre más de dos puntos. Por ejemplo:

`DRAW (X1, Y1) - (X2, Y2) - (X3, Y3)`

La especificación de coordenadas con guiones entre sí permite unir cuantos puntos se deseen por medio de rectas. Una línea recta puede borrarse mediante:

`DRAWC (X1, Y1) - (X2, Y2)`

La función POINT determina si el punto especificado está o no encendido, dando un 1 cuando lo está y un 0 cuando no. El formato es el siguiente:

`POINT (X, Y)`

Cuando se usa esta función dentro de un programa, generalmente se asigna el valor dado por POINT (X, Y) a una variable del siguiente modo:

`A = POINT (X, Y)`

Veamos a continuación un programa para trazar polígonos mediante el método de unión de puntos citado. Para ello, especificaremos cada vértice y lo uniremos mediante el mando DRAW.

■ Triángulo

Veamos a continuación un programa que traza un triángulo con sus vértices en las coordenadas (100, 5), (85, 25) y (125, 25).

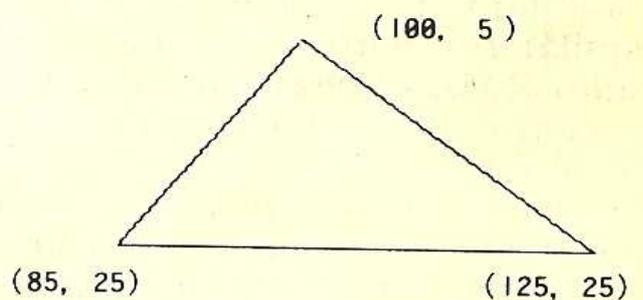
```

10 REM --- TRIANGLE ---
20 CLS
30 DRAW(100,5)-(85,25)-(125,25)-(100,
  5)
40 END
  
```

La especificación de las coordenadas para el mando DRAW puede realizarse también por medio de expresiones numéricas.

Así, el programa de arriba podría modificarse para obtenerse otro programa como el siguiente.

Triángulo trazado en la pantalla



```

10 REM --- TRIANGLE ---
20 CLS
30 X=100:Y=5 ..... Coordenadas donde comienza a trazarse
  el triángulo
40 DRAW(X,Y)-(X-15,Y+20)-(X+25,Y+20)-
  (X,Y)
50 END
  
```

■ Rectángulo

El programa a continuación traza un rectángulo con la línea recta (80, 5) - (150, 28) como se diagonal.

```

10 REM --- RECTANGLE ---
20 CLS
30 DRAW(80,5)-(150,5)-(150,28)-(80,28)
   )-(80,5)
40 END
    
```

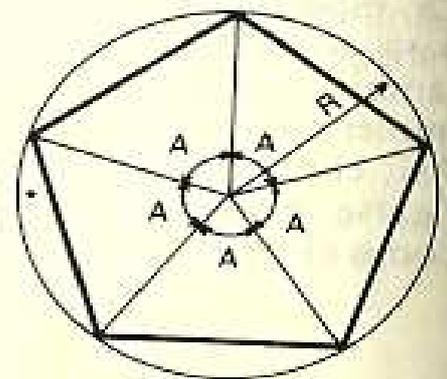
■ Pentágono equilátero

Todo polígono equilátero es aquel inscripto dentro de una circunferencia con sus vértices a distancias iguales.

La figura de la derecha ilustra un pentágono equilátero dentro de una circunferencia con radio R. Las líneas que unen los 5 vértices con el centro del círculo forman todas un mismo ángulo A, de 72° en este caso.

Por medio de las funciones ya citadas, se puede trazar un pentágono con su centro en (100, 18) y un radio igual a 15 del siguiente modo:

Pentágono equilátero



```

10 REM --- PENTAGON ---
20 CLS
30 R=15 ..... Radio
40 X=100:Y=18 ..... Centro del círculo
50 A=360/5 ..... Angulo
60 FOR I=0 TO 360 STEP A
70 DRAW(X-SIN(I)*R,Y-COS(I)*R)-(X-SIN
   (I+A)*R,Y-COS(I+A)*R)
80 NEXT I
90 END
    
```

■ **Polígono equilátero con N lados**

Cualquier polígono equilátero con N lados puede trazarse cambiando el ángulo A usado en el programa para el pentágono.

Pentágono $A = 360/5$
 Polígono con N lados $A = 360/N$

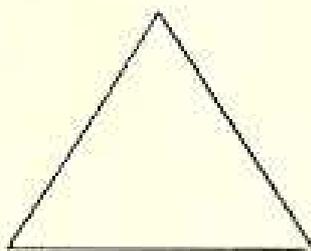
He a continuación un programa que permite trazar cualquier polígono equilátero con sólo entrar el valor de N (nº de lados):

El centro se ubicará en la posición (100, 15).

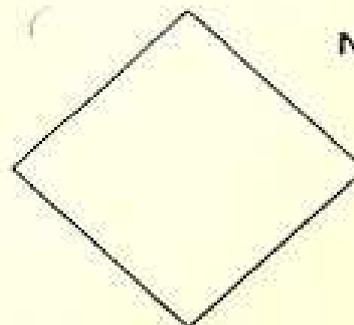
```

10 REM --- POLYGON ---
20 CLS
30 R=15 ..... Radio
40 X=100:Y=15 ..... Centro del círculo
50 INPUT "N=";N ..... Nº de lados
60 A=360/N ..... Angulo del poligono
70 FOR I=0 TO 360 STEP A
80 DRAW(X-SIN(I)*R,Y-COS(I)*R)-(X-SIN
(I+A)*R,Y-COS(I+A)*R)
90 NEXT I
100 END
    
```

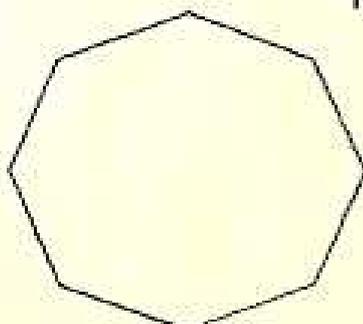
N = 3



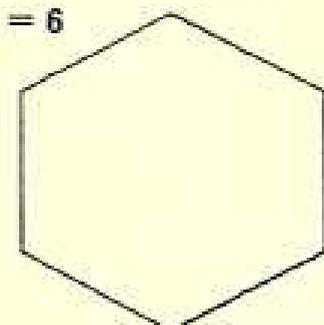
N = 4



N = 8



N = 6



3-23 TRAZADO DE CURVAS

Las curvas pueden trazarse especificando las coordenadas correspondientes en la pantalla. Sin embargo, surge el problema de qué coordenadas especificar.

El uso de ciertas fórmulas matemáticas facilita el trazado de numerosas curvas. Veamos a continuación el trazado de un círculo y una curva sinusoidal (SIN).

■ Círculos

A medida que se aumenta el valor de N en el programa para trazar polígonos equiláteros de la sección anterior, la figura resultante se acerca más a un círculo. Si bien, desde el punto de vista estrictamente geométrico, un polígono nunca puede ser un círculo, para trazar curvas en la pantalla lo que en realidad se hace es unir una serie de "líneas rectas".

El programa siguiente traza un círculo con un radio igual a 15 y con su centro (origen de las coordenadas) en la posición (100, 15).

```
10 REM --- CIRCLE ---
20 CLS
30 FOR I=0 TO 360 STEP 5
40 X=100+COS(I)*15 ..... Posición de los puntos ver-
50 Y=15-SIN(I)*15 ..... Posición de los puntos hori-
60 DRAW(X,Y) ..... zontales en el arco circular.
70 NEXT I
80 END
```

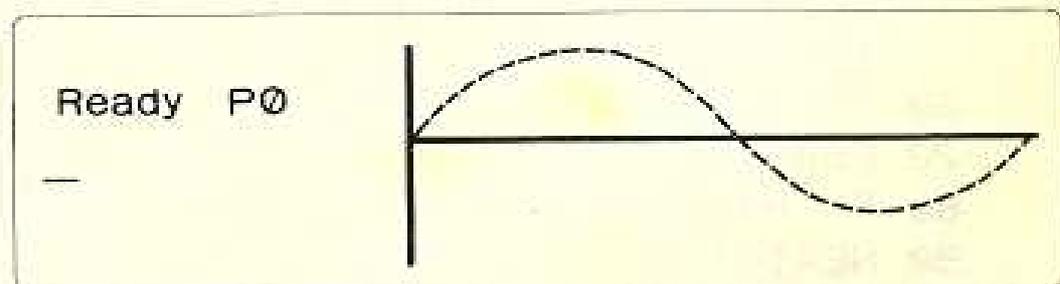
■ Curvas sinusoidales

El valor de la función SIN (I) cambia desde 0 hasta 1 y desde 0 hasta -1 a medida que el valor de I aumenta desde 0 hasta 360 grados. En base a ello, es posible trazar curvas sinusoidales mediante la unión de puntos (aumentados a las dimensiones adecuadas) a lo largo de un eje vertical. El programa a continuación traza una curva sinusoidal con los ejes X e Y.

```

10 REM --- SINE ---
20 CLS
30 A=65:B=15 ..... Origen de la curva (y de los ejes X e Y)
40 M=12 ..... Incremento
50 FOR I=0 TO 360 STEP 4
60 X=A+I/4 ..... Punto de la coordenada X
70 Y=B-SIN(I)*M ..... Punto de la coordenada Y
80 DRAW(X,Y)
90 NEXT I
100 DRAW(A,2)-(A,28) ..... Eje X
110 DRAW(A,B)-(A+92,B) ..... Eje Y
120 END

```



3-24 TRAZADO DE GRAFICOS LINEALES

Los gráficos lineales se utilizan con frecuencia para supervisar ciertos fenómenos con el pasar del tiempo (temperatura, precios, etc.). Para ello, se recomienda usar la pantalla en todo su potencial, para así poder visualizar los cambios con la mayor claridad posible.

■ Promedio de temperatura por mes

Supongamos que las temperaturas que aparecen en la tabla a continuación son los promedios para cada mes en una misma ciudad. La información se almacena dentro del programa por medio de la sentencia DATA. El gráfico se traza obteniendo la información de la sentencia DATA por medio de la sentencia READ.

| Mes | Temp. (°C) | Mes | Temp. (°C) | Mes | Temp. (°C) |
|------|------------|-------|------------|------|------------|
| En. | 11.5 | Mayo | 19.8 | Sep. | 22.3 |
| Feb. | 9.8 | Junio | 23.4 | Oct. | 18.5 |
| Mar. | 13.7 | Julio | 26.6 | Nov. | 15.9 |
| Abr. | 18.3 | Ag. | 28.2 | Dic. | 14.7 |

```
10 REM --- LINE GRAPH ---
20 CLS
30 FOR I=1 TO 3
40 PRINT 30-(I-1)*10;CHR$(147).....CHR$(147)
50 NEXT I
60 PRINT TAB(3);CHR$(154);.....CHR$(154)= L
70 FOR I=1 TO 12
80 PRINT CHR$(144);.....CHR$(144)=┘
90 NEXT I
100 FOR I=1 TO 12
110 X=36+(I-1)*8
120 READ A
130 Y=4+(30-A)*0.8
140 IF I=1 THEN 160
```

```

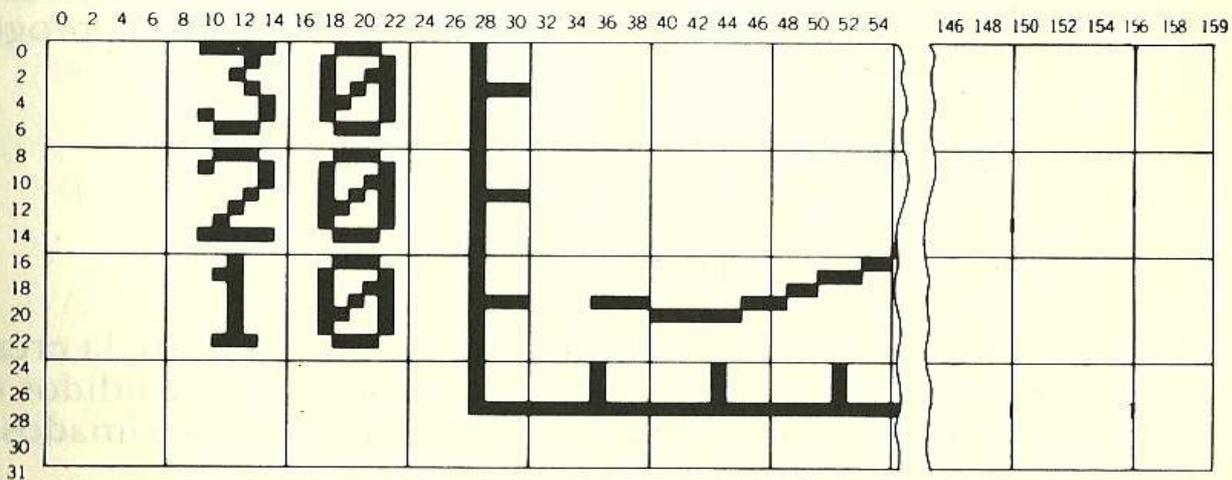
150 DRAW(P,Q)-(X,Y)
160 P=X:Q=Y
170 NEXT I
180 DATA11.5,9.8,13.7,18.3,19.8,23.4,2
      6.6,28.2,22.3,18.5,15.9,14.7
200 IF INKEY$="" THEN 200
210 END

```

Evita que lo visualizado
en pantalla avance una
línea hacia arriba

Una parte del patrón de puntos se amplía del modo ilustrado a continuación para ayudar a visualizar la relación posicional de los valores, del eje vertical Y, el eje horizontal X y de las líneas del gráfico. Siempre que prepare un programa para el trazado de gráficos, se recomienda incluir algún modelo que determine la relación posicional de los valores trazados.

Patrón de puntos



Si se omitiese la línea 200, la cual produce una condición de espera de entrada desde el teclado sin modificar lo visualizado en pantalla, haría que finalice el programa y aparezca el mensaje:

Ready P0

La línea 200 evita que finalice el programa hasta que no se presione alguna tecla y que el gráfico permanezca en pantalla intacto.

3-25 PREPARACIONES PARA EL TRAZADO DE GRAFICOS DE BARRAS

■ Trazado de un gráfico de barras usando caracteres

La ejecución del siguiente programa y la entrada de los números del 1 al 20 en N hará que se visualicen N números de caracteres "—" una a continuación del otro.

```
10 CLS
20 INPUT "N=";N
30 FOR I=1 TO N
40 PRINT CHR$(131);
50 NEXT I
60 END
```

CHR\$(131) es el código para el carácter "—".

La longitud de la barra es directamente proporcional al valor de N. Como esta unidad puede visualizar hasta 20 caracteres en una única línea, resulta necesario reducir el valor de N de cierta manera que sea posible también presentar en la pantalla aquellos valores mayores que 20. Por ejemplo, para visualizar un valor igual a 100, en el programa deben hacerse las siguientes modificaciones.

```
30 FOR I=1 TO N/6
      ó
30 FOR I=1 TO N STEP 6
```

Debe tenerse en cuenta, no obstante, que esta técnica permite la creación de barras de la misma longitud para todos los valores comprendidos entre 90 y 95 inclusive, por lo que sólo se lograrían gráficos aproximados. Los gráficos permiten solucionar este problema.

■ Use de gráficos en una gráfico de barras

En el programa que acabamos de ver se pueden incluir gráficos del siguiente modo. El empleo de gráficos permite disponer de un total de 160 unidades, lo cual resulta en una resolución B veces mayor que la lograda con el método anterior.

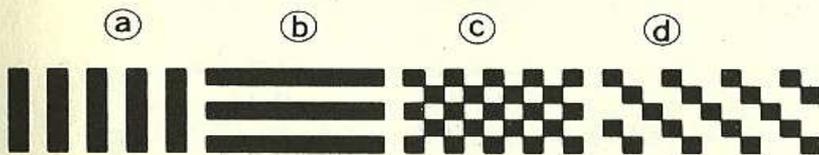
```

10 CLS
20 INPUT "N=";N
30 FOR I=1 TO N*1.5
40 DRAW(I-1,8)-(I-1,14)
50 NEXT I
60 END

```

Si bien en el programa anterior los gráficos adoptan un sólo tipo de patrón de puntos, es fácil cambiar el patrón para visualizar el gráfico en forma clara.

Para ello, se pueden agregar o substituir los ejemplos a continuación (a) hasta (d) en el programa anterior.



- (a) 35 IF I=N*1.5 THEN 40
36 IF I MOD 2=0 THEN 50
- (b) 30 FOR I=8 TO 14 STEP 2
40 DRAW(0,I)-(N*1.5-1,I)
- (c) 35 FOR J=8 TO 14 STEP 2
40 DRAW(I-1,J+(I+1) MOD 2)
45 NEXT J
- (d) 25 FOR J=8 TO 14
30 FOR I=1 TO N*1.5 STEP 3
35 X=(J-8) MOD 3+I-1
36 IF X>N*1.5-1 THEN X=N*1.5-1
40 DRAW(X,J)
50 NEXT I:NEXT J

3-26 DOS EJEMPLOS DE PROGRAMAS PARA GRAFICOS DE BARRAS

Los gráficos de barras se utilizan para representar la relación relativa entre, por ejemplo, puntajes de exámenes, ventas o cantidades de producción. Por su lado, a los gráficos se les debe dar una escala precisa, para que puedan representar lo más precisamente posible cada uno de los valores dentro de los límites de la pantalla.

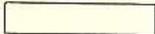
Básicamente, los gráficos de barras se clasifican en dos grupos. En el primero de ellos, se representa la relación entre un número de cantidades, asignando a cada una de ellas su barra correspondiente (por ej.: totales de producción o venta de diferentes productos). En el segundo de los casos, se representa la relación de cada cantidad en función a un total (por ej.: totales de producción o venta de diferentes productos en relación al total de las ventas de todos los productos).

Preparemos a continuación un programa para cada uno de los métodos mencionados.

Ejemplos: Cantidades de producción para los vehículos A y B.

| | 1980 | 1981 | 1982 |
|------------|--------|--------|--------|
| Vehículo A | 48,200 | 57,200 | 67,200 |
| Vehículo B | 39,200 | 31,100 | 27,500 |

 : Modelo de la barra para el vehículo A

 : Modelo de la barra para el vehículo B

■ Asignación de una barra a cada ítem.

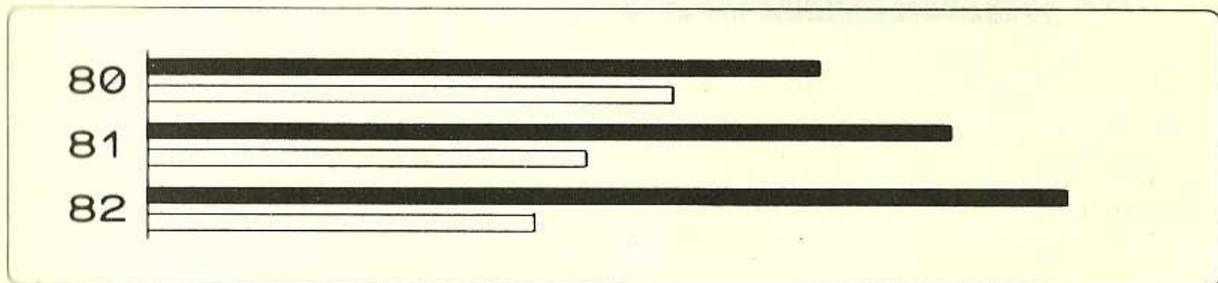
```
10 REM --- BAR GRAPH ---
20 CLS
30 FOR I=0 TO 2
40 LOCATE 0,I
50 PRINT 80+I;CHR$(136) .....CHR$(136)=I
60 FOR J=1 TO 2
70 READ A
80 FOR K=0 TO 2
```

```

90 Y=I*8+(J-1)*4+K
100 IF J=2 THEN IF K=1 THEN DRAW(24+A/
    500,Y):GOTO 120
110 DRAW(25,Y)-(24+A/500,Y)
120 NEXT K:NEXT J:NEXT I
130 IF INKEY$="" THEN 130
140 END
150 DATA48200,39200,57200,31100,67200,
    27500

```

Ejemplo de visualización



■ Comparacion de totales independientes con el gran total

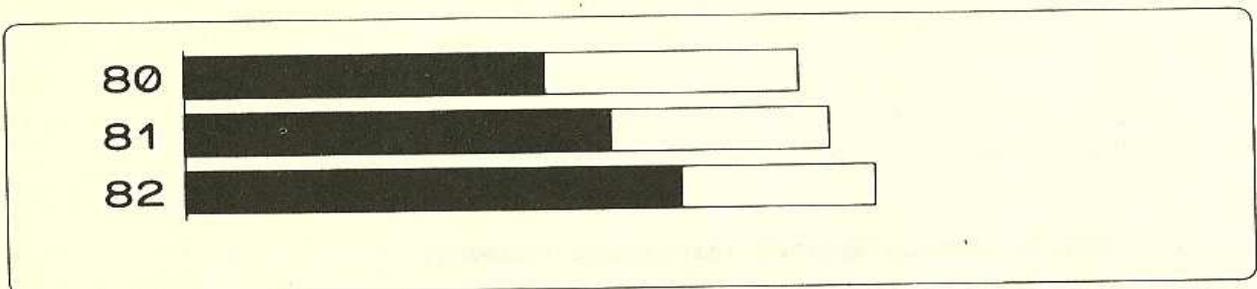
```

10 REM --- BAR GRAPH ---
20 CLS
30 FOR I=0 TO 2
40 LOCATE 0,I
50 PRINT 80+I;CHR$(136)
60 READ A,B
70 FOR J=0 TO 6
80 Y=I*8+J
90 DRAW(25,Y)-(24+A/1000,Y)
100 NEXT J
110 X=24+A/1000:Y=I*8
120 DRAW(X+1,Y)-(X+B/1000,Y)-(X+B/1000
    ,Y+6)-(X+1,Y+6)

```

```
130 NEXT I
140 IF INKEY$="" THEN 140
150 END
160 DATA48200, 39200, 57200, 31100, 67200,
    27500
```

Ejemplo de visualización



3-27 GRAFICOS EN MOVIMIENTO

La ejecución del programa a continuación hará que un asterisco * se mueva hacia la izquierda y derecha.

```
10 CLS
20 FOR I=5 TO 15
30 LOCATE I,1
40 PRINT " *"
50 NEXT I
60 FOR I=15 TO 5 STEP -1
70 LOCATE I,1
80 PRINT "* "
90 NEXT I
100 GOTO 20
```

Movimiento de izquierda a derecha

Movimiento de derecha a izquierda

Como podrá observarse, la variable de control (I) aumenta desde 5 hasta 15. Ello hace que las coordenadas del mando LOCATE que se encuentra en la línea 30 vayan cambiando desde (5,1) hasta (15,1). Téngase en cuenta que el mando PRINT de la línea 40 tiene un espacio antes de símbolo " * ".

Este espacio reemplaza el asterisco visualizado previamente, para que el segundo desaparezca de la pantalla. Este método, en el cual se visualiza el asterisco en la siguiente posición y se borra el de la posición anterior, produce una ilusión de movimiento (en este caso, de izquierda a derecha). En las líneas 60 hasta 90, el valor de la variable de control decrece en lugar de aumentar, lo que hace posible una inversión del proceso para que el asterisco se desplace esta vez en sentido inverso (de derecha a izquierda). Este es uno de los principios básicos de los programas de gráficos animados.

El movimiento vertical se logra en un modo similar; pe como el asterisco se va visualizando en líneas sucesivas, debe agregarse una setencia para borrar el asterisco que queda en la línea anterior. Veamos este cas en el programa siguiente.

```
10 CLS
20 FOR I=0 TO 3
30 LOCATE 10,I:PRINT "*";
40 IF I=0 THEN 60
50 LOCATE 10,I-1:PRINT " "
60 NEXT I
```

Movimiento de arriba hacia abajo

```

70 FOR I=3 TO 0 STEP -1
80 LOCATE 10,I:PRINT "*";
90 IF I=3 THEN 110
100 LOCATE 10,I+1:PRINT " "
110 NEXT I
120 GOTO 20

```

} Movimiento de abajo
hacia arriba

■ Ajuste de la velocidad

La velocidad con que se desplaza el asterisco a lo largo de la pantalla la controla la sentencia FOR-NEXT.

Ejecute el programa de movimiento horizontal agregando antes la línea a continuación.

```
45 FOR J=1 TO 50: NEXT J
```

Al agregarse esta sentencia, la velocidad de desplazamiento de izquierda a derecha baja levemente. La misma puede aumentarse con sólo reducir el valor final de la sentencia FOR-NEXT; para aumentar la velocidad, utilice un valor más pequeño.

■ Movimiento de un punto en forma curvilínea

El movimiento de un punto para trazar una línea curva se logra mediante el uso de las coordenadas gráficas. En el siguiente programa, un punto se desplaza formando una línea punteada dentro de un cuadro (ver el ejemplo de ejecución).

```

10 CLS
20 DRAW(13,0)-(13,31)-(137,31)-(137,0)
..... Traza un cuadro

```

```

30 DRAW(14,0)-(14,30)-(136,30)-(136,0)
..... Traza un cuadro

```

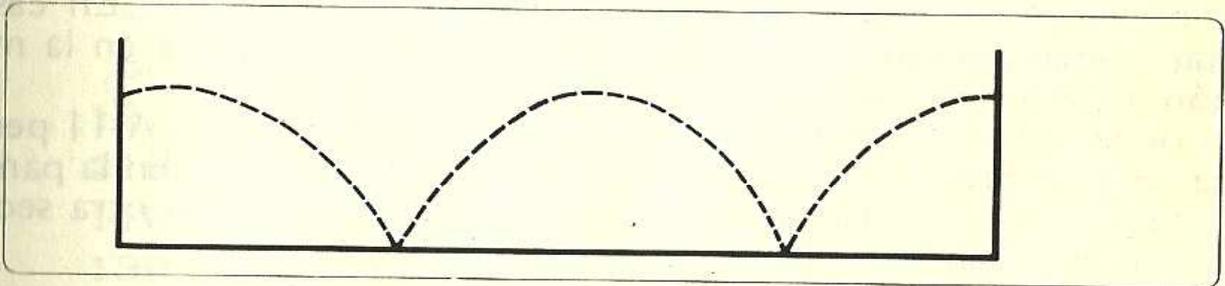
```

40 N=1:P=14
50 FOR I=0 TO 360 STEP 3
60 X=P+N:Y=29-25*ABS(COS(I))
70 DRAW(X,Y)
80 IF P=14 THEN 100
90 DRAWC(P,Q)
100 P=X:Q=Y
110 NEXT I
120 N=-N
130 GOTO 50

```

Cómputo de las coordenadas de la línea curva.

Ejemplo de ejecución



■ Función POINT

La función POINT verifica una posición específica de la pantalla para ver si está encendida o no.

Cuando la posición especificada por las coordenadas (X, Y) está encendida (hay un punto).

POINT (X, Y) ... 1

En caso contrario

POINT (X, Y) ... 0

El siguiente programa, da un ejemplo de la función POINT. Primero se visualiza el mensaje "CASIO PB-770" en la primera línea y luego, la función POINT verifica la condición (0 ó 1) de (X, Y) y copia la misma línea en la tercera línea.

```
10 CLS
20 PRINT "*** CASIO PB-770 ***"
30 FOR X=0 TO 159
40 FOR Y=0 TO 7
50 IF POINT(X,Y)=0 THEN 70
60 DRAW(X,Y+16)
70 NEXT Y
80 NEXT X
90 IF INKEY$="" THEN 90
100 END
```

Las líneas 30 hasta 80 verifican el estado de los puntos en la primera línea de coordenadas (si están encendidos o apagados). En case de no haber ninguno encendido, se salta de la línea 50 a la 70. En case de detectarse algún punto encendido, el mismo se reproduce en la misma posición de la tercera línea (de caracteres).

El uso de la impresora por coordenadas opcional FA-10 ó FA-11 permite obtener una impresión idéntica de lo que está visualizado en la pantalla. Los detalles sobre el use de las impresoras se incluyen en otra sección; el siguiente programa se incluye sólo como referencia.

```
10 CLS
20 PRINT "CASIO PB-770"
25 LPRINT CHR$(28);CHR$(37)
30 FOR X=0 TO 159
40 FOR Y=0 TO 7
50 IF POINT(X,Y)=0 THEN 70
55 U=X*0.59:W=Y*0.59
60 LPRINT "D";U;",";"-1*W;",";U+0.4;",";
    "-1*(W+0.4)
70 NEXT Y
80 NEXT X
90 IF INKEY$="" THEN 90
100 END
```

3-28 JUEGOS

Las técnicas de gráficos animados citadas en secciones previas se utilizan con frecuencia en juegos, como en el programa a continuación. Si bien no se requiere el uso de métodos muy complicados, los juegos hacen uso de rutinas que producen gráficos animados en una forma muy real.

La longitud del programa puede que sea un poco desalentadora; no obstante, se sugiere al programador que haga algunas modificaciones para, de tal modo, familiarizarse con las funciones incluidas en cada línea. Esta es una de las mejores maneras de aprender técnicas de programación. Antes que nada, entre el siguiente programa.

```
10 REM ---BLOCK DESTROY---
20 CLS
30 FOR I=0 TO 3
40 LOCATE 2,I:PRINT CHR$(137);
50 NEXT I
100 FOR I=3 TO 6
110 FOR J=0 TO 3
120 LOCATE I,J
130 PRINT CHR$(141);
140 NEXT J:NEXT I
200 LOCATE 17,1
210 PRINT CHR$(147)
220 R=1:S=0
300 FOR N=3 TO 1 STEP -1
310 LOCATE 0,0:PRINT N
320 X=10:Y=INT(RND*2+1):A=1:B=1
330 LOCATE X,Y:PRINT CHR$(236);
340 Q$=INKEY$
350 IF Q$="1" THEN GOSUB 500 ELSE IF Q
    $="0" THEN GOSUB 600
360 IF X=3 THEN A=-A
370 IF Y=0 THEN B=-B ELSE IF Y=3 THEN
    B=-B
```

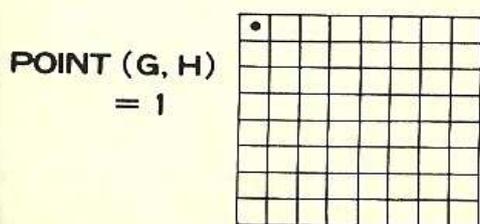
```
380 IF X=16 THEN GOSUB 800
390 LOCATE X,Y:PRINT " ";
400 IF X>16 THEN 440 391 Print CHR$(32);
410 G=(X+A)*8:H=(Y+B)*8
420 P=POINT(G,H)
430 IF P=1 THEN GOSUB 700
440 X=X+A:Y=Y+B
450 IF X>18 THEN BEEP :BEEP :BEEP ELSE
    E 330
460 NEXT N
470 CLS :LOCATE 2,1:PRINT "SCORE ";S Anotacion
480 FOR I=1 TO 6:BEEP 1:NEXT I
490 IF INKEY$="" THEN 490 ELSE END
500 LOCATE 17,R:PRINT " ";
510 R=R-1
520 IF R<0 THEN R=0 (41)
530 LOCATE 17,R:PRINT CHR$(147);
540 RETURN
600 LOCATE 17,R:PRINT " ";
610 R=R+1:IF R>3 THEN R=3 (41)
620 LOCATE 17,R:PRINT CHR$(147);
630 RETURN
700 S=S+8-X:BEEP 1
710 LOCATE X+A,Y+B:PRINT " ";
720 X=X+A:Y=Y+B:A=-A
730 IF Y=0 THEN B=-B ELSE IF Y=3 THEN
    B=-B
740 RETURN
800 IF Y=R THEN A=-A:BEEP ELSE IF Y+B
```

```

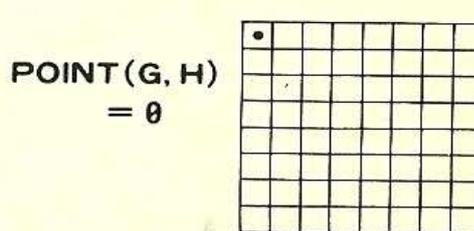
      =R/ THEN BEEP :A=-A ELSE 830
810 IF RND<0.7 THEN 830
820 LOCATE X,Y:PRINT " ";:X=X-1
830 RETURN

```

La línea 420 usa la función POINT para verificar la condición en determinada posición dentro de un bloque. Para ello, sólo verifica la coordenada izquierda superior del modo ilustrado a continuación.



Un bloque visualizado mediante CHR\$ (141).



Un bloque borrado mediante CHR\$ (32).

■ El juego en sí

Al ejecutarse el programa, a la izquierda se trazan bloques y a la derecha aparece una "raqueta". La bola se mueve a lo largo de la pantalla, y el jugador debe usar la raqueta para mantener la pelota en el área de juego. La raqueta se eleva mediante la pulsación de y se baja mediante la pulsación de . Si el jugador le erra a la bola con la raqueta, se registra un fallo y se saca la bola siguiente. El juego termina al registrarse el tercer fallo. Al final del juego, aparece el puntaje logrado.

- A, B Dirección de la bola
- G, H Coordenadas para verificar si hay o no un bloque
- I, J Variables para el trazado de bloque y los límites
- N Número de bolas que restan
- P Verifica si el bloque está encendido o no
- Q\$ Dirección de la raqueta
- R Posición de la raqueta
- S Puntaje
- X, Y Posición de la bola

3-29 TRAZADO DE GRAFICOS EN LA IMPRESORA POR COORDENADAS

La conexión de la impresora por coordenadas con interface para brabadora de cassette FA-10 ó FA-11 opcional permite imprimir lo que está visualizado en la pantalla. Si bien las impresoras por coordenadas permiten la impresión de caracteres comunes, su finalidad primordial es el trazado de gráficos. Sea cual fuere el caso, la impresión se realiza imprimiendo un número de puntos en series, hasta formar la figura final.

Este método es similarr al utilizado para trazar gráficos en la pantalla, sólo que la impresora requiere ciertos mandos especiales para aumentar la velocidad de sus operaciones mecánicas.

El use de la gran cantidad de mandos para las impresoras por coodenadas aceleran su funcionamiento y simplifican la programación.

■ Mandos

Todos los mandos para trazar gráficos en la impresora por coordenadas comienzan con LPRINT. La combinación de LPRINT con los mandos en la tabla de la página siguiente permitirá la ejecución de gran cantidad de funciones.

No obstante, antes de usar estos mandos es necesario especificar el modo gráfico mediante la siguiente sentencia.

Especificación del modo gráfico: LPRINT CHR\$(28); CHR\$(37)

La sentencia a continuación cancela el modo gráfico (por ej.: para especificar el modo de caracteres).

Especificación del modo de caracteres: LPRINT CHR\$(28); CHR\$(46)

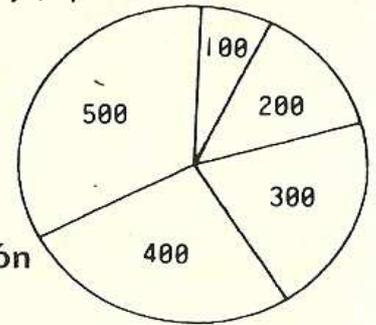
Tabla de mandos para la impresora por coordenadas

| | Mando | Nombre | Descripción |
|--------------------------|-------|------------------|--|
| Trazado gráficos | O | ORIGIN | Especificación del origen de las coordenadas ORG. |
| | D | DRAW | Une los puntos especificados por las coordenadas ORG. |
| | I | RELATIVE DRAW | Traza una línea hasta el punto especificado. |
| | M | MOVE | Se desplaza hasta la posición especificada por las coordenadas ORG sin trazar línea alguna. |
| | R | RELATIVE MOVE | Se desplaza hasta la posición especificada sin trazar línea alguna. |
| | A | QUAD | Traza un paralelogramo en los ejes X e Y con la diagonal entre dos puntos indicados por las coordenadas ORG. |
| | C | CIRCLE | Traza un círculo y un arco circular con el centro en el punto especificado por las coordenadas ORG. |
| | X | AXIS | Traza un eje de coordenadas desde el origen de las coordenadas ORG en las direcciones +Y, +X, -Y y -X. |
| | G | GRID | Traza líneas horizontales dentro de un cuadrado o rectángulo especificado. |
| | L | LINE TYPE | Traza una línea llena, de guiones, punteada o de dos puntos encadenados. |
| | B | LINE SCALE | Especifica el paso de las líneas punteadas, de guiones y de dos puntos encadenados. |
| Caracteres y símbolos | S | ALPHA SCALE | Especifica el tamaño de los caracteres y símbolos. |
| | Q | ALPHA ROTATE | Especifica el sentido rotativo de los caracteres y símbolos. |

| | Mando | Nombre | Descripción |
|-----------------------|-----------|-----------------------------|--|
| Caracteres y símbolos | Z | SPACE | Especifica el espacio entre caracteres y líneas. |
| | Y | VERTICAL ó HORIZONTAL | Especifica el sentido de impresión (vertical u horizontal) |
| | P | PRINT | Imprime una secuencia de caracteres. |
| | N | MARK | Traza una marca con la posición del bolígrafo como su centro. |
| Control | J | NEW PEN | Selección del color. |
| | F | LINE FEED | Avance del papel y retorno del carro por línea. |
| | H | HOME | Modifica las coordenadas absolutas o desplaza el patrón a una posición fácil de ver. |
| | CHR\$(64) | TEST | Verifica la condición del bolígrafo y su tinta. Ejemplo de ejecución: LPRINT CHR\$(28); CHR\$(37); CHR\$(64)  |
| Control de caracteres | T | TAB | Tabulador |
| | ? | FORMAT | Impresión de listas de programas. |


```

80 LPRINT "D48, -50, "; X; ", "; Y
90 A2=(A1+A)/2:GOSUB 200:NEXT I
95 END
100 DATA100, 200, 300, 400, 500
200 X=50+INT(20*SIN(360*S/(T+A2))):Y=-
    48+INT(20*COS(360*S/(T+A2)))
210 LPRINT "M"; X-10; ", "; Y
220 B$=STR$(A)
230 LPRINT "P"; B$
240 RETURN
    
```



Ejemplo de ejecución

La línea 40 incluye un mando READ para leer lo datos que se encuentran en la línea 100. Las líneas 60 y 70 calculan las posiciones X e Y dentro de la circunferencia del círculo para cada uno de los datos. La línea 80 une por medio de una línea recta la posición (X, Y) con el centro del círculo.

3-31 USO DE LOS PROGRAMAS PARA LA PB-770

Si bien es cierto que la PB-770 tiene muchas más funciones que la PB-770, los programas escritos para la segunda pueden usarse en la primera sin modificación alguna.

Ello no significa que los programas escritos para la PB-770 puedan, necesariamente, usarse en la PB-770 sin ninguna modificación. Para ello, puede ser que resulte necesario cambiar el programa, ya que algunos mandos no pueden usarse en la PB-770.

Los programas o datos grabados en cintas de cassette por medio de la PB-770 pueden usarse sin modificaciones en la PB-770, pero no la inversa (la PB-770 no puede recuperar la información grabada con la PB-770).

La diferencia entre

La diferencia entre estos dos modelos son:

- Mandos adicionales

POKE

- Funciones adicionales

DEG, HYP SIN, HYPCOS, HYPTAN, HYPASN, NYPACS, HYPATN, PEEK, DMS\$, HEX\$ y &H

- Mandos modificados

PRINT y CLEAR

- Función modificada

CHR\$

PRECAUCION:

Siempre que se utilicen programas escritos para la PB-770, es necesario cambiar los números de líneas que se encuentren en líneas entre los números 2200 y 2299, y en aquellas líneas cuyos dos últimos dígitos sean dos 2 (por ej.: 22, 122, 622). Ver la página 208.

Prefacio al Capítulo 4

Los siguientes términos se utilizan en la sección "Formato" de cada mando o función descriptos en el Capítulo 4.

- **"Expresión numérica"**
Valores numéricos, variables y expresiones de cómputo. Los cálculos numéricos se ejecutan en base a la prioridad de cálculo según los operadores.
- **"Expresión de caracteres"**
Constantes, variables y secuencias de caracteres.
Las secuencias de caracteres pueden encadenarse usando el signo "+". Así, la "suma" de secuencias de caracteres sería, entonces, una expresión de caracteres.
(Ejemplo) "ABC" + "DE" + "F" = "ABCDEF"
Tenga en cuenta que se pueden encadenar un máximo de 79 caracteres.
- **"Variable"**
Se utilizan para almacenar información. Hay dos tipos de variables, las numéricas (como ser A o B) y las de caracteres (como ser A\$ o B\$). Para mayores detalles, remítase al Capítulo 3.
- **"Nombre de variable"**
Una variable puede ser una letra mayúscula (A hasta Z) sola o seguida de \$, números o letras minúsculas.
(Ejemplo) A, A\$, AB, A1, XY\$, X1\$. . . etc.
Se refiere a cómo se denomina la variable.
- **"Expresión condicional"**
Toda expresión que compara dos valores por medio de un operador comparativo (=, = <, >, <>, etc.).
- **"Número de línea"**
El número al comienzo de cada línea de un programa. Los límites son de 1 hasta 9999.
- **"Comentario"**
Son partes del programa que no afectan su ejecución y tiene por finalidad facilitar el proceso de depuración del mismo mediante la explicación de cada parte del programa.
- **"Mensaje"**
Los mensajes se visualizan para informar a quien ejecuta el programa el tipo de información a entrar desde el teclado. Deben incluirse entre comillas.
- **"Nombre de archivo"**
Los nombres de archivo identifican los programas o datos que se transfieren entre cintas de cassette y la computadora.

CAPITULO 4

GUIA DE MANDOS

4-1 MANDOS MANUALES

AUTO

| | |
|----------------|---|
| Función | Genera automáticamente números de líneas de programa |
| Formato | AUTO (primer número de línea), (, incremento) ($1 \leq \text{primer número de línea} \leq 9999$) ($1 \leq \text{incremento} \leq 9999$) |

El mando AUTO simplifica el proceso de programación mediante la generación de números de línea secuenciales con un incremento preestablecido.

Por cada pulsación de la tecla , se generan continuamente números de línea a partir del número especificado por el primer argumento y con el incremento especificado por el segundo argumento. Ante la falta de ambos argumentos, los mismos toman un valor igual a 10. Este mando no puede ejecutarse si se ha especificado una clave.

La función de este mando puede cancelarse mediante los siguientes procedimientos:

- (1) Pulsando la tecla  inmediatamente después de haber aparecido en la pantalla el número de línea.
- (2) Presionando **BRK** o **CLS**.
- (3) Siempre que la función intente la generación de un número de línea mayor que 9999.
- (4) Cuando el número de línea generado automáticamente es igual a uno ya existente (símbolo  después del número de línea).

Uso

AUTO 100

Esta entrada hace que se generen números de línea a partir del número 100, con un incremento de 10, o sea, 100, 110, 120 y así sucesivamente.

AUTO 50, 50

Esta entrada hace que se generen números de línea a partir del número 50, con un incremento de 50, o sea, 50, 100, 150 y así sucesivamente.

CONT

| | |
|----------------|---|
| Función | Reanuda la ejecución de un programa interrumpido por la sentencia STOP o la entrada  . |
| Formato | CONT |

La reanudación de todo programa cuya ejecución haya sido interrumpida por la acción del mando STOP o mediante la pulsación de la tecla , se lleva a cabo mediante el mando CONT. En dicho caso, la ejecución se reanuda a partir de la sentencia que sigue al mando STOP que la interrumpió.

Este mando es muy útil, ya que permite la verificación y modificación del contenido de las variables durante la interrupción de la ejecución.

Uso

Inserte el mando STOP entre las diferentes secciones de un programa para facilitar su proceso de depuración. El programa a continuación es un ejemplo:

```

10 READ R,H
20 S=PI*R^2
30 STOP
40 U=S*H
50 PRINT R,H,S,U
60 DATA 10,20
70 END

```

Al entrarse el mando RUN, la ejecución del programa se interrumpe en la línea 30 por la acción del mando STOP.

Verifique la ejecución de las líneas 10 y 20, y revise manualmente los valores de las variables. Para ello, entre R , H  y S . Si se visualizara 10, 20 y 314.1592654, respectivamente, significa que la ejecución ha sido correcta.

Una vez finalizada la verificación, reanude la ejecución del programa entrando el mando CONT.

CONT 

 STOP

DELETE

| | |
|-----------------|--|
| Función | Borrado de una o más líneas de un programa. |
| Formatos | DELETE <i>ln</i> DELETE <i>ln</i> - DELETE - <i>lm</i> DELETE <i>ln</i> - <i>lm</i> <i>ln</i> : Primera línea que ha de borrarse <i>lm</i> : Última línea que ha de borrarse $(1 \leq ln \leq lm \leq 9999)$ |

Este mando sirve para borrar una o más líneas (especificadas) de un programa.

El mando DELETE puede tener los siguientes formatos según su aplicación:

- (1) DELETE número de línea . . . Borrado de una línea especificada
- (2) DELETE número de línea - . Borrado de la línea especificada hasta la última del programa
- (3) DELETE - número de línea . Borrado de la línea inicial hasta la especificada
- (4) DELETE número de línea *n*
- número de línea *m* Borrado de las líneas comprendidas entre *ln* y *lm*.

ln y *lm* tienen los siguientes límites:

$$1 \leq ln \leq lm \leq 9999$$

Este mando no puede utilizarse dentro de un programa.

Uso

El siguiente programa da una explicación detallada de este mando.

```

10 REM DELETE SAMPLE
20 PRINT "20:A"
30 PRINT "30:B"
40 PRINT "40:C"
50 PRINT "50:D"

```

[1] DELETE número de línea 

Para borrar, por ejemplo, la línea 30 de este programa, se entra:

DELETE 30 

El mismo efecto puede lograrse mediante:

30 

[2] DELETE número de línea –  (use la tecla  para entrar el guión)

Para borrar de la línea 30 hasta la última del programa de arriba, entre:

DELETE 30  

Lo mismo puede lograrse mediante la entrada:

DELETE 25  

Ello se debe a que no existe la línea 25 y la computadora interpreta la línea siguiente más cercana, en este caso, la número 30.

[3] DELETE – número de línea 

Este mando borra desde la línea inicial del programa hasta la especificada. Entre lo siguiente como ejemplo:

DELETE  40 

Visualice la lista del programa por medio de    y verifique que se hayan borrado las líneas 10 hasta 40. Exactamente el mismo resultado puede lograrse entrando lo siguiente:

DELETE  47 

Nuevamente, a pesar de que la línea 47 no existe, se borran todas las líneas a partir de la primera del programa hasta la menor más cercana que 47.

[4] DELETE número de línea n – número de línea m 

Para borrar las líneas 20 hasta 40 del programa anterior, se entró:

DELETE 20  40 

- El error PR aparece siempre que se intenta el uso del mando DELETE con un programa protegido por la acción del mando PASS.
- La inclusión de decimales en el número de línea provocará la aparición de un error SN.
- Nada se borra cuando en el punto [4] de arriba n es mayor que m , o si el número de línea especificado no existe.

 NEW, NEW ALL, PASS

EDIT

| | |
|-----------------|--|
| Función | Permite la corrección de un programa. |
| Formatos | EDIT EDIT <i>ln</i> ($1 \leq ln \leq 9999$) |

El mando EDIT se utiliza para poner la computadora en el modo de compaginación a fin de corregir programas.

La primera línea del programa se visualiza por medio de EDIT  (el cursor aparece al final de la línea). Desplace el cursor por medio de las teclas , ,   y   y corrija por medio de  ó  .

■ **Uso de las teclas para corrección**

- (1) ,  Desplaza el cursor hacia la derecha e izquierda. El cursor se desplaza en la dirección respectiva hasta el final o comienzo de la línea manteniendo presionada una de estas teclas.
- (2)  
  Permite desplazar el cursor hacia arriba y hacia abajo.
- (3)  Borra caracteres continuamente mientras se la mantiene presionada.
- (4)   Entra espacios continuamente mientras se la mantiene presionada.

Uso

Entre el siguiente programa corregirlo:

```

10 PRINT "AVERAGE"
20 INPUT A,B,C
30 H=(A+B+C)/3
40 PRINT H
50 GOTO 10
    
```

(1) Verificación del listado de un programa desde el comienzo

Para visualizar el listado del programa y corregirlo línea por línea, entre lo siguiente:

EDIT 

Por cada pulsación de la tecla  , aparecerá en la pantalla la siguiente línea.

(2) Verificación del listado del programa de la última línea hacia la primera (a la inversa)

Para visualizar, por ejemplo, la línea 10 después de haber visualizado el listado hasta la línea 50, lleve a cabo el siguiente procedimiento:

Listado del programa hasta la línea 50.

```
20 INPUT A, B, C
30 H = (A+B+C) / 3
40 PRINT H
50 GOTO 10_
```



Entre cuatro veces  hasta que aparezca la línea número 10.

```
40 PRINT H
30 H = (A+B+C) / 3
20 INPUT A, B, C
10 PRINT "AVERAGE" _
```

(3) Verificación de una línea en medio del programa

Para verificar una línea que se encuentra en medio del programa, entre lo siguiente:

EDIT número de línea 

Luego, por cada pulsación de la tecla  , aparecen en pantalla las líneas a continuación.

(4) Uso de las teclas  ,  y 

Las líneas 20 y 30 del programa se corrigen del siguiente modo:

```
20 INPUT A, B, C, D
30 H = (A + B + C + D) / 4
```

Cuando usa EDIT  para visualizar la línea 20, aparece lo siguiente en la pantalla:

  20 

```
Ready P0
EDIT 20
      20 INPUT A, B, C_
```

En la línea 20, después de entrar “ , ” y “D”, la corrección se completa pulsando la tecla  , lo que hace aparecer en pantalla la línea 30.

```
Ready P0
EDIT 20
      20 INPUT A, B, C, D
      30 H= (A+B+C)/3_
```

Para insertar “+D” en la línea 30, intercale dos espacios (para dos caracteres).

Pulsadas al mismo tiempo

y presione      

- Siempre que se use el mando EDIT con un programa protegido por el mando PASS (clave), aparece un error PR.
- Liberación del modo para corrección
 - (1) Pulsando la tecla .
 - (2) Pulsando la tecla .
 - (3) Cuando se comete un error en la entrada.
 - (4) Cuando no hay programa alguno almacenado.
 - (5) Cuando se apaga la computadora.

procedimiento:

PROG 5  LIST 30 

- 5 LIST (LLIST) número de línea —  (para entrar “—”, use la tecla )

Cuando se entra un guión a continuación del número de línea, se visualiza (imprime) el programa almacenado en el área de programa en curso desde el número de línea especificado hasta la última línea. Mediante el siguiente procedimiento, aparecerá en la pantalla la lista del programa que se encuentra en el área de programa en curso, desde la línea 70 hasta la última.

LIST 70  

- 6 LIST (LLIST) — número de línea 

Cuando se entra un guión previo al número de línea, se visualiza (imprime) el programa almacenado en el área de programa en curso, desde la primera línea hasta la línea correspondiente al número especificado. Mediante el siguiente procedimiento, aparecerá en la pantalla la lista del programa que se encuentra en el área de programa en curso, desde la primera línea hasta le número 100.

LIST  100 

- 7 LIST (LLIST) N- de línea n — N- de línea m

En este caso, se visualizan (imprimen) las líneas del programa contenido en el área de programa en curso, cuyos números estén comprendidos entre n y m . n debe ser menor que m , y a que las líneas que componen el programa aparecen en el pantalla por orden ascendente. El siguiente procedimiento hace que se visualicen las líneas 50 hasta 70 del programa que se encuentra en el área de programa en curso.

LIST 50  70 

Técnicas

1 Interrupción momentánea de LIST y LIST ALL

El mando LIST visualiza las líneas del programa sin detenerse. Ello hace difícil revisar el contenido del programa por medio de este mando. No obstante, la ejecución de LIST pueden interrumpirse momentáneamente presionando la tecla . Para reanudar su ejecución, vuelva a pulsar la misma tecla.

La secuencia **SHIFT** **STOP** **ANS** no detiene la ejecución de LIST.

Para interrumpir definitivamente el mando LIST, presione la tecla **BRK**, la cual cancela también la ejecución del mando LIST ALL.

(2) Interrupción de LLIST y LLIST ALL

La impresión del listado de uno o más programas por medio de los mandos LLIST y LLIST ALL, respectivamente, no puede interrumpirse momentáneamente mediante la tecla **↵**. Sólo la tecla **BRK** es capaz de cancelar la ejecución de los mandos citados.

LOAD

| | |
|-----------------|---|
| Función | Traslada un programa o información que está almacenado en una cinta de cassette a la memoria interna de la computadora. |
| Formatos | LOAD LOAD ALL LOAD, A LOAD, M LOAD, D, dirección |

El mando LOAD se usa para extraer un programa que está almacenado en una cinta de cassette y trasladarlo a la memoria de la unidad PB-770 (el almacenamiento se debió haber realizado mediante la orden de SAVE). El mismo nombre de archivo que fue usado para el almacenamiento con SAVE se deberá usar. Por tanto, introduzca con el mando LOAD el nombre del archivo; después, en forma automática la PB-700 buscará ese archivo y lo extraerá de la cinta.

En caso de que no se introdujera un nombre específico de archivo, se extraerá el primer archivo que la máquina localice.

Explicaciones

El mando LOAD tiene los siguientes formatos.

(1) LOAD

Extrae el primer programa que encuentra entre los que fueron almacenados con SAVE o SAVE "nombre de archivo", en el área específica de programa.

No habrá problema aun si el área de programa que se usó durante el almacenamiento, SAVE, y el área de programa especificada en esos momentos son diferentes.

(2) LOAD "nombre de archivo"

Con este mando se extrae el programa que fue almacenado con ese mismo nombre en el área de programa especificado de esa ocasión. En este caso, el área de programa usada durante el mando SAVE y el área de programa usada durante LOAD pueden ser diferentes.

(3) LOAD ALL

Este mando extrae los programas que fueron almacenados con el mando SAVE ALL para colocarlos en las mismas áreas de programa.

Como en este caso no hay nombre de programa, los programas que sean encontrados primero entre aquellos que fueron almacenados con el mando SAVE ALL serán extraídos y colocados en la memoria de la máquina.

(4) LOAD ALL "nombre de archivo"

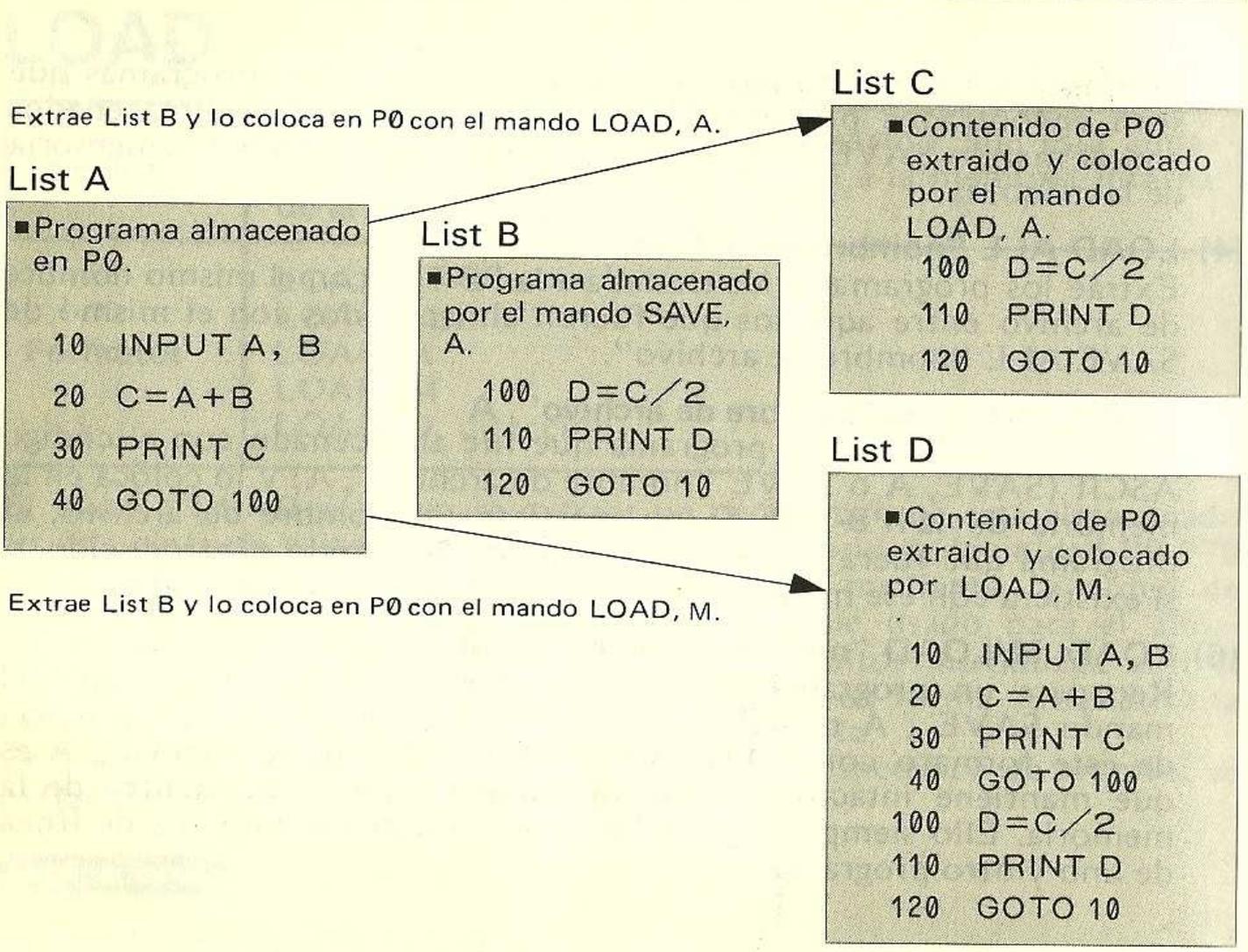
Extrae los programas y los traslada a la PB-770 con el mismo nombre de archivo entre aquellos que fueron almacenados con el mismo de SAVE ALL "nombre de archivo".

(5) LOAD, A/LOAD "nombre de archivo", A

Este mando extrae un programa que fue almacenado con el código ASCII (SAVE, A o SAVE "nombre de archivo", A) y lo coloca en la memoria de la PB-770. Si no existiera tal nombre de archivo, el programa que fuera encontrado primero es extraído e introducido, y si existiera con ese nombre entonces es extraído e introducido.

(6) LOAD, M/LOAD "nombre de archivo", M

Recupera un programa previamente grabado en cinta mediante el mando SAVE, A o SAVE "nombre de archivo", A. La diferencia de este formato con LOAD A y LOAD "nombre de archivo", A es que mantiene intacto el programa que se encuentra dentro de la memoria. Ello siempre y cuando no se repitan los números de línea de uno y otro programa.



(7) LOAD, D, dirección

Recupera de cinta datos grabados en código de máquina (SAVE, D, dirección 1, dirección 2 ó SAVE "nombre de archivo", D, dirección 1, dirección 2). Las direcciones deben ser mayores que -32769 y menores que 65536. Para mayores detalles sobre las direcciones de memoria, remítase a la sección correspondiente a CLEAR.

Uso

1 Los formatos de SAVE y LOAD deben coincidir

El formato del mando LOAD debe coincidir con el del mando SAVE utilizado. De modo similar, todo programa grabado por medio de SAVE ALL debe recuperarse por medio de LOAD ALL. En la tabla a continuación se muestra la relación entre los mandos LOAD y SAVE.

| | LOAD | LOAD "nombre de archivo" | LOAD ALL | LOAD ALL "nombre de archivo" | LOAD, A | LOAD "nombre de archivo", A | LOAD, M | LOAD "nombre de archivo", M | LOAD, D dirección | LOAD "nombre de archivo", D, dirección |
|---|------|--------------------------|----------|------------------------------|---------|-----------------------------|---------|-----------------------------|-------------------|--|
| SAVE | ○ | × | × | × | × | × | × | × | × | × |
| SAVE "nombre de archivo" | ○ | ○ | × | × | × | × | × | × | × | × |
| SAVE ALL | × | × | ○ | × | × | × | × | × | × | × |
| SAVE ALL "nombre de archivo" | × | × | ○ | ○ | × | × | × | × | × | × |
| SAVE, A | × | × | × | × | ○ | × | ○ | × | × | × |
| SAVE "nombre de archivo", A | × | × | × | × | ○ | ○ | ○ | ○ | × | × |
| SAVE D, dirección 1, dirección 2 | × | × | × | × | × | × | × | × | ○ | × |
| SAVE "nombre de archivo", D, dirección 1, dirección 2 | × | × | × | × | × | × | × | × | ○ | ○ |

2 Clave

Si se almacena un programa con una clave, también se almacenará la misma. A continuación damos referencias para cuando se almacenan programas junto con claves.

- (1) Cuando no se coloque una clave para la PB-770, se puede usar el mando LOAD. En este caso, existe una clave almacenada como contraseña para la máquina PB-770.
- (2) Cuando se coloque una clave igual a la que tiene un programa almacenado, se puede utilizar el mando LOAD.
- (3) Cuando la clave es diferente a la que tiene el programa almacenado, aparecerá un mensaje de error en cuanto se inicie el mando de LOAD, y entonces la cinta del cassette se detendrá.

3 Error es durante la ejecución de LOAD

(1) Error RW

Este mensaje de error aparece cuando se ha producido un error de paridad durante el ejecución de LOAD. En este caso, borre el programa que ha sido introducido, introduciendo el mando NEW  y luego oprima la tecla y vuelva a repetir el proceso desde el principio.

(2) Error OM

Este mensaje de error se produce cuando la capacidad de la memoria es menor que lo requerido. En este caso borre algunos programas innecesarios, cambie la dirección de memoria inicial para el área de datos o bien expanda la capacidad de la memoria RAM.



SAVE, VERIFY, CHAIN, PUT, GET, PASS

NEW/NEW ALL

| | |
|-----------------|---------------------------------------|
| Función | Borrado de uno o todos los programas. |
| Formatos | NEW NEW ALL |

Siempre que se desee entrar un nuevo programa, es necesario borrar el anterior. Para ello se utilizan los mandos NEW y NEW ALL.

1 Funciones del mando NEW

Este mando borra el programa almacenado en el área de programa especificada por el mando PROG. Este mando genera un error PR siempre que se intente el borrado de un programa al cual se ha asignado una clave (página 143, PASS).

Este mando no inicializa las variables (ver página 157, CLEAR).

2 Funciones del mando NEW ALL

El mando NEW ALL borra de una vez la totalidad de los programas alojados en todas las áreas de programa. Debe tenerse especial cuidado con su uso, ya que es efectivo aún después de asignada una clave.

Al igual que los programas, se borran también todas las variables, se pasa al área de programa P0, se especifica la unidad angular θ (grados) y se cancela el área de memoria para datos (ver CLEAR).

Uso

El procedimiento es el siguiente:

NEW  ó NEW ALL 

PASS

| | |
|----------------|---|
| Función | Protege programas por medio de una clave. |
| Formato | PASS "clave" |

No es poco común que, por accidente, se borre o destruya parcialmente un programa importante o para el cual han dedicado largas horas de trabajo.

Para ello, la PB-770 dispone de este mando, el cual protege los programas almacenados por medio de una clave.

Uso

1 Mando PASS

Este mando se utiliza con el siguiente formato;

PASS "clave de hasta 8 caracteres" 

Siempre que se utilice este mando, se inhabilitan todo tipo de correcciones a no ser que se cancele la clave asignada. Los mandos inhabilitados por esta función son;

- (1) AUTO
- (2) DELETE
- (3) EDIT
- (4) LIST, LLIST
- (5) NEW

Tampoco permite la entrada de un programa. El mando PASS afecta a todas las áreas de programa por igual, por lo cual resulta imposible asignarse una clave a una única área de programa. El intento de ejecución de cualquiera de los mandos arriba mencionados luego de haber usado el mando PASS, provocará la aparición en pantalla del error PR.

2 Liberación del mando PASS

Para la clave, se pueden usar hasta 8 caracteres. La clave se libera entrando la misma clave con el siguiente formato:

PASS "clave de hasta 8 caracteres" 

No obstante, si se olvidara de la clave asignada, ésta no podrá liberarse por el procedimiento aquí citado. Es por ello que se recomienda el uso de claves fáciles de recordar, como su propio nombre o número de teléfono. Recuérdese, también, que sólo puede tener un máximo de 8 caracteres. Si llegará a olvidar la clave asignada, la única solución sería ejecutar el mando NEW ALL, antes de lo cual se recomienda grabar los programas almacenados en cinta de cassette por medio del mando SAVE. De lo contrario, los perdería.

VER SAVE, LOAD, NEW

PROG

| | |
|---------|--|
| Función | Especifica un área de programa. |
| Formato | PROG valor o expresión numérica $0 \leq \text{valor o expresión numérica} < 10$ |

La PB-770 dispone de 10 áreas de programa especificadas mediante P0 hasta P9, donde se pueden almacenar programas independientes. El mando PROG sirve para especificar estas áreas de programa.

Uso

Al encenderse la PB-770, aparece en la pantalla lo siguiente:

Encendido

Ready P0

—

Al encenderse la unidad, el área de programa P0 se especifica automáticamente. Especifiquemos a continuación el área P9:

Encendido

SHIFT PROG 9 ↵

Ready P0

PROG 9

Ready P9

—

Si bien se puede utilizar una expresión numérica para la especificación PROG, su resultado debe ser igual o mayor que cero y menor que diez. Siempre que se excedan estos límites, aparecerá un error BS.

Ejemplos

PROG (100/20) → Especifica el área P5.

PROG (100/10) → Visualiza un error BS.

PROG (100/15) → Especifica el área P6 (se omite la parte decimal).



GOTO, GOSUB

RUN

| | |
|-----------------|---|
| Función | Ejecuta un programa. |
| Formatos | RUN RUN ln ($1 \leq ln \leq 9999$) |

El mando de RUN se usa para ejecutar un programa que está en el área de programa especificado en esos momentos.

El mando RUN tiene los siguientes formatos.

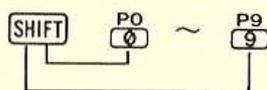
(1) RUN 

Hace iniciar la ejecución de un programa desde la primera línea del área de programa especificado. La ejecución se realiza en secuencia desde la línea con el número menor.

(2) RUN número de línea 

Este mando inicia la ejecución del programa a partir del número de línea especificado después de RUN. Si no existiera la línea especificada, la ejecución se iniciará desde la línea que tenga el valor más cercano al del valor de la línea especificada.

La ejecución se puede iniciar desde el inicio del programa en cada área de programa usando el siguiente procedimiento.



Los programas de P0 a P9 se pueden ejecutar oprimiendo la tecla **SHIFT** y las teclas de **0** a **9** al mismo tiempo.

SAVE

| | |
|-----------------|--|
| Función | Almacena un programa en cinta de cassette. |
| Formatos | SAVE SAVE ALL SAVE, A SAVE, D, dirección 1, dirección 2 SAVE "nombre de archivo" SAVE ALL "nombre de archivo" SAVE "nombre de archivo", A SAVE "nombre de archivo", D, dirección 1, dirección 2 |

Este mando sirve para almacenar sus programas en cintas de cassette. **SAVE** graba sus programas en cintas de cassette y puede tener los siguientes formatos, según su aplicación:

- (1) **SAVE** Graba en formato binario el programa del área en curso.
- (2) **SAVE ALL** Graba en formato binario los programas de todas las áreas.
- (3) **SAVE, A** Graba en formato ASCII el programa del área en curso.
- (4) **SAVE, D, dirección 1, dirección 2**
 Graba en código binario la información contenido en la memoria, desde la dirección 1 hasta la 2. La dirección 1 no puede ser mayor que -32769, mientras que la 2 tampoco puede ser mayor que 65536. Para mayor información sobre las direcciones, remitase a la parte dedicada a **CLEAR**.

- La grabación por medio del mando **SAVE** lleva menos tiempo y consume menos cinta mediante el formato binario (código de formato interno). No obstante, para ejecutar **LOAD, M** (ver mando **LOAD**), es necesario usar el mando **SAVE** con el formato del código ASCII.

Uso

La asignación de nombres de archivo a los programas grabados en cinta de cassette facilita su futura localización.

1 Nombres de archivo

SAVE "nombre de archivo"

SAVE ALL "nombre de archivo"

SAVE "nombre de archivo", A

SAVE "nombre de archivo", D, dirección 1, dirección 2

- El nombre de archivo puede tener una longitud máxima de 8 caracteres (o símbolos).
- En los programas a los cuales se han asignado claves, esta última también se graba juntamente.

2 Contador de avance de la cinta

La grabación comienza al presionarse la tecla  con la grabadora en el modo de grabación. La función de control remoto detiene la cinta automáticamente una vez finalizada la grabación. Se recomienda tomar nota de las lecturas del indicador de avance de la cinta al comienzo y al final de la grabación.

3 Confirmación

El mando VERIFY permite verificar si la grabación se llevó a cabo correctamente o no (página 152, VERIFY). La grabación, mediante SAVE, deberá repetirse si es que no ha sido correcta.

Atributos de archivos

Los nombres de los atributos de archivos de programas y datos, que han sido almacenados por SAVE o PUT, aparecen cuando son trasladados a la computadora por medio de los mandos LOAD o GET.

| Operación para almacenarlo | Aparece en pantalla durante la extracción y carga en la computadora |
|---|---|
| SAVE | PF B |
| SAVE ALL | AF B |
| SAVE, A | PF A |
| SAVE, D, dirección 1, dirección 2 | DF B |
| SAVE "ABC" | ABC PF B |
| SAVE ALL "ABC" | ABC AF B |
| SAVE "ABC", A | ABC PF A |
| SAVE "ABC", D dirección 1, dirección 2 | ABC DF B |

Los nombres de los archivos aparecen tal y como son.
El sentido de los atributos de los archivos es el siguiente.

Aparece durante el traslado a la computadora.

PB-770 PF B
Nombre de (A)(B) (C)
archivo

- (A) P : Programa
A : Todos los programas
D : Datos
- (B) Archivo
- (C) B : Binario (formato de código interno)
A : ASCII (formato del código ASCII)

SYSTEM

| | |
|----------------|---|
| Función | Visualiza el área de programa, la unidad angular en curso, la capacidad de memoria, la cantidad de memoria disponible en bytes y la dirección inicial del área de memoria para datos. |
| Formato | SYSTEM |

Entre, primeramente, SYSTEM , lo que hará aparecer algo similar a lo ilustrado debajo. Habrán ciertas diferencias, según si se utiliza o no el módulo RAM, si hay programas almacenados y del área para datos disponible.

Entre el mando SYSTEM y oprima la tecla , y si no hubiera programa o datos almacenados en la PB-770 aparecerá lo siguiente.

```

P 0123456789 ANGLE 0
  8KB 6871B
Ready P0
    
```

Tenga en cuenta las siguientes consideraciones para cada parte de lo visualizado.

- (1) La indicación P 0123456789 en la primera línea indica si hay o no programas almacenados en cada una de las áreas de programas (P0 a P9). El símbolo ♥ indica la presencia de un programa en el área correspondiente.
- (2) La indicación ANGLE 0; también en la primera línea, indica la unidad angular especificada. La indicación para los grados (ANGLE 0) es la que aparece siempre que se enciende la unidad (ANGLE 1 = radianes y ANGLE 2 = gradientes).
- (3) La indicación 8KB en la segunda línea indica la capacidad de memoria. Este valor cambia, según la cantidad de módulos RAM utilizados.

| <u>RAM</u> | <u>Cantidad</u> | <u>Capacidad de memoria</u> |
|------------|-----------------|-----------------------------|
| OR-8 | Ninguno | 8KB |
| OR-8 | 1 | 16KB |
| OR-8 | 2 | 24KB |
| OR-8 | 3 | 32KB |

- (4) La indicación "6871B" en la segunda línea es el número de bytes disponibles (sin usar). Este valor cambia según la cantidad de módulos RAM utilizados y según el área de memoria destinada al almacenamiento de datos.
- (5) En la misma línea, a continuación de la cantidad de bytes disponibles, aparece la dirección a partir de la cual comienza el área de memoria para datos. Cambia siempre que se lleva a cabo una nueva especificación y desaparece cuando no se especifica área de memoria para datos alguna.
- (6) La indicación "Ready P0" en la tercera línea indica la posibilidad de entrar un programa en el área de programa P0. Esta área de programa se especifica automáticamente siempre que se enciende la unidad.

ANGLE, CLEAR

VERIFY

| | |
|-----------------|--|
| Función | Verifica la paridad de los programas o de los datos grabados en una cinta de cassette. |
| Formatos | VERIFY VERIFY "nombre de archivo" |

El mando VERIFY verifica si los programas o los datos almacenados en una cinta de cassette a través de los mandos SAVE o PUT han sido grabados correctamente.

- Este mando tiene los siguientes formatos.
 - (1) VERIFY
Verifica el primer programa encontrado en la cinta de cassette.
 - (2) VERIFY "nombre de archivo"
Verifica el programa que tenga el mismo nombre en la cinta.
- Usted puede verificar todos los programas almacenados por los 8 formatos de SAVE (consulte SAVE) usando el mando VERIFY y la tecla  o bien, VERIFY "nombre de archivo" y la tecla . En otras palabras, no es necesario especificar el formato de SAVE.
- Si el mando SAVE no fue introducido correctamente, aparecerá en la pantalla un error RW. Si ocurriera esto, grabe los programas otra vez.

Usos

Los procedimientos son los siguientes.

- Paso 1:** Rebobinado de la cinta: regrese la cinta en donde están almacenados los programas (que fueron almacenados con el mando SAVE) a la posición inicial usando el contrador de la cinta.
- Paso 2:** Introducción del mando VERIFY: si el programa tiene un nombre de archivo, introduzca la orden VERIFY "nombre de archivo" y oprima la tecla ; pero si no tiene un nombre, introduzca la orden de VERIFY y oprima la tecla .
- Paso 3:** Oprima el botón PLAY de la grabadora de cassette. Cuando se inicia la verificación, cualquiera de los mensajes PF B, AF B, PF A DF B o DF A (para esto consulte la pág. 149) aparecerá en la pantalla. Si un programa o la información fueron correctamente almacenados, el término Ready P0—P9 aparecerá en la pantalla y la cinta de cassette se detendrá. Si el programa o la información no hubieran sido correctamente almacenados, aparecerá un error RW y la cinta de cassette se detendrá.

BEEP

| | |
|-----------------|---------------------------|
| Función | Genera una señal audible. |
| Formatos | BEEP BEEP 0 BEEP 1 |

El mando de BEEP que tiene la computadora PB-770 sirve para generar señales audibles.

Hay muchas maneras de usar las señales audibles. Por ejemplo, cuando se requiere un largo período de tiempo para la ejecución de un programa, el término de la ejecución se puede indicar con la creación de una de estas señales, insertando para ello un mando BEEP al final de la ejecución del programa. También puede ser divertido introducirlo en un juego.

El mando BEEP tiene los siguientes tres formatos.

- (1) **BEEP**
Genera un sonido grave.
- (2) **BEEP 0**
Genera el mismo sonido que BEEP.
- (3) **BEEP 1**
Genera un sonido ligeramente más agudo que los anteriores.

PROGRAMA DE EJEMPLO

```

100 REM*** BEEP ***
110 IF INKEY$="" THEN 110
120 IF INKEY$="0" THEN BEEP 0
130 IF INKEY$="1" THEN BEEP 1
140 GOTO 110

```

Este programa fue preparado para generar sonidos graves cuando se oprime la tecla **0** , y agudos cuando se oprime la tecla **1** .

CHAIN

| | |
|-----------------|--|
| Función | Traslada un programa especificado y lo ejecuta desde la primera línea. |
| Formatos | CHAIN CHAIN "nombre de archivo" |

Cuando el mando CHAIN aparece durante la ejecución de un programa, la ejecución del programa se detiene en ese punto; entonces se recupera el programa con el nombre especificado en el mando CHAIN de una cinta de cassette y se lo almacena en la memoria de la computadora para su ejecución desde el principio.

Si no se especifica el nombre de programa, se recupera el programa que fuera encontrado primero grabado con SAVE o SAVE "nombre de archivo".

El mando CHAIN tiene los siguientes formatos.

(1) CHAIN

Recupera y almacena en la memoria de la computadora el PF B que se encuentre primero para su ejecución (programa grabado con el mando SAVE ó SAVE "nombre de archivo").

(2) CHAIN "nombre de archivo"

Recupera de la cinta y almacena en la computadora el PF B con el nombre de archivo especificado para su ejecución.

- Debido a que el programa se almacena en la computadora en el área de programa especificada en ese momento, el programa anterior se borra como con NEW.
- Los programas almacenados con SAVE ALL y SAVE, A no pueden recuperarse y almacenarse en la computadora con el mando CHAIN.
- Si al programa recuperado y almacenado en la computadora se le hubiese asignado una clave, la misma acompañará al programa.
- Aun cuando se ejecute el mando CHAIN, las variables no son eliminadas.

Usos

Introduzca las listas de 1 a 3 en las áreas del programa P1 a P3 y luego almacénelas en una cinta de cassette.

- < Lista 1 > Programa que calcula el área de un círculo.
(Nombre de archivo: "PRO. 1")
- < Lista 2 > Programa que calcula el área de un triángulo.
(Nombre de archivo: "PRO. 2")
- < Lista 3 > Programa que calcula el área de un cuadrado.
(Nombre de archivo: "PRO. 3")

Además, introduzca la List a 0 en P0 y ejecútela. Se usa List a 0 para seleccionar entre la List a 1 a List a 3 por el mando CHAIN en las líneas 60 a 80, para recuperarlas y almacenarlas en la computadora PB-770, y ejecutar los cálculos.

Lista 0

```
10 REM CHAIN PRO.0
20 CLS :PRINT "AREA CALCULATIONS"
30 PRINT "1CIRCLE 2TRIANGLE 3RECTANGL
   E"
40 PRINT "SELECT NO.   "
50 BB$=INKEY$IF VAL(BB$)>3 THEN 50 EL
   SE IF VAL(BB$)<1 THEN 50
60 IF BB$="1" THEN CHAIN "PRO.1"
70 IF BB$="2" THEN CHAIN "PRO.2"
80 CHAIN "PRO.3"
```

Lista 1

```
10 REM PRO.1
20 INPUT "RADIUS";RR
30 S=PI*RR^2
40 PRINT S
50 END
```

Lista 3

```
10 REM PRO.3
20 INPUT "LENGTH";HH
30 INPUT "WIDTH";LL
40 S=HH*LL
50 PRINT S
60 END
```

Lista 2

```
10 REM PRO.2
20 INPUT "HEIGHT";HH
30 INPUT "BASE";LL
40 S=HH*LL/2
50 PRINT S
60 END
```

CLEAR

| | |
|----------------|--|
| Función | Borra todas las variables. Borra todas las variables y crea un área para datos. |
| Formato | CLEAR (dirección inicial del área para datos) |

El mando CLEAR se usa para borrar todas las variables numéricas y las variables de caracteres y símbolos.

Las variables numéricas se borran a 0 y las variables de caracteres a nada " " (es decir sin carácter). Al mismo tiempo, se eliminan las variables registradas en el programa, y las variables de matriz definidas.

Debido a que se borra también la memoria intermedia para FOR los lazos FOR-NEXT no puede continuar.

Al especificarse la dirección de memoria inicial para un área de datos, se reserva en la memoria un área donde se pueden escribir datos mediante el mando POKE y leerlos mediante la función PEEK. Este método es muy útil para el almacenamiento de información importante.

Usos

(1) Borrado de variables

El programa siguiente se usa para mostrar datos que representan la suma de los datos y el número de datos. Sin embargo, si el programa es ejecutado una vez más después de haber oprimido la tecla **BRK**, la respuesta correcta no se puede obtener debido a que los valores numéricos asignados a las variables S y N en la primera ejecución se mantienen. Por tanto, el mando CLEAR fue insertado entre la línea 10 y la línea 30, como se muestra en el programa, para que se puede obtener una respuesta correcta cada vez que se realice una ejecución.

```

10 PRINT "TOTAL "
30 INPUT "D=", D
40 S=S+D
50 N=N+1
60 PRINT "S(";N;")=";S
70 GOTO 30

```

```

10 PRINT "TOTAL "
20 CLEAR
30 INPUT "D=", D
40 S=S+D
50 N=N+1
60 PRINT "S(";N;")=";S
70 GOTO 30

```

(2) Creación de un área para datos

Antes que nada, forme un área para datos usando el mando CLEAR. Supogamos que el programa de arriba se encuentra en el área de programa P0. Realice la siguiente entrada:

CLEAR 3000 ↵

Ello hará que la memoria de la dirección 3000₍₁₀₎ en adelante se reserve para el almacenamiento de datos.

Luego verifique el número restante de bytes mediante la siguiente entrada:

SYSTEM ↵

CLEAR 3000 ↵
SYSTEM ↵

```
P ♥ 123456789 ANGLE 0
      8KB 1615B 3000
Ready P0
_
```

El valor resultante debe ser 1615 bytes, ello siempre que no haya programa alguno en otra área y que no se haya ampliado la memoria RAM. Entre entonces:

CLEAR 5000 ↵

y ejecute el mando SYSTEM nuevamente.

CLEAR 5000 ↵
SYSTEM ↵

```
P ♥ 123456789 ANGLE 0
      8KB 3615B 5000
Ready P0
_
```

Corrobore que el número de bytes haya aumentado. De este modo, el número de bytes disponible para los programas en BASIC y sus variables es indirectamente proporcional al tamaño del área para datos especifica. Siempre que se utilizan PEEK y POKE para, respectivamente, leer y escribir datos en las direcciones de memoria especificadas, es necesario crear el área para datos usando el mando CLEAR.

Este mando (CLEAR) pone todas las variables, numéricas y de caracteres, en su estado inicial. No obstante, el contenido del área para datos permanece intacto. Por lo tanto, no es necesario preocuparse por el área para datos en programación común (cuando no se utilizan PEEK y POKE).

La información almacenada en las áreas para datos puede grabarse y recuperarse de cintas de cassette más rápidamente mediante el uso de los mandos LOAD, D, dirección y SAVE, D, dirección 1, dirección 2, en lugar de PUT y GET.

(3) Especificación de la dirección de memoria inicial

La memoria inicial que puede especificarse debe ser mayor que -32769 y menor que 65536. Por otro lado, la dirección debe ser mayor que las direcciones destinadas al almacenamiento del sistema (&H0000 hasta &H0528) y que la última dirección de un programa, siempre que haya alguno almacenado en la memoria.

El valor especificado para la dirección inicial del área para datos y la dirección de memoria que en realidad se especifica tienen la siguiente relación:

| Dirección especificada (hexadecimal) | Valor especificado para la dirección inicial (decimal) |
|---|---|
| 0000H | 0, 32768, -32768 |
| 2000H | 8192, 40960, -24576 |
| 4000H | 16384, 49152, -16384 |
| 6000H | 24576, 57344, -8192 |
| 7FFFH | 32767, 65535, -1 |

NOTA: 0 a 1320, 32768 a 34088 y -32768 a -31448 son áreas que utiliza el sistema. No pueden, por lo tanto, especificarse (aparece un error OM).

A continuación se muestran las posiciones de memoria que pueden especificarse, según la capacidad de la RAM. Para aquellos programas que no usan PEEK y POKE, no es necesario crear áreas para datos. En caso de que se haya creado una innecesariamente, bórrela usando el mando CLEAR como se muestra en la tabla a continuación.

De este modo, se puede aprovechar toda la memoria disponible para el programa y sus variables.

El uso de un mando CLEAR que no especifica área para datos alguna no modifica la capacidad del área que pueda haberse creado con anterioridad.

| Capacidad de la RAM | Direcciones que pueden usarse (decimal) | Cuando no se crea área para datos |
|---------------------|---|-----------------------------------|
| 8KB | 0~8191 | CLEAR 8192 |
| 16KB | 0~16383 | CLEAR 16384 |
| 24KB | 0~24575 | CLEAR 24576 |
| 32KB | 0~32767 | CLEAR 32767 |

* Si la especificación de un área para datos se borra mediante NEW ALL  , no aparece en el pantalla el número de dirección.

VER SYSTEM, PEEK, POKE, NEW ALL

CLS

| | |
|---------|---|
| Función | Borra todo de la pantalla y coloca el cursor en la posición inicial (extremo superior izquierdo). |
| Formato | CLS |

El mando CLS se usa para borrar todo lo que esté en la pantalla y para colocar al mismo tiempo el cursor en la posición original (extremo izquierdo superior).

Se usa para borrar la pantalla antes de trazar gráficas en la pantalla.

- (1) Cuando este mando se realiza manualmente, el cursor se coloca en la posición correspondiente a 0, 1.
- (2) Cuando se ejecuta este mando en un programa, el cursor aparece en la posición (0, 0).

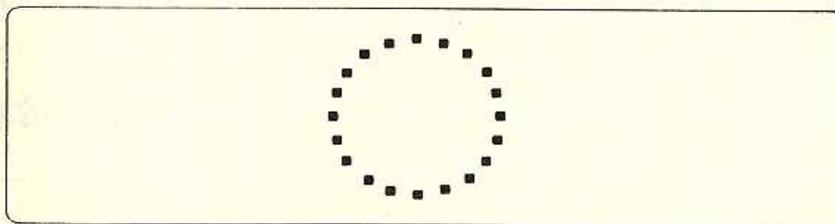
Usos

```

5 ANGLE 0
10 CLS
20 FOR I=0 TO 360 STEP 12
30 DRAW(SINI*15+80,COSI*15+15)
40 NEXT I
50 END

```

Este programa se usa para trazar en la pantalla el siguiente patrón.



Para trazar en la pantalla un patrón como se muestra en el ejemplo anterior, se debe borrar antes la pantalla. Por esa razón, el mando CLS se debe usar al principio del programa.

DIM

| | |
|----------------|--|
| Función | Declara una matriz. |
| Formato | DIM nombre de variable matricial (subíndice) [, nombre de variable matricial (subíndice)] |

El mando DIM prepara una matriz en el área de memoria, con el nombre especificado. El mando DIM puede crear variables matriciales numéricas de precisión simple, numéricas de precisión media y de caracteres.

DIM nombre de variable matricial (subíndice [, nombre de variable matricial (.)
(Valor máximo del subíndice: 255)

Ejemplos de sentencias de declaración de varias serie de datos:

| Formato | Clasificación | Ejemplo |
|---|---|-------------------|
| DIM nombre variable matricial (1) | Matriz numérica de una dimensión de precisión simple. | DIM A (5) |
| DIM nombre variable matricial (I, J) | Matriz numérica de dos dimensiones de precisión simple. | DIM A (2, 3) |
| DIM nombre variable matricial (! (I) | Matriz numérica de una dimensión de media precisión. | DIM A! (5) |
| DIM nombre variable matricial ! (I, J) | Matriz numérica de dos dimensiones de media precisión. | DIM A! (2, 3) |
| DIM nombre variable matricial \$ (I) * N | Matriz de caracteres de una dimensión. | DIM A\$ (5)*20 |
| DIM nombre variable matricial \$ (I, J) * N * puede omitirse | Matriz de caracteres de dos dimensiones. | DIM A\$ (2, 3)*20 |

I, J y N representan números enteros o expresiones numéricas dentro de $0 \leq I < 256$, $0 \leq J < 256$ y $0 \leq N < 80$, en donde la parte fraccional de valor numérico se descarta.

El nombre de las variables matriciales se representa por un carácter alfabético en mayúsculas, (A a Z).

El valor de dimensión máximo es 2, lo que significa que pueden especificarse matrices de una y dos dimensiones.

Una matriz numérica de media precisión se puede especificar colocando el signo “!” justo después del nombre de la variable matricial; además, una matriz de caracteres se puede especificar colocando el signo “\$” just después del nombre de la variable matricial.

Usos

Introduzca el siguiente programa y ejecútelo.

```

10 CLEAR
20 DIM A(2,3), B(2,3)
100 END

```

Se puede introducir una línea con una expresión operacional entre las líneas 20 y 100.

Cuando aparezca el mensaje “Ready P0” después de haber ejecutado el programa, verifique la lista de variables matriciales del modo siguiente.

```

LIST V ↵
A( ) B( )
Ready P0

```

El nombre de la variable matricial asignada por la orden DIM se puede confirmar usando

```
LIST V ↵
```

como fue mencionado antes.

Verifiquemos el contenido de cada variable de la serie agregando la siguiente lista al programa de arriba. Se debe declarar una variable matricial mediante la orden DIM antes de usarlo.

```

30 FOR I=0 TO 2
40 FOR I=0 TO 3
50 PRINT A(I,J);B(I,J);
60 NEXT J:NEXT I

```

Cuando ejecute el programa despues de haber agregado las líneas 30 a 60, aparecerá lo siguiente en al pantalla.

■ Ejemplo de ejecución

```

  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0
Ready P0

```

Aquí, el contenido de las 24 variables de la matriz, $A(0,0)$, $B(0,0)$ — $A(2,3)$, $B(2,3)$, se muestra como "0".

Es importante notar que el contenido de todas las matrices se transforma en 0 cuando se ejecuta el mando DIM.

Mientras el contenido de las matrices numéricas mencionadas antes es 0, cuando se declaran matrices de caracteres, su contenido se transforma en ausencia de valor, en cuyo caso nada aparece en la pantalla. Ningún valor quiere decir, aquí, que la matriz de caracteres está vacía. Debe tenerse en cuenta la diferencia que hay entre series de espacios y series vacías. Una serie de espacios es una serie qui tiene un espacio (A(I) = " "$) y una serie vacía es una serie sin valores (A(I) = ""$).

PROGRAMA DE EJEMPLO 1

* Programa para ordenar (una matriz numérica de una dimensión)

```

10 CLEAR
20 DIM D(5)
30 FOR I=1 TO 5
40 PRINT "DATA"; I; " ="; INPUT D(I)
50 NEXT I
60 REM SORT
70 F=0
80 FOR I=1 TO 4

```

```

90 IF D(I)<D(I+1) THEN X=D(I):D(I)=D(
    I+1):D(I+1)=X:F=1
100 NEXT I
110 IF F=1 THEN 70
120 REM RESULT
130 FOR I=1 TO 5
140 PRINT D(I);
150 NEXT I
160 END

```

Este programa sirve para ordenar 5 números que se han introducido, haciéndolos aparecer en la pantalla ordenados en secuencia colocado primero el de mayor valor.

Si bien el mando DIM de la línea 20 declara una matriz con los elementos D(0) hasta D(5), en este programa sólo se usan D(1) hasta D(5).

Las líneas 60 a 110 contienen la parte del programa que sirve para ordenar, mientras que las líneas 120 a 150 contienen la porción que sirve para hacer aparecer en la pantalla el resultado de la operación de ordenación mencionada.

Es conveniente usar variables matriciales combinándolas con el mando FOR-NEXT, tal y como se muestra en el programa de ejemplo.

PROGRAMA DE EJEMPLO 2

* Obtención de total en forma vertical y horizontal (Matriz numérica de dos dimensiones).

```

5 CLEAR
10 DIM A(3,3),X(3),Y(3)
20 FOR I=1 TO 3
30 FOR J=1 TO 3
40 PRINT "<; I; ", "; J; ")"; "="
50 INPUT A(I,J)
60 NEXT J:NEXT I
70 REM SUBTOTAL

```

```

80 FOR I=1 TO 3
90 FOR J=1 TO 3
100 X(I)=X(I)+A(I, J)
110 Y(J)=Y(J)+A(I, J)
120 NEXT J:NEXT I
130 REM RESULT
140 FOR I=1 TO 3
150 PRINT "X("; I; ")="; X(I)
160 PRINT "Y("; I; ")="; Y(I)
170 FOR K=1 TO 500:NEXT K
180 NEXT I
190 END

```

Este programa asigna los datos de la Tabla 1 a un cuadro (matriz) de dos dimensiones que se muestra en la Tabla 2, para obtener subtotales en forma horizontal, X(1), X(2), X(3), y en forma vertical, Y(1), Y(2), Y(3).

Tabla 1.

| | | | |
|-------|-------|-------|-------|
| 14 | 9 | 21 | X (1) |
| 35 | 4 | 53 | X (2) |
| 6 | 15 | 11 | X (3) |
| Y (1) | Y (2) | Y (3) | |

Tabla 2

| | | | |
|----------|----------|----------|-------|
| A (1, 1) | A (1, 2) | A (1, 3) | X (1) |
| A (2, 1) | A (2, 2) | A (2, 3) | X (2) |
| A (3, 1) | A (3, 2) | A (3, 3) | X (3) |
| Y (1) | Y (2) | Y (3) | |

Las líneas 80 a 120 sirven para obtener un subtotal, mientras que las líneas 130 a 180 sirven para presentar en la pantalla los valores obtenidos.

VER CLEAR, NEW ALL, ERASE, LIST V

DRAW/DRAWC

| | |
|------------------|---|
| Funciones | DRAW: Dibuja un punto DRAWC: Borra un punto |
| Formatos | DRAW (X ₁ , Y ₁) [-(X ₂ , Y ₂)] DRAWC (X ₁ , Y ₁) [-(X ₂ , Y ₂)] |

El mando DRAW traza puntos o líneas en la pantalla, mientras que DRAWC los borra.

Debido a que no sólo caracteres sino también puntos y líneas pueden dibujarse en la pantalla, se pueden hacer muchos tipos de gráficos.

Las coordenadas que se pueden especificar con la orden DRAW (X, Y) o con DRAWC (X, Y) son las siguientes.

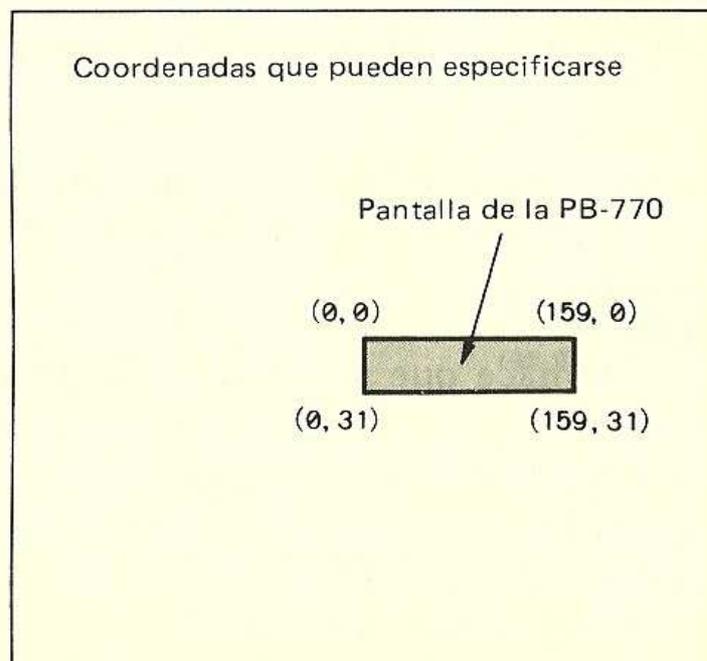
$$-255.5 < X < 255.5$$

$$-255.5 < Y < 255.5$$

Debido a que el alcance de las coordenadas de los puntos en la pantalla es de $0 \leq X \leq 159$, y $0 \leq Y \leq 31$, se puede considerar la siguiente pantalla virtual.

(-255.5, -255.5)

(255.5, -255.5)



(-255.5, 255.5)

(255.5, 255.5)

En esta figura,  es equivalente a la pantalla de la PB-770. El extremo izquierdo superior de la pantalla es el origen (0, 0).

Usos

Los mandos DRAW y DRAWC se usan en los siguientes casos.

| | |
|---|--|
| Draw (X ₁ , Y ₁) | Dibuja un punto en las coordenadas (X ₁ , Y ₁). |
| DRAWC (X ₁ , Y ₁) | Borra el punto en las coordenadas (X ₁ , Y ₁). |
| DRAW (X ₁ , Y ₁)—(X ₂ , Y ₂) | Dibuja una línea de las coordenadas (X ₁ , Y ₁) a (X ₂ , Y ₂). |
| DRAWC (X ₁ , Y ₁)—(X ₂ , Y ₂) | Borra la línea de las coordenadas (X ₁ , Y ₁) a (X ₂ , Y ₂). |

Las X y Y mencionadas aquí representan valores numéricos, nombres de variables, y expresiones numéricas con las siguientes gamas:

$$-255.5 < X < 255.5 \qquad -255.5 < Y < 255.5$$

Aparecen en la pantalla redondeados a dos enteros.

El siguiente programa permite dibujar un rectángulo en la pantalla.

```

10 REM DRAW
20 CLS
30 DRAW(10,10)—(10,20)—(150,20)—(150,10)—(10,10)
40 END
    
```

Se puede dibujar una figura por medio de coordenadas contínuas con el signo “—”, tal y como se muestra en el programa siguiente.

PROGRAMA DE EJEMPLO

* Este es un programa que hace que aparezca en la pantalla un carácter ampliado dos veces.

```

10 CLEAR :DIM A(7,7): CLS
20 K$=INKEY$
30 IF K$="" THEN 20
40 LOCATE 19,0:PRINT K$
50 FOR I=0 TO 7
60 FOR J=0 TO 7
70 A(I,J)=POINT(I+152,J)
80 NEXT J:NEXT I
    
```

```

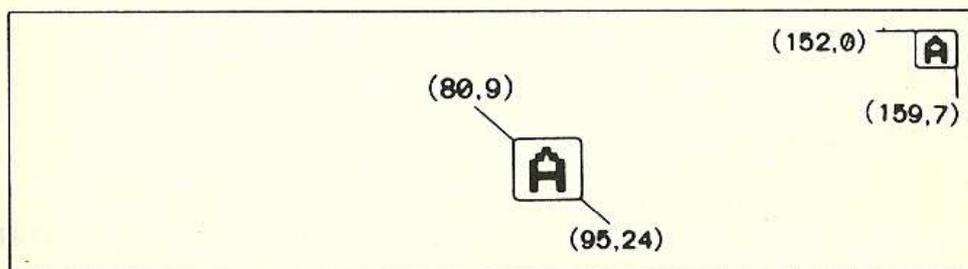
90 FOR I=0 TO 7
100 FOR J=0 TO 7
110 IF A(I, J)<>1 THEN 160
120 DRAW(2*I+80, 2*J+9)
130 DRAW(2*I+81, 2*J+9)
140 DRAW(2*I+80, 2*J+10)
150 DRAW(2*I+81, 2*J+10)
160 NEXT J:NEXT I
170 LOCATE 0, 1
180 END

```

Este programa usa el mando INKEY\$ para introducir un carácter, y primero exhibe el carácter en la posición que se ha especificado por medio del mando LOCATE (19, 0), tal y como se muestra en la figura siguiente.

Este carácter se dibuja en al pantalla usando puntos en una área cuadrada con una línea diagonal formada por (152,0)-(159,7) en donde los puntos encendidos y apagados son verificados con la función POINT, después los valores de 0 y 1 son asignados a la variable matricial A (I, J).

Usando esta variable matricial como dato, el carácter que es amplificado dos veces aparecerá en una área cuadrada con una diagonal formada por (80,9)-(95,24).



VER POINT

END

| | |
|----------------|---|
| Función | Hace que termine la ejecución de un programa. |
| Formato | END |

El mando END se usa para hacer terminar la ejecución de un programa. Los pasos que estén contenidos en un lazo de programa (lazo FOR-NEXT y GOSUB) se pueden suspender mediante un mando END. Se puede colocar en un programa cualquier número de sentencias de END mientras tengan sentido.

El mando END colocado al fin del programa puede omitirse.

PROGRAMA DE EJEMPLO

```

500 FOR I = 0 TO 1000
510 K$ = INKEY$
520 IF K$ = "A" THEN 1000
530 IF K$ = "B" THEN 2000
540 IF K$ = "C" THEN 3000
550 IF K$ = "D" THEN END
560 NEXT I
570 END
:
:

```

Este programa se puede usar como parte de otro programa para presentar un menú.

Después de introducido "A", "B", o "C" el programa salta a la línea 1000, 2000 ó 3000, respectivamente.

Cuando se introduce "D", la ejecución del programa se termina inmediatamente. Sin embargo, cuando se introduce otra letra fuera de las mencionadas (de A a D), o cuando no se introduce nada, la ejecución termina automáticamente después de un cierto período de tiempo.

VER STOP

ERASE

| | |
|----------------|--|
| Función | Libera las variables registradas y las variables matriciales |
| Formato | ERASE nombre de variable [, nombre variable] |

El mando ERASE puede liberar las variables registradas y variables matriciales que se pueden confirmar con LIST V.

La especificación de un nombre de variable para que sea liberada se realiza de la siguiente forma:

Nombres de variables

mostradas por LIST V: AB, AB\$, A (), A! (), A\$ ()

ERASE: AB, AB\$, A, A!, A\$

La ejecución del mando ERASE borra la memoria temporal destinada a los lazos FOR-NEXT, por lo cual no puede utilizar dentro de estos lazos. Cuando se especifica el nombre de una variable que no se ha registrado, la ejecución pasa a la siguiente operación.

VER

DIM, LIST V

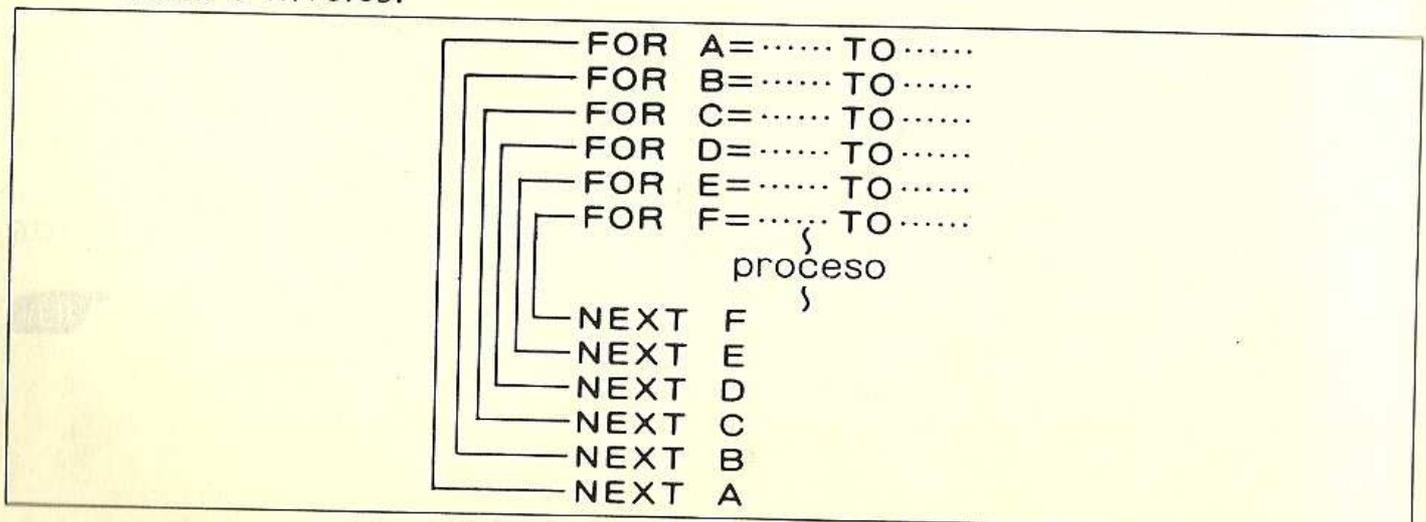
FOR~TO~STEP/NEXT

| | |
|-----------------|--|
| Función | Repite la ejecución de un lazo desde FOR hasta NEXT, el número de veces especificado. |
| Formatos | <pre>FOR variable numérica = i TO j [STEP k] } NEXT variable numérica (misma variable que la de FOR) i : valor inicial j : valor final k : incremento.</pre> |

Los mandos FOR-NEXT sirven para ejecutar el número de veces que se especifica las sentencias comprendidas entre FOR y NEXT.

Estas ordenes tienen sin embargo las siguientes limitaciones.

- (1) Las variables deben ser variables numéricas.
- (2) El paso (STEP) k se puede omitir. En este caso funciona automáticamente un incremento positivo de uno (equivalente a STEP 1).
- (3) Se puede usar una variable o una expresión numérica para el valor inicial de i , el valor final de j , y el incremento de k .
- (4) Si el valor de k fuera mayor que 0 ($k > 0$) e i mayor que j ($i > j$), la ejecución del lazo FOR-NEXT se realizará solo una vez.
- (5) Si el valor de k fuera menor que 0 ($k < 0$) y el del i menor que j ($i < j$), entonces, la ejecución del lazo FOR-NEXT se realizará solo una vez.
- (6) Cuando se ejecuta un lazo FOR-NEXT y el recorrido del lazo se dirige a la siguiente vez (se mueve a la siguiente línea), la variable que sirve para controlar el número de veces de ejecución se incrementa en uno (n) multiplicado por la unidad de incremento (k), es decir $i + nk$, lo que evidentemente es mayor que j . (n =entero).
- (7) Es posible introducir dentro del lazo FOR-NEXT otros sublazos, hasta 6 niveles.



- (8) Si el flujo de las líneas de un lazo FOR-NEXT fuera ilógico, se producirá y se anunciará un error FO.

```

FOR A=..... TO .....
FOR B=..... TO .....
NEXT A
NEXT B

```

- (9) Cuando no se ejecute FOR y aparezca NEXT, se producirá un error FO.
- (10) No se puede omitir la variable del mando NEXT.
- (11) Si se ejecuta ERASE o CLEAR, ocurrirá un error FO en la siguiente sentencia de NEXT, debido a que se ha inicializado el contenido del lazo FOR-NEXT.
- (12) Los valores de las variables de control se borran antes de llevarse a cabo el ciclo de todos los lazos. Por lo tanto, para lazos separados dentro de un mismo programa se puede utilizar la misma variable de control sin necesidad de borrarla cada vez.
- (13) Un salto dentro de un lazo usando GOTO, o GOSUB no se puede realizar.

Usos

El siguiente programa se usa para demostrar el cambio de una variable, dependiendo del valor inicial, valor final y el incremento.

```

10 INPUT "FOR I=",I,"TO",J,"STEP",K
20 PRINT "FOR I=";I;"TO";J;"STEP";K
30 FOR A=I TO J STEP K
40 PRINT "VARIABLE:A=";A
50 FOR X=1 TO 100:NEXT X
60 NEXT A
70 PRINT :GOTO 10

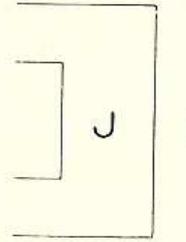
```

A continuación se dan algunos ejemplos de ejecución.
En el siguiente programa, el lazo FOR-NEXT tiene dos niveles.

```

10 CLEAR
20 DIM A(9,9)
30 FOR I=1 TO 9
40 FOR J=1 TO 9
50 A(I,J)=I*J
60 NEXT J
70 NEXT I

```



En este ejemplo, la multiplicación se realiza y el resultado es asignado a una variable de seire A (I, J) entre los dos lazos de FOR-NEXT.

Como la ejecución se realiza 9 veces por el lazo interno FOR-NEXT, y 9 veces también por el lazo externo FOR-NEXT, un total de 9×9 operaciones son realizadas.

En el ejemplo de arriba, debe notarse que el lazo interno (J) está totalmente encerrado dentro del lazo externo (I). Esto es necesario siempre que se incluyen lazos (ver el punto 7 en la página 172). Ello tiene también validez para los niveles 3 al 6.

El siguiente programa cambia la operación saltando del circuito dependiendo del resultado aritmético de la sentencia FOR-NEXT.

```

10 CLEAR
20 FOR I=1 TO 100
30 X=X+I
40 IF X >= 1000 THEN 100
50 NEXT I
60 END
100 PRINT I; X
110 END

```

Este programa realiza la suma de $1+2+3 \dots$, y cuando el resultado excede 1000, salta a la línea 100 y hace aparecer en la pantalla el valor de la variable de ese momento y el resultado de la suma.

Frecuentemente se usa un método para salirse del circuito en medio de él usando el mando IF-THEN.

También se puede salirse del circuito mediante un mando GOSUB, para continuar su ejecución usando un mando RETURN.

El siguiente programa representa un ejemplo cuando no existe nada dentro del lazo FOR-NEXT.

```

10 LOCATE 8, 2 : PRINT "HIT!";
20 FOR I=0 TO 50
30 NEXT I
40 CLS
50 FOR I=0 TO 50
60 NEXT I
70 GOTO 10

```

Allí no hay ninguna orden para procesar entre FOR y NEXT, en las líneas 20 a 30 y las líneas 50 a 60. Sin embargo, los mandos FOR y NEXT de los lazos son ejecutados el número de veces especificado. Estos mandos, lazos sin nada que realizar, que parecen no tener sentido, se usan frecuentemente para que pase un determinado tiempo y se denominan "lazos de espera".

En el programa, la expresión "HIT!" aparece en la pantalla por un cierto tiempo, se borra por otro espacio de tiempo y vuelve a aparecer por el mismo espacio de tiempo.

Como el valor final de la variable aumente, el tiempo de espera también se prolonga.

PROGRAMA DE EJEMPLO

* Este es un programa que hace aparecer en la pantalla el signo "*" en la primera línea, y en las siguientes líneas va incrementando el número de ellos, uno por vez.

```

10 CLS
20 N=1
30 FOR I=1 TO N
40 PRINT "*";
50 NEXT I
60 PRINT
70 N=N+1
80 IF N>=20 THEN END ELSE 30

```

El programa incrementa el valor final del mando FOR-NEXT, uno cada vez, e incrementa el número de "*" que aparecen en la pantalla, uno por vez. Su ejecución obtiene lo siguiente.

```
*  
**  
***  
****  
*****  
*****
```

VER IF~THEN~ELSE

GET

| | |
|----------|--|
| Función | Recupera datos almacenados en cinta de cassette y los asigna a una variable. |
| Formatos | GET variable [, variable] GET "nombre de archivo" variable [, variable] |

El mando GET se usa para recuperar datos almacenados en una cinta de cassette (grabados por medio del mando PUT), asignándolos a una variable.

Cuando el nombre del archivo se omite como se muestra en los formatos anteriores, se recupera del archivo que aparece primero en la cinta de cassette (MT). Los datos recuperados pueden ser secuencialmente asignados a variables diferentes colocándole una coma entre cada variable

pero sin embargo, cuando los datos son asignados a una variable numérica, el espacio que está al principio del dato es ignorado.

Uso

El siguiente programa graba los valores de las variables A, B, C y D bajo el nombre de archivo "TEST" en una cinta de cassette.

```

10 REM PUT MT
20 A=10 : B=20 : C=30 : D=40
30 PUT "TEST" A, B, C, D
40 END

```

El siguiente programa usa el mando GET para recuperar los datos del archivo "TEST" grabado en cinta de cassette.

```

10 REM GET MT
20 GET "TEST" E, F, G, H
30 PRINT E ; F ; G ; H
40 END

```

En este ejemplo, los datos de A, B, C y D son recuperados y asignados a las variables E, F, G y H. Un punto importante es que los datos son almacenados en la cinta en el orden A, B, C, D. Aun cuando los datos son extraídos en la forma de secuencia de A, B, D, D, mientras la cinta corre, el dato A es asignado a E, el B asignado a F, y así secuencialmente. Si el mando GET se ejecutara con un número de variables que excediera el número de variables usadas para grabarlos, ocurrirá un error DA.

PROGRAMA DE EJEMPLO

- * Este es un programa para almacenar un nombre de una persona y datos de él, y para extraer esos datos cuando se introduzca el nombre. Se aceptas hasta 16 caracteres por persona.

```
10 CLEAR
20 DIM N$(10),D$(10)
30 INPUT "PUT:0 GET:I END:E";M$
40 IF M$="0" THEN GOSUB 100
50 IF M$="I" THEN GOSUB 300
60 IF M$="E" THEN END
70 GOTO 30
100 I=0
110 INPUT "NAME: ";N$(I)
120 PUT N$(I)
130 IF N$(I)="0" THEN 180
140 INPUT "DATA: ";D$(I)
150 PUT D$(I)
160 I=I+1
170 IF I<10 THEN 110
180 RETURN
300 J=0
310 INPUT "NAME: ";A$
```

```

320 GET N$(J)
330 IF N$(J)="0" THEN PRINT "NOT FOUND
      ":GOTO 380
340 GET D$(J)
350 IF A$<>N$(J) THEN 370
360 PRINT A$;"=";D$(J):GOTO 380
370 J=J+1:GOTO 320
380 RETURN

```

El programa está formado de dos subrutinas.

Las líneas 100 a 180 forman la subrutina para almacenar los datos en la cinta de cassette, mientras que las líneas 300 a 380 representan la subrutina para buscar y recuperar el dato de cinta.

En la variable matricial N\$(I) se introduce un nombre, y este dato es introducido en D\$(I). Después la grabación y la recuperación se realizan con estas dos variables, como un par.

Cuando se introduce "O" para el menú de la línea 30, la subrutina de almacenamiento a partir de la línea 100 se ejecuta con la introducción de un nombre y el dato en forma continua. Hasta 11 datos se pueden introducir. Sin embargo, como es necesario colocar "0" al final del último nombre (para indicar el fin), se pueden realizar hasta 10 introducciones de datos.

Cuando se introduce "I" para el menú en la línea 30 para buscar el nombre de una persona, entonces se recuperan el nombre y el dato de la cinta de cassette.

Después que el nombre es localizado, el nombre y el dato aparecerán en la pantalla y el programa regresa al menú de la línea 30.

Introduzca "E" en el menú para que termine el programa.

PUT, SAVE

GOSUB/RETURN

| | | |
|-----------------|---|--|
| Función | Produce un salto hacia una subrutina, y una vez que ha terminado esa subrutina regresa a la siguiente línea en donde se originó el salto. | |
| Formatos | GOSUB número de línea GOSUB PROG <i>n</i> RETURN | $1 \leq \text{número línea} < 10000$ $0 \leq n < 10$ (<i>n</i> indica el número de área de programa.) |

El mando GOSUB hace que la ejecución del programa salte a una subrutina en la línea especificada por una variable o una expresión numérica, y el mando RETURN hace que se regrese de la subrutina a la línea siguiente desde donde se hizo el salto.

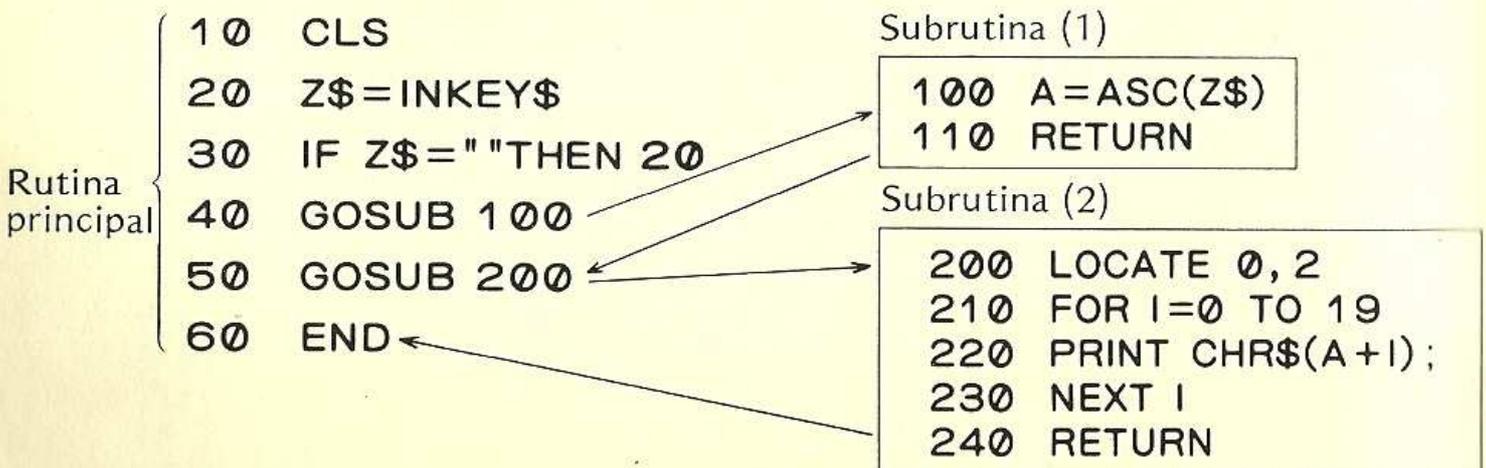
Ejemplo

| | Numero de línea | Area de programa |
|--------------------|-----------------|------------------|
| Constante numérica | GOSUB 200 | GOSUB PROG 4 |
| Variable numérica | GOSUB N | GOSUB PROG L |
| Expresión numérica | GOSUB N*5 | GOSUB PROG L/10 |

La computadora PB-770 permite hacer un salto a una subrutina y regresar a otras áreas del programa, así como también un salto a una subrutina en la misma área de programa, tal y como se muestra en los formatos anteriores.

Usos

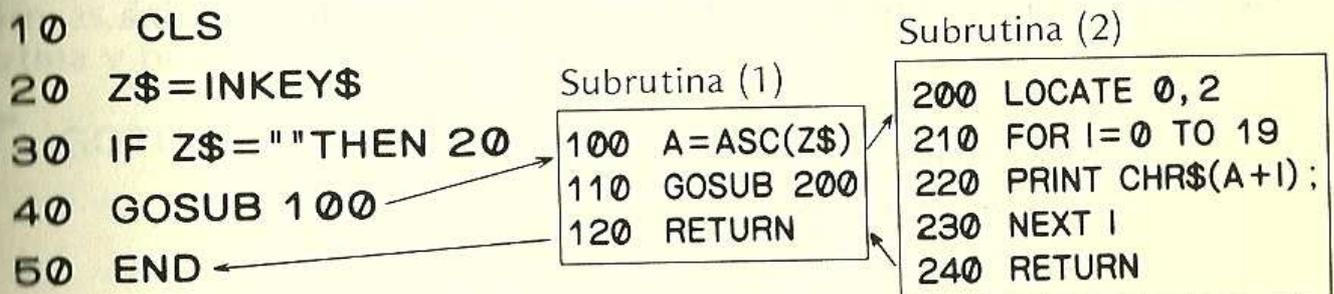
En el siguiente programa, el recorrido de la ejecución se indica por medio de los cambios de la distribución de los grupos de líneas del programa.



Las líneas 10 a 60 representan el área principal del programa, mientras que las líneas 100 a 110 representan una subrutina; la segunda subrutina está formada por las líneas 200 a 240.

Cuando se ejecuta el mando RETURN, la ejecución regresa a la sentencia que sigue al mando GOSUB original y la ejecución continúa.

El programa se puede rediseñar haciendo los pocos cambios siguientes.



Aun cuando el contenido de este programa es exactamente el mismo que el del anterior, este nuevo tiene una doble estructura en donde la subrutina (2) está incluida en la subrutina (1).

Hasta 12 niveles de introducción de subrutinas se pueden realizar con el mando GOSUB/RETURN.

Pero se deben tomar precauciones en relación a los siguientes puntos cuando se use el mando de GOSUB/RETURN.

- (1) Una subrutina debe tener por los menos un mando RETURN, y puede tener tantas como se deseen.
- (2) Si la línea especificada por un mando GOSUB no existe, se producirá un error UL.
- (3) Cuando se ejecute un mando GOSUB PROG n , y si un programa no estuviera introducido en el área n de programa, se producirá un error UL en esa área de programa.
- (4) Cuando no hubiera un mando GOSUB para el mando RETURN, se producirá un error GS.

- (5) Se pueden realizar hasta 12 niveles de introducción de subrutinas. (Si hubiera por ejemplo 13 niveles o más, se producirá un error NO).
- (6) Cuando se incluya una fracción en el número de una línea especificado por el mando GOSUB, o en un área de programa por el mando GOSUB, la ejecución se realiza descartándose la fracción.

PROGRAMA DE EJEMPLO

* He aquí el ejemplo de un programa de un juego de máquina.

```
10 CLS
20 GOSUB 200
30 GOSUB 300
40 GOSUB 400
50 LOCATE 3,0
60 PRINT "POINT";N
70 LOCATE 0,0:END
200 X=INT(10*RND)
210 Y=INT(10*RND)
220 Z=INT(10*RND)
230 RETURN
300 LOCATE 6,2
310 PRINT X;" ";Y;" ";Z;
320 RETURN
400 IF X=Y THEN IF Y=Z THEN N=100:RETURN
410 IF X=Y THEN N=40:RETURN
420 IF Y=Z THEN N=30:RETURN
430 IF X=Z THEN N=20:RETURN
440 N=0
450 RETURN
```

Este programa presenta en la pantalla 3 numerales. Si los tres numerales fueran idénticos, se da 100 de puntuación, pero si solo 2 numerales fueran idénticos, entonces se dará 40, 30, 20 de puntuación dependiendo de la localización dada.

Si los tres numerales fueran diferentes, aparecerá en la pantalla 0.

Las líneas 200 a 230 representan la subrutina que genera los 3 numerales con la función RND, mientras que las líneas 300 a 320 son la subrutina que hace aparecer los numerales en la pantalla. Las líneas 400 a 450 son la subrutina que verifica la puntuación dada.

Las líneas 10 a 70 es la parte principal del programa que controla cada subrutina y presenta en la pantalla los puntos dados.

GOTO

GOTO

| | |
|-----------------|---|
| Función | Produce un salto incondicional a una línea especificada. |
| Formatos | GOTO número de línea $1 \leq \text{número de línea} < 10000$ GOTO PROG <i>n</i> $0 \leq n < 10$ <i>n</i> : Indica el No. de área de programa. |

El mando GOTO hace que la ejecución de un programa salte en forma incondicional a una línea específica o al principio de un área de programa que es especificada con una variable o una expresión numérica. Si el número de línea especificado, o el área de programa especificada no existe, se producirá un error UL. Si estuviera una fracción incluida en el número de línea especificado o en el número de área de programa determinado, la fracción será descartada cuando se ejecute el programa.

Ejemplo

| | Número de línea | Área de programa |
|--------------------|-----------------|------------------|
| Constante numérica | GOTO 500 | GOTO PROG 4 |
| Variable numérica | GOTO N | GOTO PROG N |
| Expresión numérica | GOTO N*5 | GOTO PROG N/10 |

PROGRAMA DE EJEMPLO

* Este es un programa para tener un reloj digital.

```

10 INPUT "H="; H, "M="; M, "S="; S
20 CLS
30 IF H<10 THEN LOCATE 8,0:PRINT H; " ";
   " "; GOTO 50
40 LOCATE 7,0:PRINT H; " "; " ";
50 IF M<10 THEN LOCATE 12,0:PRINT M; "

```

```

        :";:GOTO 70
60 LOCATE 11,0:PRINT M;":";
70 IF S<10 THEN LOCATE 16,0:PRINT S;"
        :";:GOTO 90
80 LOCATE 15,0:PRINT S;
90 FOR I=2 TO 120:NEXT I
100 S=S+1
110 IF S>=60 THEN 200
120 GOTO 70
200 S=0:M=M+1
210 IF M>=60 THEN 300
220 GOTO 50
300 M=0:H=H+1
310 IF H>=24 THEN H=0
320 GOTO 30

```

Cuando se ejecuta este programa, la máquina le pedirá la hora (H), los minutos (M) y los segundos (S). Después de que se ha introducido la hora de ese momento y se ha oprimido la tecla , en la pantalla aparecerá continuamente la hora con los segundos.

Como el reloj interno de la microcomputadora no se usa para este programa, no muestra la hora exacta, pero indica aproximadamente la hora.

Si la hora está adelantada, ajústela incrementando el valor final del mando FOR de la línea 90, y si se atrasa, ajústela reduciendo el valor final.

En este programa se usan mucho mandos GOTO.

El programa produce un circuito infinito usando mandos GOTO en las líneas 120, 220 y 320.

VER GOSUB/RETURN, IF-THEN-ELSE

IF~THEN~ELSE

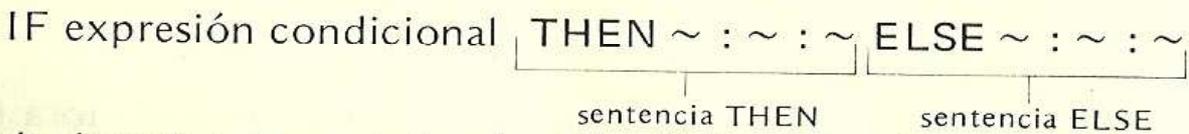
| | |
|-----------------|--|
| Función | Ejecuta un salto de línea condicionado a lo especificado en la parte de IF, dirigiéndose a lo establecido por THEN o ELSE. |
| Formatos | IF expresión condicional THEN { número de línea mando } ELSE { número de línea mando } En este caso no se puede usar una expresión numérica como número de línea. |

El mando IF-THEN realiza un salto condicionado mientras que el mando GOTO lo realiza en forma incondicional.

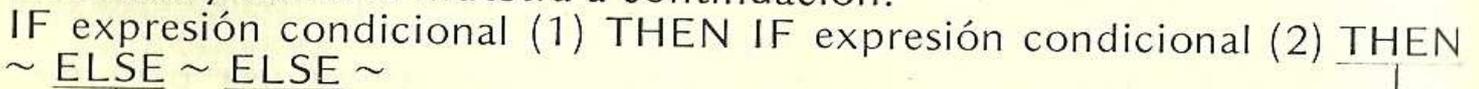
Cuando la expresión condicional es verdadera, la sentencia THEN es ejecutada, pero cuando no lo es, entonces la sentencia ELSE es ejecutada. Si no hubiera una sentencia de ELSE se ejecutará la línea siguiente.

Un número de línea que representa el destino de un salto, o un mando se pueden introducir en las sentencias THEN y ELSE.

Además se pueden realizar sentencias multiples en las sentencias THEN o ELSE usando para ello dos puntos, tal y como se muestra a continuación.



Se puede insertar una sentencia IF adicional en la parte de THEN o ELSE tal y como se muestra a continuación.



Se ejecuta cuando la expresión condicional (1) es verdadera y cuando la expresión condicional (2) es falsa.

Se ejecuta cuando la expresión condicional (1) es falsa.

Se ejecuta cuando las expresiones condicionales (1) y (2) son verdaderas.

Usos

A continuación se muestran ejemplos del uso de las sentencias IF-THEN-ELSE.

Los ejemplos verifican si un triángulo se puede hacer o no despues que se han introducido tres lados.

```
(1)      10 INPUT "A=" ; A, "B=" ; B, "C=" ; C
          20 IF A+B>C THEN 40
          30 GOTO 50
          40 IF ABS(A-B)<C THEN 60
          50 PRINT "NOT TRI" : GOTO 10
          60 PRINT "TRI" : GOTO 10
```

```
(2) 10 INPUT "A=";A, "B=";B, "C=";C
    20 IF A+B>C THEN IF ABS(A-B)<C THEN 40
    30 PRINT "NOT TRI" : GOTO 10
    40 PRINT "TRI" : GOTO 10
```

```
(3) 10 INPUT "A=";A, "B=";B, "C=";C
    20 IF A+B>C THEN IF ABS(A-B)<C THEN 40
    30 PRINT "NOT TRI" : GOTO 10
    40 IF A=B THEN IF B=C THEN PRINT "E.TRI" : GOT
        O 10 ELSE PRINT "I.TRI" : GOTO 10
    50 IF B=C THEN PRINT "I.TRI" : GOTO 10
    60 IF A=C THEN PRINT "I.TRI" : GOTO 10 ELSE
        PRINT "TRI" : GOTO 10
```

Los programas de ejemplo (1) y (2) verifican si se puede formar triángulo (TRIángulo) o no (NOT TRIángulo), habiendo indtroducido los tres lados A, B, y C.

En el ejemplo 1, se verifican dos condiciones de triángulo ($A+B>C$, $|A-B|<C$) en líneas separadas (líneas 20 y 40); mientras que en el ejemplo (2) son verificadas en una sola (línea 20).

En el ejemplo 3, el programa verifica si se trata de un triángulo equilátero (Equilátero TRIángulo) o es un isósceles (Isósceles TRIángulo), así como también si es un triángulo (TRI) o no lo es (NOT TRI).

En el ejemplo (3) se verifica primero si se forma un triángulo (TRI) o no en la línea 20. Si no se forma un triángulo, aparecerá en la pantalla la expresión "NOT TRI" y se regresará a la línea 10.

Si se forma un triángulo, el que sus lados fueran iguales o no, es verificado en la línea 40. Si los tres lados fueran iguales, aparecerá en al pantalla la expresión "E. TRI". Si sólo dos lados fueran iguales, en ese caso aparecerá en la pantalla "I. TRI"; y finalmente si los tres lados fueran desiguales simplemente aparecerá "TRI", operación que se realiza por las líneas 40 a 60, para regresar a la línea 10.

PROGRAMA DE EJEMPLO

* Este es un programa para trazar un patrón con puntos en la pantalla.

```

10 CLS
20 X=0:Y=0:N=1:M=1
30 DRAW(X,Y)
40 X=X+N:Y=Y+M
50 IF X>=158 THEN N=-1
60 IF Y>=30 THEN M=-1
70 IF X<=0 THEN N=1
80 IF Y<=0 THEN M=1
90 IF X=1 THEN IF Y=31 THEN BEEP 1:END
100 GOTO 30

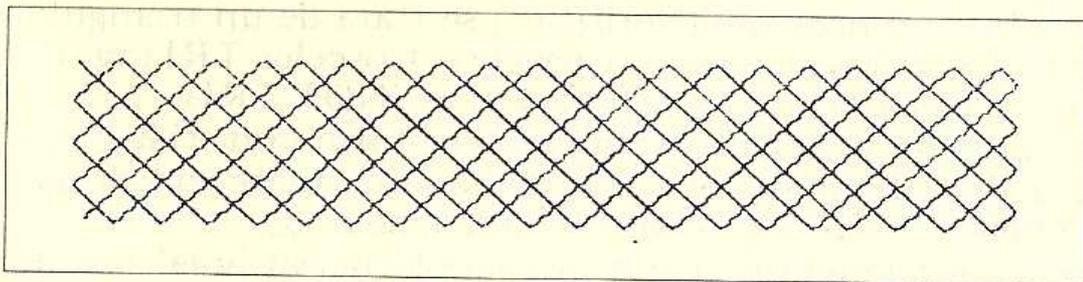
```

Este programa determina si el valor de las coordenadas (X, Y) formadas por puntos en la pantalla está dentro de los límites de la pantalla, haciéndolo por medio del mando IF-THEN (líneas 50 a 80) para controlar si los valores de X, Y se incrementan o se reducen.

Cuando los valores de X y de Y son: $X = 1$, $Y = 31$ (línea 90), se genera un sonido y el programa termina.

A continuación se muestra el resultado de la ejecución del programa.

■ Ejemplo de la ejecución



INPUT

| | |
|-----------------|---|
| Función | Pide la introducción de datos (valor numérico, carácter) para una variable mediante el teclado. |
| Formatos | INPUT variable [, variable] INPUT "texto de aclaración", variable [, "texto de aclaración", variable] INPUT "texto de aclaración"; variable [, "texto de aclaración"; variable] |

El mando INPUT se usa para introducir datos para una variable desde el teclado.

Los formatos básicos de la sentencia INPUT son los siguientes:

| | |
|-----------|------------------------|
| Ejemplo 1 | INPUT A |
| Ejemplo 2 | INPUT X, Y, Z |
| Ejemplo 3 | INPUT "AGE", A |
| Ejemplo 4 | INPUT "NAME"; A\$ |
| Ejemplo 5 | INPUT "X="; X, "Y="; Y |

Cuando el mando INPUT se ejecuta, la PB-770 exhibe en la pantalla un mensaje pidiendo la introducción de datos, y espera a que se introduzcan.

Por ejemplo, cuando el ejemplo 1 se ejecuta, "?" aparece en la pantalla y el cursor parpadea a la derecha de "?". La demanda para la introducción de datos se termina.

Exhibición durante la ejecución del mando INPUT.

Ready P0

? _ (Cursor)

La introducción de datos se hace oprimiendo teclas. Una vez que los haya introducido, debe siempre oprimir la tecla **ENTER** o . Tome en cuenta que la tecla **ENTER** funciona igual que la tecla  durante la ejecución de la sentencia INPUT.

- Las variables que se pueden usar en una sentencia INPUT son las siguientes:

[Ejemplos]

- Variable numérica INPUT X
- Variable de caracteres INPUT X\$ (se pueden introducir hasta 7 caracteres).
- Variable registrada. INPUT XY
INPUT XY\$ (Se pueden introducir hasta 16 caracteres).
- Variable matricial INPUT X! (i), Y! (i, j)
Matriz (array) numérica de media precisión
INPUT X (i), Y (i, j)
Matruz numérica de precisión simple
INPUT A\$ (i), INPUT A\$ (i, j)
Matruz de caracteres

Usos

(1) Entrada de valores numéricos

Revisemos el uso de la sentencia INPUT y sus funciones usando un programa sencillo.

- 10 INPUT A Mando para introducir. Le adjudica los datos introducidos a la variable A.
- 20 PRINT A Mando para dar salida la información. Exhibe en la pantalla el contenido de la variable A.
- 30 GOTO 10 Mando para saltar. Desplaza la ejecución del programa a la línea 10.

Después de introducir este programa, introduzca RUN y oprima  o   para ejecutarlo. Entonces “?” aparecerá en la pantalla. Ahora intoduzca 3.6 y oprima  . Si lo hace correctamente, los mismos números aparecen en la pantalla por la acción de la sentencia PRINT, de la siguiente manera:

 RUN  3.6 

```

RUN
?3.6
 3.6
? _
    
```

Un valor puede entrarse mediante la entrada de una expresión de cálculo que dé como resultado dicho valor. Ello se limita a las sentencias INPUT que utilizan variables numéricas.

CLS RUN $100-20/5$

```
RUN
?100-20/5
  96
?-
```

(2) Entrada de caracteres

Introduzca caracteres modificando el programa del inciso (1). El programa será el siguiente:

```
10 INPUT A$
20 PRINT A$
30 GOTO 10
```

Cuando se introducen caracteres, se debe usar una variable de caracteres como la indicada en el programa anterior. Cuando ejecute este programa, se le pedirá que introduzca los datos mediante la exhibición en la pantalla de "?", de la misma manera que para la introducción de valores numéricos. Cuando introduzca la serie de caracteres ABC, aparecerán en la pantalla de la siguiente manera:

CLS RUN ABC

```
RUN
?ABC
ABC
?_
```

Cuando se introduzca el valor numérico 123, aparecerá en la pantalla 123; pero en este caso 123 representa en realidad una serie de caracteres y no un valor numérico.

Siempre que introduzca datos en una variable de caracteres, no será necesario entrecomillar la serie de caracteres con ". Si las usa, los datos serán introducidos como datos de caracteres.

■ **Datos que se pueden introducir en cada variable mediante una sentencia de INPUT**

A. Variables numéricas

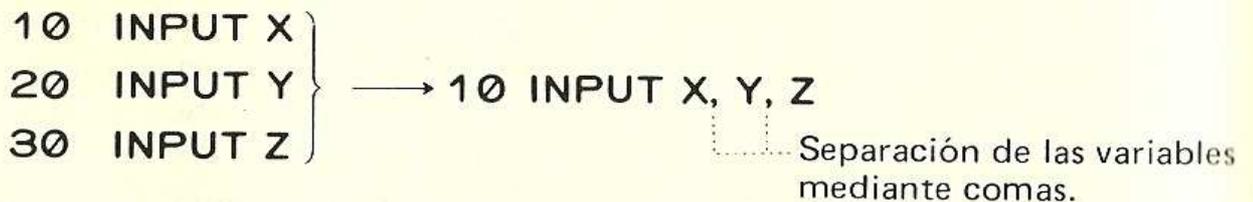
- a. $\pm 1 \times 10^{-99}$ a $\pm 9.999999999999 \times 10^{99}$ y 0
- b. Expresiones de operaciones para un valor numérico (ejemplo: $200 \times (5 + 2)$).
- c. Variable numérica de A a Z (variable fija).
- d. Variable registrada
- e. Variable matricial

B. Variables de caracteres

- a. Variable de caracteres fija Hasta 7 caracteres y símbolos.
- b. Variable de caracteres registrada Hasta 16 caracteres y símbolos.
- c. Variable matricial Hasta 79 caracteres y símbolos.

(3) Entrada con más de dos variables . . . Ejemplo 2

Se pueden introducir más de dos variables en una sentencia INPUT. Más de dos sentencias INPUT se pueden colocar en una sola sentencia, tal y como se ilustra a continuación:



Cuando ejecute esta sentencia INPUT, “?” aparecerá en la pantalla para pedirle primero el valor de X. Una vez que haya introducido el valor de X, se le pedirán los valores de Y y Z, uno por uno. Una vez que haya introducido el valor de Z, esta sentencia INPUT terminará.

Se pueden usar variables, tales como variables numéricas, de caracteres, etc... en todas las combinaciones y secuencias deseadas, con esta sentencia INPUT, tal y como se ilustra a continuación:

```
10 INPUT A$, X
```

No olvide que debe usar “,” (una coma) para separar las variables.

(4) Sentencia INPUT que exhibe un mensaje

Si se introduce una serie de caracteres entrecomillada, con “ ”, entre la sentencia INPUT y la variable, esta serie de caracteres aparecerá en la pantalla tal y como esté. Esto se llama un texto de explicación. La introducción incorrecta de datos se puede reducir aclarando lo que se pide mediante un mensaje.

Entre la sentencia INPUT a continuación y ejecutela:

```
10 INPUT "AGE" ; A
```

Entonces lo siguiente aparecerá en la pantalla:

CLS RUN 

RUN
AGE? _

Ahora, la sentencia INPUT se cambia de la siguiente manera:

10 INPUT "AGE", A

Cuando ejecute este programa, lo siguiente aparecerá en la pantalla:

CLS RUN ↵

RUN
AGE_

Para que "?" aparezca después del mensaje, es necesario usar ";" entre el mensaje y la variable.

Esta sentencia INPUT se puede cambiar para introducir dos o más variables.

10 INPUT "HEIGHT=" ; X, "WEIGHT=" ; Y

Podemos considerar que las siguientes dos sentencias INPUT se colocaron en una sola sentencia INPUT:

10 INPUT "HEIGHT=" ; X }
20 INPUT "WEIGHT=" ; Y } → 10 INPUT "HEIGHT=" ;
X, "WEIGHT=" ; Y

Cuando ejecute este programa, lo siguiente aparecerá en la pantalla:

CLS RUN ↵

1 7 5 ↵

6 5

RUN
HEIGHT=?175
WEIGHT=?65_

■ **Número de caracteres en una serie de caracteres usada para un mensaje**

El máximo número de caracteres que se pueden usar para una serie de caracteres entrecomillada, con " ", es de 79 caracteres incluyendo los números de la línea y el mando INPUT.

LET

| | |
|----------------|-----------------------------|
| Función | Asigna datos a una variable |
| Formato | LET variable = expresión |

El mando LET colocado al principio de una sentencia de asignación se usa generalmente en los siguientes formatos:

(Ejemplo 1) LET A=10 LET A\$="GAME"

(Ejemplo 2) LET X=SIN (S-PI/4) LET X\$=A\$+B\$

La orden de asignación le asigna el valor de la expresión que está a la derecha del signo = a la variable que está a la izquierda del signo =. Una expresión numérica debe corresponder a una variable numérica, y una expresión de caracteres debe corresponder a una variable de caracteres. Si la correspondencia no es correcta, aparecerá en la pantalla un error TM.

■ Gama para valores numéricos

Podrán asignarse a una variable numérica cualquier valor dentro de los siguientes límites:

$$-10^{100} < \text{valor numérico} < 10^{100}$$

■ Gama para secuencia de caracteres

Cuando el lado izquierdo del = es una variable de caracteres fija — hasta 7 caracteres.

Cuando el lado izquierdo del = es una variable de caracteres registrada — hasta 16 caracteres.

Cuando el lado izquierdo del = es una variable matricial de caracteres — hasta 79 caracteres.

■ LET puede omitirse.

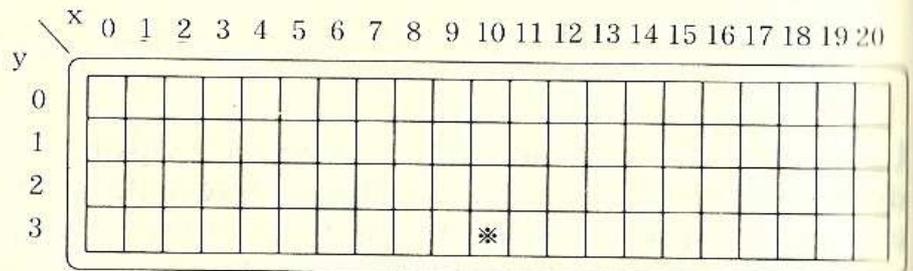
10 LET A=1 es igual que 10 A=1

LOCATE

| | |
|----------------|--|
| Función | Especifica la posición del cursor |
| Formato | LOCATE X, Y $0 \leq X < 20$, $0 \leq Y < 4$ |

La pantalla de exhibición de la PB-770 está provista con 20 x 4 posiciones de exhibición, tal y como se ilustra en seguida. La exhibición se hace generalmente a partir del extremo izquierdo de la pantalla mediante la ejecución de la sentencia PRINT. Sin embargo, la posición de exhibición se puede cambiar libremente usando el mando LOCATE.

Por ejemplo, si usted especifica LOCATE 10, 3, la posición de presentación se especificará en (10, 3).



Sin embargo, cuando para la coordenada Y se especifica 3, 10 la pantalla se desplaza hacia arriba. Para evitar este desplazamiento, excepto para LOCATE 19, 3 (posición inferior derecha), use un punto y coma al final de la sentencia PRINT.

Usos

- Lista A
- 10 X= 1
 - 20 X=X+ 1
 - 40 PRINT "X=" ; X
 - 50 GOTO 20

- Lista B
- 10 X= 1
 - 20 X=X+ 1
 - 30 LOCATE 0, 0
 - 40 PRINT "X=" ; X
 - 50 GOTO 20

Cuando usted ejecute la Lista A, lo siguiente aparecerá en la pantalla:

| | |
|------|--|
| X= 2 | Los números aparecen en forma continua a partir de la parte inferior de la pantalla y desaparecen hacia la parte superior de la pantalla. |
| X= 3 | |
| X= 4 | |
| X= 5 | |

Cuando usted agregue la línea 30, como se indica en la Lista B, la exhibición en la pantalla cambiará de tal manera que "X = Número" aumentará gradualmente en el extremo superior izquierdo de la pantalla.

POKE

| | |
|----------------|---|
| Función | Escribe un dato en la dirección especificada. |
| Formato | POKE dirección, dato |

El mando POKE escribe un dato en la dirección de memoria especificada. Tanto la dirección como el dato se especifican mediante números enteros (se descarta todo decimal).

Las direcciones y datos deben estar dentro de los siguientes límites.

$$-32769 < \text{dirección} < 65536$$

$$0 \leq \text{dato} < 256$$

Para mayor información sobre las direcciones, remítase a la parte destinada al mando CLEAR.

PRECAUCION:

Nunca almacene datos en el área del sistema (&H0000 hasta &H0528) ni en el área de memoria donde se encuentran almacenados los programas. En particular, si se almacena cualquier dato en el área del sistema por medio de POKE, la computadora dejará de funcionar correctamente. En dicho caso, apáguela una vez, vuélvala a encender y entre NEW ALL  , para borrar todos los programas y el contenido de las variables.

PROGRAMA DE EJEMPLO

Escriba datos en la dirección especificada usando el siguiente programa.

```

10 REM POKE EXAMPLE
20 INPUT "-32769<ADDRESS<65536", A
30 INPUT "0<=DATA<256", D
40 POKE A, D
50 PRINT PEEK A

```

 CLEAR, PEEK

PRINT/LPRINT

| | | |
|------------------|-----------|---|
| Funciones | | PRINT: Obtención de datos en la pantalla. LPRINT: Obtención de datos en el impresor. |
| Formatos | Pantalla | PRINT expresión (, expresión) PRINT expresión (; expresión) PRINT \$ variable de careteres registrada |
| | Impresora | LPRINT expresión (, expresión) LPRINT expresión (; expresión) |

Los mandos PRINT y LPRINT son prácticamente iguales, con la única diferencia de que los datos aparecen en la pantalla o en el impresor. Sin embargo, cuando se usan con la función TAB, ocurren algunas diferencias.

Usos

Usando una sentencia PRINT, se pueden exhibir en la pantalla diferentes clases de datos, tales como caracteres, expresiones numéricas, así como valores numéricos.

Por ejemplo, se puede usar de la siguiente manera:

```
PRINT 1.414
PRINT A * B - 2
```

Como A y B son variables, se obtiene y exhibe el resultado del cálculo.

Cuando estas sentencias PRINT se ejecutan, se realiza un cambio de línea después de que los datos se exhiben en la pantalla. Ejecute el siguiente programa como ejemplo:

```
10 A = 0 : B = 3
20 PRINT 1.414
30 PRINT A * B - 2
40 INPUT C
```

Entonces lo siguiente aparecerá en la pantalla:

CLS RUN

```

RUN
  1.414
  -2
  ? ..... Cursor
    
```

Una sentencia PRINT puede incluir varias expresiones o series de caracteres separadas mediante una coma (,).

```
10 A=0: B=3
20 PRINT 1.414, A*B-2
30 INPUT C
```

Cuando ejecute este programa, se realizará un cambio de línea después de que los datos se exhiban en la pantalla, de la misma manera que en el caso precedente.

```
RUN
 1.414
-2
?
..... Cursor
```

Si bien la pantalla de exhibición de la PB-770 solamente cuenta con 4 líneas, después de exhibirse los datos en la 4^a línea, las líneas se mueven como si se desplazaran hacia arriba.

```
10 PRINT 1, 2, 3, 4
20 END
```

```
4
Ready P0
-
      RUN
      1
      2
      3
      4 } Movimiento hacia arriba
```

Si bien la exhibición de los datos en la pantalla se realiza con un cambio de línea al usar una coma, si varias expresiones y series de caracteres se separan con punto y coma (;), aparecerán en la misma línea de la pantalla, de la manera que se ilustra a continuación:

```
10 A=0: B=0
20 PRINT 1.414; A*B-2
30 END
```

```
RUN
 1.414-2
Ready P0
-
```

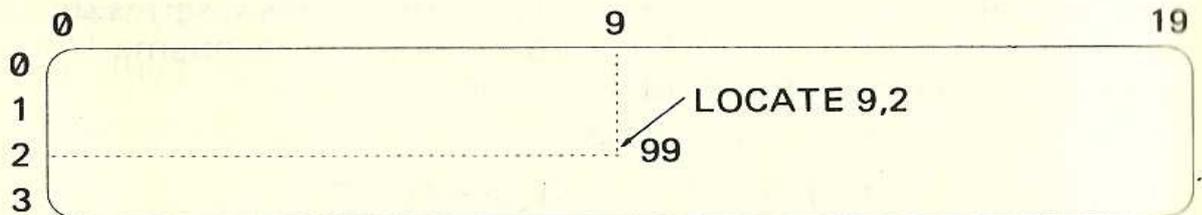
Es posible introducir un mensaje fácil de leer usando el siguiente método:

```
10 PRINT "ANSWER=" ; A*B-2
20 END
```

```
RUN
ANSWER=-2
Ready P0
-
```

Los números y caracteres se exhiben o imprimen alineados con el margen izquierdo, tal y como se ilustra en los ejemplos anteriores. Sin embargo, en el caso de los valores numéricos, como se incluye una posición para el signo, si el valor es positivo, se dejará un espacio aunque se omita el signo +. Con la sentencia PRINT se puede usar el mando LOCATE y funciones tales como TAB y USING, con el objeto de especificar el formato y la localización en la pantalla de los datos.

```
10 CLS
20 A=1
30 LOCATE 9, 2
40 PRINT A
50 A=A+1
60 IF A<100 THEN 30
70 END
```

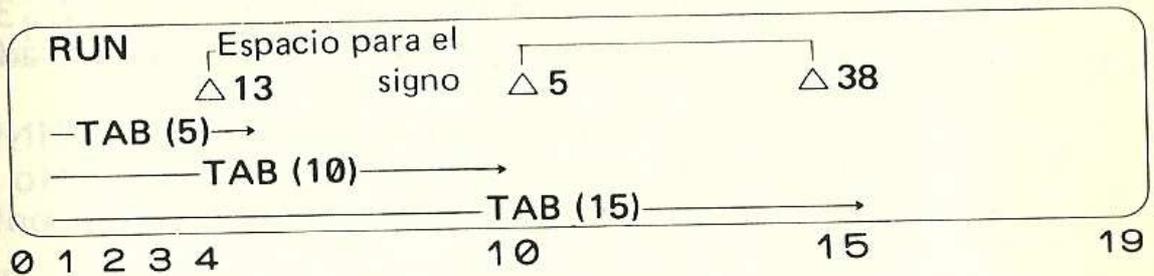


Los valores numéricos 1~99 son exhibidos en orden en la ubicación de las coordenadas (10, 2) y (11, 2) de la pantalla, tal y como se ilustra. El espacio para el signo aparece en el punto de la coordenada (9,2).

```

10 CLS
20 A=13 : B=5 : C=38
30 PRINT TAB(5) ; A ; TAB(10) ; B ; TAB(15) ; C
40 END

```



La exhibición en la pantalla se realiza empezando a partir de la ubicación especificada mediante la función TAB, cuando se usa una sentencia PRINT junto con la función TAB.

La exhibición se puede realizar con un formato uniforme de acuerdo con el formato de la función USING, tal y como se ilustra a continuación:

```

10 A=3.1415 : B=31.415 : C=314.15
20 PRINT USING "###.##" ; A
30 PRINT USING "###.##" ; B
40 PRINT USING "###.##" ; C
50 FOR I=1 TO 1000 : NEXT I
60 END

```

```

  3.14
 31.42
314.15 ..... Redondeo a 2 decimales

```

* Funciones especiales del mando PRINT (no pueden usarse con LPRINT)

(1) Definición del patrón de presentación en pantalla

El patrón de visualización puede definirse colocando el signo "\$" antes de la variable de caracteres registrada (por ej.: \$AB\$). Para ello, deben asignarse valores hexadecimales (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, ó E) a la variable de caracteres registrada.

(2) Códigos de control

CLS y TAB pueden usarse junto con el mando PRINT. Para usar los códigos de control, se utiliza la función CHR\$ junto con PRINT. El código de control debe ir entre paréntesis a continuación la función CHR\$.

| Código | Función |
|--------|--|
| &H02 | LINE TOP (cursor al principio de la línea) |
| &H06 | LINE END (cursor al final de la línea) |
| &H07 | BEL (igual que BEEP 1) |
| &H09 | TAB (igual que TAB (10)) |
| &H0B | HOME (cursor a la coordenada (0,0)) |
| &H0C | CLS (igual que CLS) |
| &H0D | RETURN (avance a la línea siguiente) |
| &H1C | → (cursor un espacio hacia la derecha) |
| &H1D | ← (cursor un espacio hacia la izquierda) |
| &H1E | ↑ (cursor una línea hacia arriba) |
| &H1F | ↓ (cursor una línea hacia abajo) |

VER LOCATE, TAB, USING

PUT

| | |
|-----------------|--|
| Función | Graba datos de variables en cinta de cassette. |
| Formatos | PUT variable [, variable] PUT "nombre de archivo" variable [, variable] |

El mando PUT graba los datos de las variables en una cinta de cassette. El formato del archivo de los datos que se van a grabar es ASCII. El mando GET se usa para extraer el archivo de datos.

- (1) PUT A [, B, C]
- (2) PUT "SALES" A [, B, C]

Como se muestra en el formato anterior (1), el nombre del archivo se puede omitir. En este caso, el nombre del archivo tampoco deberá especificarse en el mando GET correspondiente. Es posible grabar dos o más variables en un archivo de datos usando un solo mando PUT, siempre y cuando éstas se separen con una coma (,). Las variables introducidas en el mando PUT se van grabando en el orden en que han sido introducidas. Los datos numéricos se graban de la misma manera como si se presentaran en la pantalla sin usar la función USING.

Usos

El siguiente programa graba el contenido de las variables matriciales de A (0) a A (10) mediante un mando PUT. Se supone que estas variables contienen datos.

```

10 REM PUT
20 DIM A(10)
30 FOR K=1 TO 10
40 A(K)=K
50 PUT A(K)
60 NEXT K
70 END

```

El siguiente programa extrae los datos almacenados en una cinta de cassette y los transmite a la computadora PB-770 mediante un mando GET.

```
10 REM GET
20 DIM B(10)
30 FOR K=1 TO 10
40 GET B(K)
50 PRINT "B(";K;")=";B(K)
60 NEXT K
70 END
```

En este ejemplo, se usa un lazo FOR-NEXT para mover el contenido de la lista A a la B. Cuando vaya a almacenar varios datos, ponga atención a la secuencia en la cual se vayan a almacenar.

VER GET, SAVE

READ/DATA/RESTORE

| | |
|------------------|---|
| Funciones | <p>READ: Extrae datos de una sentencia DATA y los asigna a una variable.</p> <p>DATA: Almacena datos (constantes, caracteres) en un programa que se va a extraer con una sentencia READ.</p> <p>RESTORE: Especifica la línea a partir de la cual se extraen datos de sentencias DATA.</p> |
| Formatos | <p>READ variable [, variable]</p> <p>DATA datos [, datos][, "datos de caracteres"]</p> <p>RESTORE</p> <p>RESTORE número de línea ($1 \leq \text{número de línea} < 10000$)</p> |

Una sentencia READ debe usarse con una sentencia DATA. Cuando una sentencia READ se ejecuta, los datos son extraídos de una sentencia DATA y son asignados a variables, cada dato a una variable.

Usos

Este es el ejemplo más sencillo de un programa usando las sentencias READ/DATA.

```

10 READ A
20 READ B$
30 PRINT A;B$
40 DATA 7,"B"
50 END

```

Cuando usted ejecuta este programa, 7 es asignado a la variable A, y "B" es asignado a la variable de caracteres B\$. El tipo de variable y de dato deben coincidir.

Es posible introducir en una sola sentencia READ cuantas variables desee; por lo tanto, las líneas 10 y 20 se pueden escribir en una sola línea de la siguiente manera:

```

10 READ A, B$

```

No hay diferencia si los datos de caracteres se ponen o no entre comillas, tal como se muestra a continuación.

40 DATA 7, B

Sin embargo, si el data no se escribe entre comillas, cuando introduzca un espacio, éste se ignorará como dato. Por lo tanto, si se requieren espacios, éstos deberán entrecomillarse.

DATA 7, " ", " A", C Este espacio se ignora.

NOTA: Los espacios a continuación del dato no se ignoran. Por lo tanto si se coloca un espacio a continuación de un valor numérico, aparece un error ST.

DATA 1, 2 , 3, 4 Estos espacios no se ignoran.

No se pueden introducir comillas ni comas en datos de caracteres, además de en el formato anterior.

Las sentencias DATA que no contienen información alguna se consideran como una serie sin caracteres.

DATA , , es igual a DATA " ", " " "

Si bien las variables en una sentencia READ deben corresponder a los datos de DATA en una relación de uno a uno, no hay límite en cuanto al número de variables o de datos que se puedan introducir en cada sentencia.

```

10 CLEAR
20 DIM C(10)
30 READ A, B
40 FOR I=1 TO 10
50 READ C(I)
60 NEXT I
70 DATA 1,2,3
80 DATA 4,5,6,7,8,9
90 DATA 10,11,12
    
```

Cuando usted ejecuta este programa, los datos son asignados a cada variable de la siguiente manera:

| | | | | | | | |
|---|---|------|------|-----------|------|------|-------|
| A | B | C(1) | C(2) | C(3)..... | C(8) | C(9) | C(10) |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| 1 | 2 | 3 | 4 | 5..... | 10 | 11 | 12 |

Si el número de datos es menor que el número de variables a las cuales se van a asignar los datos mediante una sentencia READ, ocurrirá un error DA. Sin embargo, si el número de datos es mayor que el número de variables, no ocurrirá un error, pero los datos excedentes serán ignorados. Una sentencia DATA se puede introducir antes que una sentencia READ. La sentencia DATA de la cual se extraerá un dato mediante una sentencia READ, puede ser especificada usando una sentencia RESTORE.

La sentencia RESTORE tiene dos formatos diferentes, uno en el cual se escribe un número de línea y otro en el cual no se escribe. Si no se escribe el número de línea, cuando RESTORE se ejecuta la siguiente sentencia READ extrae los datos de la primera sentencia DATA.

```

10 READ A, B
20 RESTORE
30 READ C, D
40 PRINT A; B; C; D
50 DATA 7, 2
60 END

```

Cuando usted ejecuta este programa, la asignación será: A = 7, B = 2, C = 7 y D = 2.

También se pueden especificar sentencias DATA incluyendo el número de línea.

```

10 RESTORE 50
20 READ A, B
30 PRINT A; B
40 DATA 3.7, 6.5
50 DATA 7.1, 9.3 ..... Datos especificados mediante la
60 DATA 5, 10.2          sentencia RESTORE en la línea 10.
70 END

```

Cuando usted ejecuta este programa, la asignación será: $A = 7.1$ y $B = 9.3$.

Se pueden usar variables y expresiones numéricas como especificación de un número de línea de una sentencia **RESTORE** pero la variable o expresión numérica utilizada debe corresponder a un número de línea que tenga una sentencia **DATA**.

Se debe tener cuidado cuando se use una sentencia **READ** en un programa que se vaya a mover a varias áreas de un programa.

| | | | | | |
|-----------|-----------|---------------------|-----------|-----------|-------------------|
| P0 | 10 | READ A,B | P1 | 10 | READ X, Y |
| | 20 | GOTO PROG 1 | | 20 | PRINT X;Y |
| | 30 | DATA 1,2,3,4 | | 30 | DATA 71,65 |
| | | | | 40 | END |

Cuando usted ejecuta este programa, los datos que aparecerán no serán 71 y 65, sino 3 y 4. En otras palabras, aunque la ejecución del programa se haya cambiado al área **P1** mediante **GOTO PROG 1**, la sentencia **DATA** en **P0** se continúa usando.

Esto es útil cuando la sentencia **DATA** de un programa principal se va a usar en una subrutina.

Si es necesario asignar 71 y 65 a **X** y **Y**, respectivamente, en este programa, especifique el número de la línea al principio del programa **P1** de la siguiente manera:

5 RESTORE 30

PRECAUCION:

Asegúrese de no escribir sentencias **DATA** en las líneas 2200 hasta 2299 y en aquellas líneas que terminan con el número 22 (por ej.: 22, 322, 922). De lo contrario, tales sentencias **DATA** serán ignoradas.

REM

| | |
|----------------|---|
| Función | Permite la inclusión de comentarios en un programa. |
| Formato | REM comentario |

A diferencia de otros mandos, el mando REM no ejecuta nada. Se puede introducir cualquier comentario libremente a continuación de REM; por tanto se puede dar la explicación de un programa en puntos importantes tal y como se muestra a continuación, de tal manera que cada parte del programa pueda entenderse con sólo ver la lista.

```

10  REM  *PROCESSING RESULTS*
20  DIM  A (100)
      .
      .
100  REM  *PRINT OUT*
      .
      .
500  END

```

Todos los caracteres y símbolos introducidos a continuación de REM se consideran como comentarios; por lo tanto, no se pueden introducir otros mandos después, como en las llamadas sentencias múltiples.

Ejemplo: 150 REM *TOTAL *: INPUT N
└──────────┘
No se ejecuta

STOP

| | |
|----------------|-------------------------------------|
| Función | Suspende la ejecución del programa. |
| Formato | STOP |

Si durante la ejecución de un programa el flujo encuentra un mando STOP, aparecerá en la pantalla un mensaje anunciando que se ha interrumpido la ejecución del programa en esa línea. La ejecución de un programa detenido por STOP, puede reanudarse a partir de la siguiente línea introduciendo un mando CONT.

Usos

Con el siguiente ejemplo se verifica el mando STOP.

```

10 A=1:B=5
20 C=A+B
30 STOP
40 PRINT C
50 END

```

Cuando se ejecuta este programa, aparecerá lo siguiente en la pantalla:
STOP P0-30

Esto indica que la ejecución se ha suspendido por el mando STOP de la línea 30 del área del programa P0.

En esta situación se puede verificar el contenido de la variable, de la siguiente forma:

```

A [ENTER] → 1      Exhíbe el valor de A
C [ENTER] → 6      Exhíbe el valor de C

```

Además, se puede asignar un valor opcional a la variable introduciendo

```
C = 0 ↵
```

En la práctica, este mando se utiliza para interrumpir la ejecución de un programa donde se desea, con fines de corrección de errores en el programa, verificar el contenido de las variables utilizadas.

1) Se puede reanudar la ejecución usando la orden CONT aun si la operación subsiguiente se realiza mientras está un efecto la orden STOP.

- (1) Cálculo manual
- (2) Asignación a una variable (usando LET)
- (3) Confirmación del contenido de la variable.
- (4) Ejecución de los siguientes mandos:

ANGLE, BEEP, CLEAR, CLS, DIM, ERASE, PRINT, LPRINT, TRON, TROFF

2) Si las siguientes operaciones se ejecutan mientras está en efecto un mando STOP, la reanudación por medio del mando CONT no será posible.

- (1) La ejecución de los mandos manuales (EDIT, SAVE, LOAD, LIST, etc.).
- (2) Ejecución de PUT/GET
- (3) Cuando se produce un error.

 CONT

TRON/TROFF

| | |
|-----------------|--|
| Función | Rastrea la ejecución del programa y termina esta operación de rastreo. |
| Formatos | TRON TROFF |

Los mandos TRON y de TROFF se usan durante el proceso de eliminación de pasos ilógicos.

- TRON..... Especifica el modo de rastreo.
- TROFF..... Libera la modalidad de rastreo.

Cuando se ha especificado la modalidad de rastreo, se ejecuta un programa mientras aparece en al pantalla el número de área del programa y el número de línea, de la siguiente forma:

En pantalla [0 : 1 1 0]

Número de línea que se ejecuta.
 Area de programa que se ejecuta.

Debido a que TRON y TROFF son mandos de programa, se pueden usar introduciéndolos en un programa, sin embargo son usadas generalmente introduciéndolos directamente.

Usos

Introduzca el siguiente programa y ejecútelo.

```

P0
10 BEEP 1
20 GOTO PROG 1

P1
10 BEEP 0
20 GOTO PROG 0
    
```

Cuando se hace ejecutar este programa, se producirán dos sonidos en forma alternada. Pulse la tecla **BRK** y luego introduzca **TRON** para especificar la modalidad de rastreo. Ejecute ahora el programa. Ahora el área del programa y el número de línea que ahora se ejecutan aparecen continuamente en la pantalla, como se muestra a continuación.

| | |
|------------|------------|
| { 0 : 10 } | { 0 : 20 } |
| { 1 : 10 } | { 1 : 20 } |
| { 0 : 10 } | { 0 : 20 } |

Podrá notarse que el intervalo entre los sonidos del mando BEEP es más prolongado en el modo de rastreo. Ello se debe a que en este modo la ejecución es más lenta.

Además, durante la ejecución de la sentencia INPUT, se detiene y coloca en la pantalla el signo de interrogación, "?" después de [número de área, número de línea.] El resultado puede verse en la pantalla durante la ejecución de la sentencia de PRINT. Este mando es muy conveniente durante el proceso de eliminación de pasos ilógicos debido a que el proceso de ejecución se puede rastrear.

4-3 FUNCIONES NUMERICAS

SIN

| | |
|----------------|--|
| Función | Determina el seno de X (SIN X). |
| Formato | SIN expresión numérica $-5400^\circ < \text{expresión numérica} < 5400^\circ$ |

La función SIN sirve para determinar el seno del ángulo X, el cual puede expresarse en tres diferentes unidades angulares a elegirse (DEG, RAD ó GRA). Al encenderse la computadora, ésta selecciona automáticamente la unidad DEG (grados).

Ejemplo

Cálculo de SIN (X) en grados (DEG).

```
5 ANGLE 0
10 PRINT SIN30
20 PRINT SIN45
30 PRINT SIN90
50 END
```

Al ejecutarse este programa (**R U N** ), aparecen consecutivamente los senos de 30, 45 y 90 grados.

Ejecute este programa agregando antes de ello la siguiente línea.

```
25 STOP
```

En este caso, se visualiza un resultado y se detiene el programa.

Ejemplo de ejecución

```
0.5
0.7071067812
STOP P0-25
-
```

Para reanudarlo, presione **CONT**  ó **SHIFT** **CONT**  . Así se visualizará el resultado de SIN90 (ver la página 129)

Ejemplo de ejecución

```
1
Ready P0
-
```

El siguiente programa selecciona una de las unidades angulares (DEG, RAD, GRA) y calcula el seno de X.

```

10 REM SIN X EXAMPLE
20 INPUT "ANGLE=";K
30 ANGLE K
40 INPUT "SIN X:X=";X
50 PRINT "SIN";X;"=";SINX
60 STOP
70 END

```

Al ejecutarse este programa, se solicita primeramente la unidad angular.

ANGLE = ?

Especifique la unidad del siguiente modo:

ANGLE 0 → GRADOS
 ANGLE 1 → RADIANES
 ANGLE 2 → GRADIENTES

Elija, como en el ejemplo, la unidad RADIANES. Entre para ello un "1".

SIN X X = ? En la pantalla

Entre el ángulo (recuerde que es en radianes), digamos $\pi/4$, y el resultado deberá ser el siguiente.

Ejemplo de ejecución

```

SIN 0.7853981634=0.
7071067812
STOP P0-60
-

```

Si bien se modifica la unidad angular mediante el mando ANGLE, la gama de entrada para cada unidad angular se limita a las siguientes cifras:

$$\begin{aligned} \text{DEG} & -5400^\circ < \text{expresión numérica} < 5400^\circ \\ \text{RAD} & -30\pi < \text{expresión numérica} < 30\pi \\ \text{GRA} & -6000 < \text{expresión numérica} < 6000 \end{aligned}$$

Cuando la expresión numérica excede los límites especificados, aparece un error BS.

Para el argumento, se puede usar una expresión numérica, un número entero (como ser 30) o una variable.

Cuando se usa solamente un número entero o una variable, el argumento no necesariamente tiene que estar dentro del paréntesis. Sin embargo, cuando se usa una expresión numérica, el resultado variará según si se usa o no un paréntesis. Veamos dos ejemplos al respecto:

SIN X+Y Se suma Y al resultado del seno de X.
 SIN (X+Y) Se calcula el seno de X + Y.

VER ANGLE, COS, TAN

Memorandum

Hay tres unidades para expresar el ángulo de una función trigonométrica. Ellas son: grados (DEG), radianes (RAD) y gradientes (GRAD).

DEG 1° es la $1/360^{\text{ava}}$ parte de la circunferencia de un círculo.
 RAD 1 rad. es la $1/2\pi^{\text{ava}}$ parte de la circunferencia de un círculo.
 GRAD 1 grad. es la $1/400^{\text{ava}}$ parte de la circunferencia de un círculo.

Los grados y radianes se utilizan principalmente, y tienen la relación a continuación.

$$1^\circ = \pi / 180 \text{ rad} = 3.141592654 / 180 \text{ rad}$$

COS

| | |
|----------------|--|
| Funcion | Determina el coseno del ángulo X (COS X). |
| Formato | COS expresión numérica $-5400^\circ < \text{expresión numérica} < 5400^\circ$ |

La función COS se utiliza para calcular el coseno del ángulo X. Tanto las unidades angulares, como los límites para el argumento X y el tipo de precisión son iguales a los presentados en la función SIN X.

Ejemplo

Programa en el que se usa COS X.

```

10 REM COS X EXAMPLE
20 INPUT "ANGLE=";K
30 ANGLE K
40 INPUT "COS X: X=";X
50 PRINT "COS"; X; "="; COSX
60 STOP
70 END

```

Al ejecutarse este programa, se solicita la entrada de la unidad angular mediante:

ANGLE = ?

Supongamos que se usan gradientes. Entre entonces un "2". La entrada del ángulo se solicita con la siguiente visualización:

COS X X = ?

Entre un ángulo de 1355 gradientes. El resultado será, entonces, igual a $-0,7604059656$.

 SIN, ANGLE, TAN

TAN

| | |
|----------------|--|
| Función | Determina la tangente del ángulo X (TAN X). |
| Formato | TAN expresión numérica $-5400^\circ < \text{expresión numérica} < 5400^\circ$ Excepto expresión numérica = $(2n - 1) \times 1$ ángulo recto ($n = \text{entero}$) |

La función TAN se utiliza para calcular la tangente del ángulo X. Las unidades son las mismas usadas en SIN X y COS X.

Ejemplo

Programa para obtener la tangente del ángulo X.

```

10 REM TAN X EXAMPLE
20 INPUT "ANGLE=";K
30 ANGLE K
40 INPUT "TAN X:X=";X
50 PRINT "TAN";X;"=";TANX
60 STOP
70 END

```

Cuando se averigua la tangente de un ángulo de 45° usando la unidad DEG (grados: ANGLE 0), el resultado se visualiza del siguiente modo:

TAN 45 = 1

Si se intentara obtener la tangente de 90° , aparecería en la pantalla un error MA. Ello se debe a que la tangente de un ángulo de 90° es de valor infinito y la computadora no puede calcularlo.

Cuando el valor de X en TAN X es $\pm 90 * (2n-1)$, donde n es un entero, deben tomarse las precauciones del caso, ya que podrí aparecer un error como el mencionado.

VER SIN, COS, ANGLE

ASN, ACS, ATN

| | |
|------------------|--|
| Funciones | ASN determina el seno de arco de X ($\text{seno}^{-1} X$) ACS determina el coseno de arco de X ($\text{cos}^{-1} X$) ATN determina la tangente de arco de X ($\text{tan}^{-1} X$). |
| Formatos | ASN X, ACS X, ATN X. $ X \leq 1$ (ASN X, ACS X) $ X \leq 10^{100}$ (ATN X) |

Las funciones ASN, ACS y ATN se utilizan para computar las funciones trigonométricas inversas: Sen^{-1} , COS^{-1} y Tan^{-1} .

El seno, coseno y tangente se utilizan para determinar las funciones trigonométricas en base a valores angulares. Las funciones trigonométricas inversas, por su lado, sirven para determinar valores angulares cuando se conocen las funciones trigonométricas de los mismos.

Ejemplo

Veamos a continuación un programa que usa la función ASN X.

```

10 REM ASN X EXAMPLE
20 INPUT "ANGLE=" ; K
30 ANGLE K
40 INPUT "ASN X: X=" ; X
50 PRINT "ASN" ; X ; "=" ; ASN X
60 STOP
70 END

```

Al ejecutarse este programa, se visualizan los dos siguientes mensajes:

ANGLE = ? 0 Para especificar la unidad angular DEG.
ASN X X = ? 1 Para entrar el valor de la función trigonométrica.

Al entrarse estos valores, aparece lo siguiente en la pantalla:

ASN 1 = 90

En otras palabras, se obtuvo el ángulo con un seno igual a uno (1). Ejecute este programa nuevamente usando ACS y ATN en lugar de ASN. La especificación de la unidad angular (ANGLE) para estas funciones trigonométricas inversas es igual a la utilizada para las funciones SIN, COS y TAN.

Los límites de los ángulos para la unidad DEG (grados) es la siguiente:

$$\begin{array}{l} -90^\circ \leq \text{ASN} \leq 90^\circ \\ 0^\circ \leq \text{ACS} \leq 180^\circ \\ -90^\circ \leq \text{ATN} \leq 90^\circ \end{array}$$

Como SIN X y COS X no exceden teóricamente 1, el valor del argumento X en ASN X y ACS X no debe ser mayor que 1.

VER SIN, COS, TAN, ANGLE

HYP SIN/HYPCOS/HYPTAN

| | |
|-----------------|--|
| Función | Determina funciones hiperbólicas. |
| Formatos | HYP SIN numerical expression HYPCOS numerical expression $-10^{100} < \text{Numerical expression} \leq 230.2585092$ HYPTAN numerical expression $ \text{Numerical expression} < 10^{100}$ |

Estas funciones sirven para determinar funciones hiperbólicas. Las siguientes son las expresiones numéricas que les corresponde a cada una de ellas.

$$\text{HYP SIN} : \sinh x = (e^x - e^{-x}) / 2$$

$$\text{HYP COS} : \cosh x = (e^x + e^{-x}) / 2$$

$$\text{HYP TAN} : \tanh x = (e^x - e^{-x}) / (e^x + e^{-x})$$

HYPASN/HYPACS/HYPATN

| | |
|-----------------|--|
| Función | Determina funciones hiperbólicas inversas. |
| Formatos | HYPASN numérical expresión HYPACS numérical expresión $ \text{Numérical expresión} < 5 \times 100^{99}$ HYPATN numérical expresión $ \text{Numérical expresión} < 1$ |

Estas funciones sirven para determinar funciones hiperbólicas inversas. Las siguientes son las expresiones numéricas que les corresponde a cada una de ellas.

$$\text{HYP ASN} : \sinh^{-1} x = \log (x + \sqrt{x^2 + 1})$$

$$\text{HYP ACS} : \cosh^{-1} x = \log (x + \sqrt{x^2 - 1})$$

$$\text{HYP ATN} : \tanh^{-1} x = \frac{1}{2} \log \frac{1+x}{1-x}$$

SQR

| | |
|----------------|---|
| Función | Determina la raíz cuadrada de un argumento |
| Formato | SQR expresión numérica Expresión numérica ≥ 0 |

La función SQR se utiliza para obtener la raíz cuadrada del siguiente modo:

$$\text{SQR } X = X^{0.5} = \sqrt{X}$$

En este caso, el valor de X debe ser mayor que 0 (cero).

Ejemplo

En el programa a continuación, se entra el área de un círculo para averiguar el radio del mismo.

```

10 REM SQR X EXAMPLE
20 INPUT "CIRCLE AREA=";S
30 R=SQR(S/PI)
40 PRINT "CIRCLE RADIUS=";R
50 LOCATE 0,2
60 END

```

Al ejecutarse este programa, aparece lo siguiente en pantalla:

```
CIRCLE AREA = ?
```

Entre un área igual a 100. Así, el radio obtenido será igual a 5,641895835.

Si ejecuta el programa y entra un número negativo, aparecerá en la pantalla un error MA, ya que siempre que el argumento X de SQR X sea negativo, SQR (S/Pi) se convertiría en un número imaginario. Para evitar este tipo de error, se recomienda agregar la siguiente línea para verificar si el argumento es positivo o negativo.

```
35 IF S<0 THEN 20
```

LOG, LGT

| | |
|-----------|---|
| Funciones | LOG X ... Determina el valor de un logaritmo natural $\log_e X$ LGT X ... Determina el valor de un logaritmo común $\log_{10} X$ |
| Formatos | LOG expresión numérica Expresión numérica > 0 LGT expresión numérica |

LOG X computa el valor de un logaritmo natural; su formato es $\log_e X$ ($\ln X$). En este caso, "e" es la raíz del logaritmo natural y tiene el siguiente valor:

$$e = 2,718281828 \dots$$

LGT X computa el valor de un logaritmo común; su formato es $\log_{10} X$. La raíz del logaritmo común es igual a 10.

Ejemplo

El siguiente programa computa LOG X para valores sucesivos de X.

```

10 REM LOG X EXAMPLE
20 INPUT "X="; X
30 PRINT "LOG"; X; "="; LOGX
40 GOTO 20

```

Al ejecutar este programa, aparece en pantalla:

X = ?

solicitando el valor máximo para el argumento X de LOG X.

Si se entrara un "1", el valor de LOG X se visualiza como LOG(1)=0 y se solicita la entrada del siguiente valor de X.

Ejemplo de ejecución

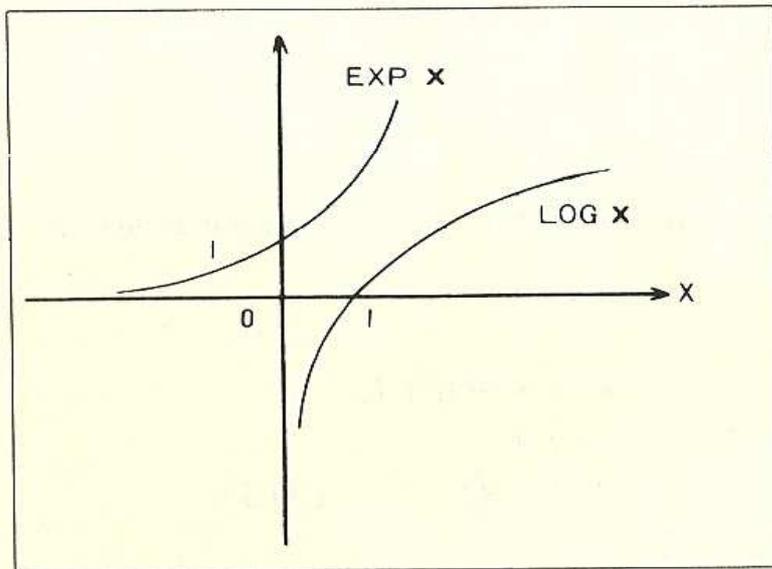
CLS RUN ↵

1 ↵

```

RUN
X=?
LOG 1=0
X=?_
    
```

La función logarítmica LOG X es inversamente proporcional a la función exponencial EXP X. Su relación puede verse en el siguiente gráfico:



En la función logarítmica como la del gráfico, X debe ser mayor que 0. Si el valor de X fuera negativo, aparece en pantalla un error MA.

Mientras que para averiguar el logaritmo de X con raíz e se utiliza LOG X, es posible obtener el logaritmo con otra raíz que no sea e (log_yX, por ejemplo) mediante la siguiente fórmula:

$$\text{LOG X/LOG Y}$$

Es así como se puede averiguar también el logaritmo natural de X usando la siguiente fórmula en lugar de LGT X:

$$\text{LOG X/LOG 10}$$

EJEMPLO DE PROGRAMA

* Este programa permite obtener diferentes valores logarítmicos con las raíces entradas.

```

10 REM ** LOG X/LOG Y **
20 INPUT "X="; X
30 INPUT "Y="; Y
40 PRINT "LOG"; X; "/LOG"; Y; "="; LOGX/LO
  GY
50 LOCATE 0,3
60 STOP
70 END

```

Este programa calcula el valor de $\log_Y X$ por medio de la fórmula $\text{LOG } X/\text{LOG } Y$.

Veamos a continuación un ejemplo de ejecución.

Ejemplo de ejecución

```

RUN 
10 
  2 

```

```

LOG 10/LOG 2=3.3219
28095
STOP PO-60
-

```

 EXP

EXP

| | |
|----------------|--|
| Función | Sirve como función exponencial (e^x). |
| Formato | EXP expresión numérica $-10^{100} < \text{expresión numérica} \leq 230.2585092$ |

La función EXP se utiliza para calcular el valor de una función exponencial (e^x). El valor de "e", raíz de la función exponencial, es igual a:

$$e = 2,718281828 \dots$$

Como frecuentemente se menciona un "aumento exponencial", y en vista de que la naturaleza de esta función aumenta de acuerdo al incremento del valor del argumento X, el valor de EXP X aumenta considerablemente.

Ejemplo

Entre el siguiente programa para observar cómo cambia el valor de EXP X.

```

10 REM ** EXP X **
20 INPUT "A="; A
30 FOR X=1 TO A
40 PRINT "EXP"; X; "="EXPX
50 FOR I=1 TO 300:NEXT I
60 NEXT X
70 END

```

Al ejecutarse este programa, se solicita la entrada del valor máximo para el argumento X de EXP (X).

A = ?

Entre, por ejemplo, "10" y presione la tecla . Se visualizarán así los resultados del modo ilustrado en la página siguiente.

Ejemplo de ejecución

```
EXP 1= 2.718281828
EXP 2= 7.389056099
EXP 3= 20.08553692
EXP 4= 54.59815003
EXP 5= 148.4131591
EXP 6= 403.4287935
EXP 7= 1096.633158
EXP 8= 2980.957987
EXP 9= 8103.083928
EXP 10= 22026.46579
```

Como podrá verse, el valor de EXP X aumenta súbitamente.

Vuelva a ejecutar el mismo programa entrando esta vez "231" para A. Los valores de EXP X se visualizarán uno tras otro y, cuando el valor del argumento sea igual a 231, aparece un error MA.

Ello se debe a que el límite para el argumento X de EXP (X) es:

$$X \leq 230.2585.$$

El valor de EXP X con X = 230.2585 es el siguiente:

Ejemplo de ejecución

```
EXP 230.2585= 9.999907006E99
```

El cálculo casi genera una condición de error por superación del límite de cálculo de la computadora en base al resultado previamente mencionado.

ABS

| | |
|----------------|--|
| Función | Determina el valor absoluto de un argumento. |
| Formato | ABS expresión numérica |

ABS X determina el valor absoluto de X y se expresa matemáticamente del siguiente modo:

$$\text{ABS } X = |X|$$

En cuanto al argumento X de ABS X:

Cuando $X \geq 0$ (X es positivo) ABS X = X,
 y cuando $X < 0$ (X es negativo) ABS X = -X.

En otras palabras, ABS X lleva a cabo los cálculos de tal manera que permite convertir el resultado en un número positivo (número absoluto).

Ejemplo

Veamos a continuación un programa que usa la función ABS.

```

10 REM ABS X EXAMPLE
20 READ A,B,C,D
30 X=A:GOSUB 40:X=B:GOSUB 40:X=C:GOSU
  B 40:X=D:GOSUB 40:END
40 PRINT "ABS";X;"=";ABSX
50 FOR I=1 TO 200:NEXT I
60 RETURN
70 DATA 5,-5,0,-7.5

```

Este programa asigna los valores 5, -5, 0 y -7,5 a las variables A, B, C y D por medio de la sentencia READ y calcula luego ABS A hasta ABS D.

El resultado aparece en la pantalla del siguiente modo:

Ejemplo de ejecución

```

ABS 5=5
ABS -5=5
ABS 0=0
ABS -7.5=7.5

```

Tal como puede verse a continuación, el uso de la función SGN permite realizar una operación similar a la de la función ABS.

$ABS X$ es igual que $X * SGN X$

EJEMPLO DE PROGRAMA

Este programa no genera error alguno aún si se entra un valor negativo.

```

10 INPUT X
20 S=SQR(ABSX)
30 L=LOG(ABSX)
40 PRINT "SQRX=";S
50 FOR I=1 TO 200:NEXT I
60 PRINT "LOGX=";L
70 END

```

Siempre que el argumento X de las funciones $SQR X$ y $LOG X$ sea negativo, aparece un error MA.

Es por ello que en este programa se lleva a cabo el cálculo usando el valor absoluto de X .

 SGN

INT

| | |
|----------------|--|
| Función | Determina el mayor número entero que no exceda el valor del argumento. |
| Formato | INT expresión numérica |

INT X determina el mayor número entero que no exceda el valor de X. Veamos a continuación varios ejemplos con los siguientes valores para X: 3,9, 0,5, -0,5 y -3,9.

INT 3.9 = 3
INT 0.5 = 0
INT -0.5 = -1
INT -3.9 = -4

Cuando el valor del argumento es positivo, se descarta la parte decimal del número. Cuando es negativo, sin embargo, deben tomarse ciertas precauciones. Por ejemplo, si el argumento es -0,5, no es 0 sino -1 el número entero mayor que no excede -0,5.

Ejemplo

Veamos el programa a continuación:

```
10 REM INTX EXAMPLE
20 READ A,B,C
30 X=A:GOSUB 40:X=B:GOSUB 40:X=C:GOSUB
  B 40:END
40 PRINT "INT";X;"=";INTX
50 FOR I=1 TO 200:NEXT I
60 RETURN
70 DATA 5.3,0.5,-3.9
```

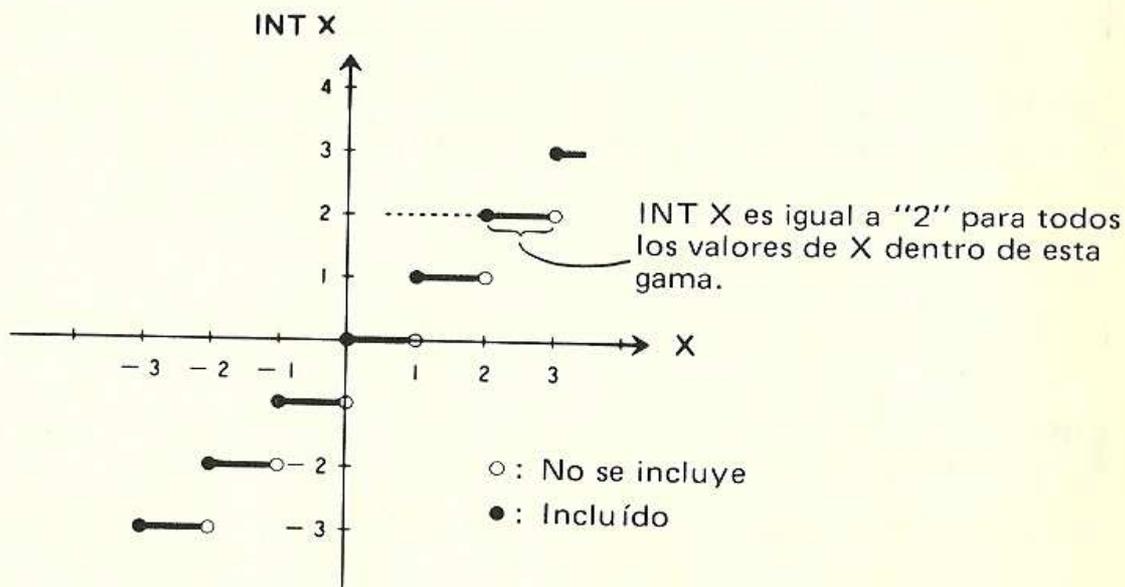
Ejemplo de ejecución

INT 5.3 = 5
INT 0.5 = 0
INT -3.9 = -4

Al ejecutarse este programa, se visualiza el resultado que aparece en la página anterior.

La función INT puede graficarse dando los valores de X para las coordenadas horizontales y los valores de INT X para las coordenadas verticales.

Gráfico de INT X



Este gráfico es útil para observar la diferencia entre los valores negativos y positivos.

La función INT se utiliza con frecuencia junto con otras funciones, como ser la función RND. Por otro lado, las funciones FRAC, ROUND, etc. son similares a la función INT.

EJEMPLO DE PROGRAMA

* Este programa visualiza 5 números enteros aleatoriamente entre 0 y 9.

```
10 FOR I=1 TO 5
20 PRINT INT(10*RND);
30 NEXT I
40 END
```

En este ejemplo, se combinaron las funciones INT y RND.

FRAC, ROUND, RND

FRAC

| | |
|----------------|--|
| Función | Determina el valor de la parte fraccionaria del argumento. |
| Formato | FRAC expresión numérica |

FRAC X determina el valor de la parte fraccionaria de X. Veamos a continuación algunos ejemplos:

$$\text{FRAC } 1.123 = 0.123$$

$$\text{FRAC } -1.123 = -0.123$$

Como puede observarse, sólo descarta la parte entera del número.

Ejemplo

Pruebe el siguiente ejemplo entrando diferentes valores.

```

10 REM FRAC X EXAMPLE
20 INPUT "NUMBER"; X
30 PRINT FRACX
40 GOTO 10

```

EJEMPLO DE PROGRAMA

* Este programa genera un número aleatorio con 9 decimales y toma un dígito por vez para convertirlo en un entero de un sólo dígito.

```

10 DIM Y(9)
20 X=RND
30 FOR I=1 TO 9
40 Y(I)=INT(10*X)
50 PRINT "X="; X
60 PRINT "Y("; I; ")="; Y(I)
70 FOR J=1 TO 150:NEXT J
80 X=FRAC(10*X)
90 NEXT I

```

Este programa asigna un valor generado por la función RND a la variable X. El valor se multiplica por 10 y luego se utiliza la función INT para obtener un número entero de un sólo dígito. Se asigna el valor a la variable Y(1), y se toma la parte decimal que resta por medio de la función FRAC, la cual se asigna a X. El proceso comienza nuevamente desde el principio. El mismo se repite 9 veces y los 9 enteros de un sólo dígito obtenidos se asignan a las variables matriciales Y (1) hasta Y (9).

0.8928802520
 Y(1) Y(2) Y(3) Y(4) Y(5) Y(6) Y(7) Y(8) Y(9)

INT, RND, ROUND

Agregue al programa las dos líneas a continuación y vuelva a ejecutarlo:

SGN

| | |
|----------------|---|
| Función | Determina el signo (+ ó -) de un argumento. |
| Formato | SGN expresión numérica |

La función SGN X juzga el signo (+ ó -) del valor del argumento X. Puede dar los tres siguientes resultados:

$X > 0$ (Si es positivo) SGN X = 1
 $X = 0$ (Si es igual a 0) SGN X = 0
 $X < 0$ (Si es negativo) SGN X = -1

Ejemplo

Ejecute el siguiente programa:

```

10 REM SGN X EXAMPLE
20 PRINT "JUDGEMENT OF + OR -"
30 INPUT "NUMBER";X
40 A=SGNX
50 IF A=1 THEN PRINT "+":GOTO 30
60 IF A=0 THEN PRINT "0":GOTO 30
70 IF A=-1 THEN PRINT "-":GOTO 30
  
```

Las líneas 50 a 70 visualizan un "+" cuando el valor entrado es positivo, un "-" cuando es negativo y un cero (0) cuando el valor es igual a cero, todo ello en base al uso de la función SGN X.

EJEMPLO DE PROGRAMA

* He aquí un programa para obtener una curva sinusoidal.

```

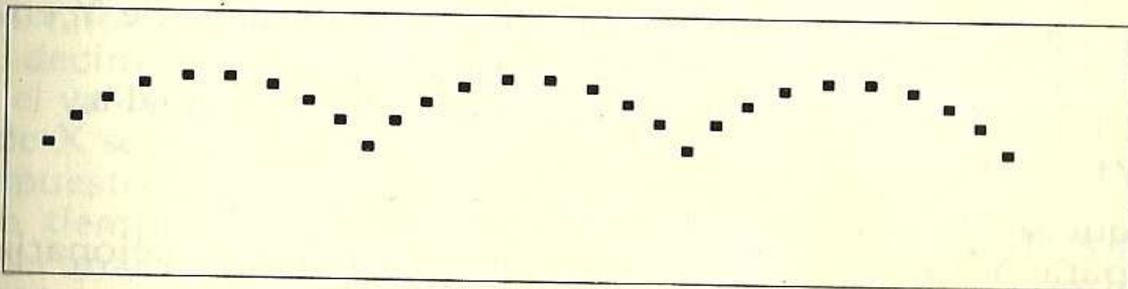
10  CLS :FOR X=0 TO 540 STEP 20
20  S=SGN(SINX)
30  Y=S*INT(S*10*SINX)
40  IF S<0 THEN 70
50  DRAW(X/4,16-Y)
60  GOTO 80
70  DRAW(X/4,16+Y)
80  NEXT X
90  END

```

Este programa obtiene el seno de X ($\text{SIN } X$) a incrementos de 20° , dado que X es igual a 0 hasta 540° . Estos valores se multiplican por 10 para producir un valor de un sólo dígito, el cual se asigna a Y .

Allí, se toma la función SGN para $\text{SIN } X$ y, aun si $\text{SIN } X$ produce un número negativo, la función INT obtiene solamente la parte entera.

De este modo, se trazan mediante el mando DRAW los valores de X e Y , con el centro de coordenadas en $(0,16) - (159,16)$. Si un valor es positivo, se traza $(X/4,16-Y)$; en cambio, si es negativo, se traza $(X/4,16+Y)$ (ver el mando DRAW). La ejecución de este programa permite trazar una curva sinusoidal como la que se observa a continuación.

Resultado de la ejecución

ROUND

| | |
|----------------|--|
| Función | Determina el valor de la expresión numérica 1 redondeada a la posición especificada por la expresión numérica 2. |
| Formato | ROUND (expresión numérica 1, expresión numérica 2) Expresión numérica 2: Número de posiciones |

ROUND (X, Y) determina el valor de X redondeado a la posición 10^Y del siguiente modo:

$$\text{ROUND}(12345, 2) = 12000$$

El valor numérico 12345 se redondea a la posición 10^2 , o sea la posición de las centenas.

Ejemplo

El siguiente programa sirve para clarificar el funcionamiento de ROUND.

```

10 FOR Y=3 TO -5 STEP -1
20 X=12345.67891
30 Z=ROUND(X,Y)
40 PRINT USING"#####.#####";X,Z:PRINT
50 FOR J=0 TO 150:NEXT J
60 NEXT Y
70 END

```

Este programa lleva a cabo un cálculo en el cual el valor de Y en ROUND (X, Y) disminuye de a uno (1) desde 3 hasta -5.

El valor especificado para el segundo argumento Y es igual a

$$|Y| < 100$$

Siempre que se exceda esta límite, aparece un error BS.

Cuando para Y se especifica un valor con parte fraccionaria, ésta se descarta.

Ejemplo de ejecución

```

12345.67891
10000.00000
12345.67891
12000.00000
12345.67891
12300.00000
12345.67891
12350.00000
12345.67891

```

EJEMPLO DE PROGRAMA

Este programa visualiza números binarios de 8 bit aleatorios y sus equivalentes decimales.

```

10 Y=0
20 FOR I=7 TO 0 STEP -1
30 X=ROUND(RND,-1)
40 Y=Y+(2^I)*X
50 PRINT X;
60 NEXT I
70 PRINT "=";Y
80 LOCATE 0,1
90 J=0 TO 500:NEXT J
100 END

```

Como los valores a continuación del punto decimal se redondean a una posición decimal por medio de la función RND (página 239) en la línea 30, el valor de X sólo puede ser igual a 1 ó 0.

El valor de X se genera 8 veces para proporcionar un número aleatorio de 8 bit compuesto por ceros y unos.

Al mismo tiempo, en la línea 40 se transforma el valor de 8 bit en un número decimal equivalente, el cual se visualiza en forma continua. He a continuación un ejemplo de ejecución:

RUN 

```

RUN
00101001 = 41

```

PI

| | |
|----------------|--|
| Función | Da la relación de la circunferencia de un círculo con su diámetro (π). |
| Formato | PI |

PI da aproximaciones para la relación de la circunferencia de un círculo con su diámetro (π).

El valor de π es el siguiente:

$$\pi = 3,141592654 \dots$$

Ejemplo

El siguiente programa calcula el área de un círculo:

```

10 REM PI EXAMPLE
20 PRINT "CIRCLE AREA"
30 INPUT "RADIUS";R
40 S=PI*R^2
50 PRINT "S=";S
60 END

```

Si se entra un 5 como valor para el radio, el área del círculo visualizada será:

Ejemplo de ejecución

S= 78.53981634

RND

| | |
|---------|---------------------------------------|
| Función | Da un número aleatorio |
| Formato | RND $0 < \text{número aleatorio} < 1$ |

La función RND da un número aleatorio de 10 dígitos mayor que 0 y menor que 1.

Los mismos se utilizaron en un principio para la simulación de fenómenos estadísticos o modelos de probabilidad; en la actualidad, se aplican para simular pronósticos económicos y en juegos electrónicos. En particular, la diversión de los juegos electrónicos se debe en gran parte gracias a esta función generadora de números aleatorios.

Ejemplo

El programa a continuación genera y presenta en la pantalla 10 números aleatorios.

```

10 FOR N=1 TO 10
20 PRINT RND
30 FOR X=1 TO 500:NEXT X
40 NEXT N
99 END

```

Estos resultados son tan sólo ejemplos. Cada vez que ejecute el programa, se obtendrán naturalmente resultados diferentes.

```

0.6791506196
0.9598232115
0.2057199883
0.5039057551
0.306977109
0.1065778556
0.4177075471
0.5017414683
0.7551551958
0.4560918328

```

Como son muchos los dígitos que se generan, no pueden usarse así como están. Es por ello que, cuando se los usa para ciertas aplicaciones, deben modificarse mediante el uso de las funciones INT y ROUND del siguiente modo:

(1) Generación de enteros hasta la posición deseada

$\text{INT}(\text{RND} * 10^L)$ L indica el número de dígitos.

(2) Generación de enteros desde N hasta el límite superior M.

$\text{ROUND}(\text{RND} * (M - N), -1) + N$ N y M son enteros. ($N < M$)

DEG

| | |
|----------------|--|
| Función | Convierte sexagesimales en hexadecimales |
| Formato | DEG (grados (, minutos (, segundos))) |

La función DEG convierte valores sexagesimales en valores decimales. Los grados, minutos y segundos de un valor sexagesimal tienen la siguiente relación con los de los valores decimales.

$$\text{DEG (grad., min, seg.)} = \text{grad.} + \text{min.}/60 + \text{seg.}/3600$$

La entrada debe estar dentro de los siguientes límites.

$$|\text{DEG (grad., min., seg.)}| < 10^{100}$$

Entre valores sexagesimales en el siguiente programa, para verificar la función DEG.

```

10 REM DEG EXAMPLE
20 INPUT "DEG. = ", A
30 INPUT "MIN. = ", B
40 INPUT "SEC. = ", C
50 D=DEG(A,B,C)
60 PRINT "DEG(";A;",";B;",";C;")"
70 PRINT "=";D
80 GOTO 20

```

DMSS

PEEK

| | |
|----------------|--|
| Función | Lee el contenido de la dirección de memoria especificada. |
| Formato | PEEK ($\frac{\text{Dirección de memoria}}{\text{Expresión numérica}}$) |

La función PEEK produce el valor almacenado en la dirección de memoria especificada. Los límites para la especificación de las direcciones se dan a continuación. Todo decimal que aparezca en la especificación de una dirección se descarta.

$$-32769 < \text{dirección} < 65536$$

Para mayor información sobre las direcciones, vea la sección dedicada al mando CLEAR.

El siguiente programa presenta en la pantalla el contenido de la dirección de memoria entrada.

```

10 REM PEEK EXAMPLE
20 INPUT "Address = ", A
30 B=PEEK(A)
40 PRINT B
50 GOTO 20

```

VER CLEAR, POKE

4-4 FUNCIONES DE CARACTERES

ASC

| | |
|-----------------|---|
| Función | Determina el código decimal del primer carácter de una secuencia de caracteres. |
| Formatos | ASC "secuencia de caracteres" ASC (variable de caracteres) |

Todos los caracteres, números y símbolos que puedan visualizarse en la pantalla de la PB-770 tienen un número correspondiente al Código ASCII.

"A" 65
"B" 66
"6" 54

(Ver la TABLA DE CODIGOS PARA CARACTERES, página 328)

Estos códigos pueden averiguarse directamente mediante el use de la función ASC que la PB-770 incorpora u observando la TABLE DE CODIGOS PARA CARACTERES.

Ejemplo

Entre PRINT ASC ("E") 

Así se visualizará
el código 69 correspondiente
a la letra E.

Si se intentara la entrada de más de un carácter en el ejemplo anterior, como ser:

PRINT ASC ("EF") 

sólo aparecerá en la pantalla el código de la letra inicial, en este caso E. Cuando se desea determinar el código ASCII de un carácter dentro de una secuencia de caracteres (como ser "ABCDEF"), debe usarse la función MID\$ explicada más adelante (ver MID\$).

EJEMPLO DE PROGRAMA

He aquí un programa que visualiza el código ASCII del carácter entrado.

```
10 REM ASCII CODE
20 CLS
30 INPUT "WHICH CHARACTER"; A$
40 PRINT ASC(A$)
50 GOTO 30
```

La entrada del carácter se lleva a cabo en la línea 30, mientras que la línea 40 visualiza el código. Como la línea 50 retorna el control a la línea 30, el programa continúa su ejecución en forma indefinida.

El resultado de la ejecución sería, entonces, el siguiente:

Ejemplo de ejecución

```
WHICH CHARACTER? A
65
WHICH CHARACTER? T
84
WHICH CHARACTER? R
82
WHICH CHARACTER? 7
55
WHICH CHARACTER? 1
49
```

Para interrumpir la ejecución del programa, presione la tecla **BRK** .

VER CHR\$

CHR\$

| | | |
|----------------|---|------------------------------|
| Función | Determina el caracter correspondiente a un código ASCII dado. | |
| Formato | CHR\$ (código) | $0 \leq \text{código} < 256$ |

CHR\$ se utiliza para obtener el carácter, número o símbolo correspondiente a un código ASCII dado.

Ejemplo

Ent e PRINT CHR\$ (66) ↵

Se visualizará así la letra "B" correspondiente al código ASCII 66. Para determinar dos caracteres, siga el siguiente procedimiento:

PRINT CHR\$ (71); CHR\$ (80) ↵.

Así se visualizarán los caracteres "G" y "P" correspondientes a los códigos 71 y 80.

Los caracteres que pueden entrarse desde el teclado de la PB-770 se limitan a letras mayúsculas y minúsculas, números y algunos símbolos. Los demás caracteres, como ser gráficos, pueden llevarse a la pantalla por medio de la función CHR\$ (ver la TABLA DE CARACTERES en la página 328.) El código especificado puede ser de 0 hasta 256. Se omite la parte decimal.

EJEMPLO DE PROGRAMA

```

5 U=0
10 FOR I=33 TO 254
20 PRINT I
30 U=U+1
40 FOR J=1 TO 14
50 PRINT CHR$(I);
60 NEXT J
70 PRINT
80 IF U<3 THEN 110
90 K$=INKEY$: IF K$="" THEN 90
100 U=0
110 NEXT I
120 END

```


VAL

| | |
|-----------------|---|
| Función | Convierte una secuencia de caracteres en un valor numérico. |
| Formatos | VAL "secuencia de caracteres" VAL (variable de caracteres) |

VAL es una función que convierte un carácter en un valor numérico. Para poder comprender esta función, es necesario saber la diferencia entre un carácter y un valor numérico.

Compare los siguientes ejemplos de programa.

| Programa (1) | Programa (2) |
|--------------|-----------------|
| 10 READ A,B | 10 READ A\$,B\$ |
| 20 C=A+B | 20 C\$=A\$+B\$ |
| 30 PRINT C | 30 PRINT C\$ |
| 40 END | 40 END |
| 50 DATA 3,5 | 50 DATA 3,5 |

En el Programa (1), la información numérica se asigna a las variables numéricas A y B, mientras que el resultado se visualiza asignándolo previamente a la variable C. El resultado del programa es, entonces:

— 8

En la presentación en la pantalla, queda un espacio en blanco del lado izquierdo del resultado, el cual corresponde al signo + (siempre omitido) ya que el resultado es positivo.

En el Programa (2), por su lado, los números 3 y 5 se asignan a las variables A\$ y B\$ donde se tratan como datos de caracteres.

En cuanto a las operaciones con variables de caracteres, sólo se pueden encadenar. Es así como el resultado asignado a la variable C\$ sería:

35

En este caso, al ser el contenido de una variable de caracteres el visualizado, no se reserva espacio para signo alguno, ya que el resultado no es un valor numérico sino dos caracteres encadenados.

Este espacio en blanco es muy importante, por lo que veremos más claramente la diferencia ejecutando los programas (3) y (4) a continuación:

Programa (3)

```

10 FOR I=1 TO 10
20 READ A
30 PRINT A;
40 NEXT I
50 END
60 DATA3,8,-6,7,21
70 DATA223,18,8,1,0
    
```

Ejemplo de ejecución 3 8-6 7 21 223 18 8
 1 0

Programa (4)

```

10 FOR I=1 TO 10
20 READ A$
30 PRINT A$;
40 NEXT I
50 END
60 DATA3,8,-6,7,21
70 DATA223,18,8,1,0
    
```

Ejemplo de ejecución 38-672122318810

Uso

La función VAL se utiliza para realizar cálculos como los del Programa (1) de arriba, usando los números obtenidos de las variables A\$ y B\$, como en el Programa (2).

```

10 READ A$,B$
20 C=VAL(A$)+VAL(B$)
30 PRINT C
40 END
50 DATA3,5

```

La ejecución de este programa visualiza el resultado "8", igual al del Programa (1). En el uso de la función VAL, deberán tenerse en cuenta las siguientes precauciones:

- (1) Cuando la secuencia contiene un carácter que no sea un valor numérico, un punto decimal, un signo (+ ó -) o un signo exponencial "E", se ignoran todos los caracteres que le siguen. (De aquí en más, las apariciones de más signos exponenciales "E" se ignoran.)
- (2) Se salta el primer espacio de las secuencias de caracteres.
- (3) Cuando el carácter inicial de una secuencia no es un valor numérico, un punto decimal o un signo, o cuando la secuencia de caracteres contiene sólo un signo o un punto decimal, la función VAL da un valor igual a 0 (cero).
- (4) Siempre que haya más de 2 números luego del signo exponencial "E", aparecerá un error SN.

EJEMPLO DE PROGRAMA

He aquí una subrutina que solicita la entrada de un número, de 1 a 5, para la selección de cinco diferentes procesos.

```

100 Z$=INKEY$: IF Z$="" THEN 100
110 IF Z$<"1" THEN 100
120 IF Z$>"5" THEN 100
130 GOSUB VAL(Z$)*1000

```

En esta subrutina, al pulsarse cualquier tecla del 1 al 5, el programa salta a cada subrutina contenida en las líneas 1000 a 5000. Si se presionara una tecla que no corresponda a los números 1 al 5, el programa vuelve al estado de espera de entrada inicial. Es muy conveniente para la selección de subrutinas al comienzo de un programa donde se realizan varios procesos.

STR\$

STR\$

| | |
|----------------|---|
| Función | Convierte un valor numérico en una secuencia de caracteres. |
| Formato | STR\$ (expresión numérica) |

La función STR\$ convierte un valor numérico en una secuencia de caracteres.

Uso

¿Cuál sería el resultado de la ejecución de los Programas (1) y (2)?

```

Programa (1)
10 A=25:B=30
20 C$=STR$(A+B)
30 PRINT C$
40 END

```

```

Programa (2)
10 A=25:B=30
20 C$=STR$(A)+STR$(B)
30 PRINT C$
40 END

```

Si bien estos dos programas parecen similares, el resultado del primero es "55", mientras que el del segundo es igual a "25 30".

En el Programa (1), la función STR\$ convierte el resultado de la expresión numérica $A + B$ en una secuencia de caracteres. En el Programa (2), en cambio, se convierten las variables numéricas A y B en secuencias de caracteres y se las encadena. Esta diferencia aparece en los resultados.

EJEMPLO DE PROGRAMA

* Programa de suma.

```

10 REM ADDITION
20 FOR I=1 TO 5
30 X=INT(RND*100)
40 Y=INT(RND*100)
50 Z=X+Y
60 PRINT STR$(X); "+"; RIGHT$(STR$(Y), L
EN(STR$(Y))-1);
70 INPUT "=", A

```

```

80 IF A=Z THEN PRINT "OK" ELSE 60
90 NEXT I
100 END

```

(Ver RIGHT\$, LEN.)

Este programa genera cinco problemas de suma. El operador suministra la respuesta para la suma de dos enteros de hasta dos dígitos. El par de números utilizado no puede predecirse, ya que se utiliza la función RND. La función STR\$ se utiliza en la línea 60, donde se visualiza el problema.

Si bien `60 PRINT X; "+"; Y;`

sin usar STR\$ parece ser razonable, este método visualiza un espacio en blanco para el signo + antes de los valores numéricos:

`15+ 30=?`

Este es un buen ejemplo de la utilidad de la función STR\$.

Referencia VAL cumple la función inversa a STR\$.

LEFT\$

| | |
|-----------------|---|
| Función | Toma el número especificado de caracteres a partir de la izquierda de una secuencia de caracteres. |
| Formatos | LEFT\$ ("secuencia de caracteres", expresión numérica) LEFT\$ (variable de caracteres, expresión numérica) |

LEFT\$ es una función que toma el número de caracteres especificado a partir de la izquierda de una secuencia de caracteres.

Ejemplo

```

10 AB$="LEFT RIGHT" ... Como en este caso se asigna una
20 B$=LEFT$(AB$,4)      secuencia de caracteres con más
30 PRINT B$              de 6 caracteres, debe usarse una
                          variable registrada, la cual acepta
                          hasta 16 caracteres.

```

Al ejecutarse el programa, aparece en la pantalla la indicación "LEFT" (izquierda). En otras palabras, se toman cuatro caracteres a partir de la izquierda de la secuencia de caracteres contenida en AB\$.

Cuando se especifica un 0 como el número de caracteres a tomarse, no se obtiene ningún carácter; en cambio, cuando se especifica un número mayor que el número de caracteres que compone la secuencia, se toman todos los caracteres de la misma. El error BS aparece siempre que se intente especificar un número de caracteres mayor que 255.

El número de caracteres que han de tomarse puede especificarse por medio de una variable o una expresión numérica.

EJEMPLO DE PROGRAMA

* Veamos a continuación un programa cuya longitud de la secuencia visualizada aumenta de a un carácter:

```

10 AB$="READ LEFT"
20 N=LEN(AB$)
30 FOR I=1 TO N
40 BC$=LEFT$(AB$, I)
50 PRINT BC$
60 NEXT I
70 END

```

Ejemplo de ejecución

```

R
RE
REA
READ
READ
READ L
READ LE
READ LEF
READ LEFT

```

VER RIGHT\$

RIGHT\$

| | |
|-----------------|--|
| Función | Toma el número especificado de caracteres a partir de la derecha de una secuencia de caracteres. |
| Formatos | RIGHT\$("secuencia de caracteres,, , expresión numérica) RIGHT\$(variable de caracteres expresión numérica) |

La función RIGHT\$ toma el número especificado de caracteres a partir de la derecha de una secuencia de caracteres.

Ejemplo

```
10 AB$ = "LEFT RIGHT" . . . . Como en este caso se asigna una
20 B$ = RIGHT$(AB$,5)          secuencia de caracteres con más
30 PRINT B$                     de 6 caracteres, debe usarse una
                                variable registrada, la cual acepta
                                hasta 16 caracteres.
```

Al ejecutarse este programa, el mensaje RIGHT visualizado en la pantalla indica que se tomaron cinco caracteres a partir de la derecha de la secuencia de caracteres contenida en AB\$. Cuando se especifica 0 como el número de caracteres que han de tomarse, no se obtiene ningún carácter; en cambio, si el número especificado es mayor que el número de caracteres que compone la secuencia, se toman todos los caracteres. El error 65 aparece siempre que se intente especificar un número de caracteres mayor que 255. El número de caracteres que ha de tomarse puede especificarse ya sea por una variable como por una expresión numérica.

EJEMPLO DE PROGRAMA

He a continuación un programa que inserta una secuencia de caracteres dentro de otra.

```
10 AB$="AM PM"
20 BC$="NOON"
30 CD$=LEFT$(AB$,2)
40 CD$=CD$+" "+BC$
50 CD$=CD$+RIGHT$(AB$,3)
60 PRINT CD$
70 END
```

Resultado

AM NOON PM

En este programa, se inserta la secuencia de caracteres en BC\$ en la secuencia de caracteres AB\$, usando las funciones LEFT\$ y RIGHT\$.

VER LEFT\$

MID\$

| | |
|-----------------|---|
| Función | Toma el número de caracteres especificado a partir de la posición especificada contando desde la izquierda de una secuencia de caracteres. |
| Formatos | MID\$ ("expresión de caracteres", expresión numérica 1, expresión numérica 2) MID\$ (variable de caracteres, expresión numérica 1, expresión numérica 2) |

La función MID\$ toma el número de caracteres especificado a partir de la posición especificada contando desde la izquierda de una secuencia de caracteres. En esta función, se combinan LEFT\$ y RIGHT\$.

MID\$ (A\$, 3 , 2)

La expresión de arriba indica que se tomarán dos caracteres de A\$. Estos dos caracteres son el tercero y cuarto contando desde la izquierda de la secuencia de caracteres en de A\$.

Ejemplo

```

10 CLEAR
20 DIM A$(0)*20
30 A$(0)="LEFT_CENTER_RIGHT"
40 B$=MID$(A$(0),6,6)
50 PRINT B$
60 END

```

..... Se pueden asignar hasta 20 caracteres a una matriz de caracteres. (ver la página 162)

Al ejecutarse este programa, se toma una secuencia de 6 caracteres comenzando por el sexto carácter desde la izquierda de A\$(0) y aparece en la pantalla la indicación CENTER.

Cuando se utiliza la función MID\$, deben tomarse la siguientes precauciones.

Cuando se entra MID\$ (expresión de caracteres, n,m):

- (1) Se descartan los decimales que puedan tener n y m.
- (2) No se toma carácter alguno cuando m es igual a cero.

- (3) Cuando se omite “,m”, se obtienen todos los caracteres a partir del carácter especificado por “n”.
- (4) Cuando m es mayor que el número de caracteres a partir de la posición especificada hasta el final de la secuencia, se obtienen todos los caracteres a partir de dicha posición dada.
- (5) Cuando n es mayor que la cantidad total de caracteres que compone la secuencia, no se obtiene ningún carácter.
- (6) Para n y m pueden usarse variables y expresiones numéricas.
- (7) El mensaje de error BS aparece siempre que se exceden los límites siguientes:

$$1 \leq n < 256 \quad \text{y} \quad 0 \leq m < 256$$

EJEMPLO DE PROGRAMA

He aquí un programa que cuenta el número de letras “r” en una secuencia de caracteres.

```

10 DIM A$(0)*50
20 N=0
30 INPUT "DATA=";A$(0)
40 M=LEN(A$(0))
50 FOR I=1 TO M
60 IF MID$(A$(0),I,1)="r" THEN N=N+1
70 NEXT I
80 PRINT N
90 END

```

En este programa, se entran caracteres y se cuenta y visualiza el número de letras “r” minúsculas que haya dentro del texto entrado.

Pueden entrarse hasta 50 caracteres, incluyendo los espacios.

Por ejemplo, si se entrara la siguiente frase:

Learning to master your CASIO Personal Computer

se cuenta el número de letras “r” desde el primer hasta el último carácter de la secuencia. En este caso, el número visualizado sería “5”. Pruébalo Vd. mismo.

LEFT\$, RIGHT\$

LEN

| | |
|-----------------|--|
| Función | Determina la longitud (número de caracteres) de una secuencia de caracteres. |
| Formatos | LEN ("secuencia de caracteres") LEN (variable de caracteres) |

LEN es una función que determina la longitud de una secuencia de caracteres asignada a una variable de caracteres.

Ejemplo

Si se entrara:

```
CLEAR ↵
PRINT LEN (A$) ↵
```

se visualiza "0".

Ello sucede ya que el mando CLEAR repone (lleva a su condición inicial) el contenido de todas las variables (incluyendo, en este caso, A\$).

```
AB$ = "CASIO _ COMPUTER" ↵
PRINT LEN (AB$) ↵
```

En este ejemplo, se visualiza el número "14".

La función LEN pueden determinar la longitud de secuencias de 0 hasta 79 caracteres.

EJEMPLO DE PROGRAMA

* Visualiza secuencias de caracteres alineadas por el margen derecho.

```
10 INPUT AB$
20 L=20-LEN(AB$)
30 LOCATE L,3
40 PRINT AB$;:LOCATE 0,0
50 END
```

Al entrarse una secuencia de caracteres en este programa, la misma se visualiza alineada con la última posición (primera de la izquierda) de la pantalla. 16 es el número máximo de caracteres que pueden entrarse por línea.

INKEY\$

| | |
|---------|---|
| Función | Permite la entrada de un carácter desde el teclado. |
| Formato | INKEY\$ |

La función INKEY\$ es un mando de entrada similar a INPUT, con ciertas diferencias en su operación. Con INKEY\$, puede entrarse un carácter (correspondiente a la pulsación de una tecla) por vez. En caso de no pulsarse ninguna tecla, no se entra carácter alguno y la ejecución continúa hacia el siguiente mando. Para esta función, no es necesario presionar la tecla  para entrar la información.

| Aplicación | INKEY\$ | INPUT |
|------------------------------------|---|--|
| Visualización durante su ejecución | No se visualiza nada | Mensaje de solicitud en entrada. Es posible hacer que no aparezca el signo "?". |
| Entrada de la información | Pulsación de un tecla durante su ejecución (no entra información alguna si no se presiona ninguna tecla). | Se entra la información cuando se pulsa la tecla  . |
| Tipos de datos | Un carácter por vez (todas las entradas consideradas caracteres). | Caracteres o valores numéricos dentro de los límites dispuestos por la variable utilizada. |
| Ejecución del siguiente mando | Inmediata (no es necesario pulsar ) | Después de pulsada la tecla  . |

Ejemplo

En vista de que la función INKEY\$ permite la entrada de un carácter por vez, la misma puede sólo asignarse a una variable de caracteres:

Variable de caracteres = INKEY\$

```

100 A$=INKEY$
110 IF A$="" THEN 100
120 IF A$="E" THEN END
130 .....

```

El programa termina cuando se presiona la tecla "E". La ejecución pasa a los mandos siguientes. En caso de no pulsarse tecla alguna, el programa va y vuelve entre las líneas 100 y 110.

INKEY\$ acepta todas las teclas, excepto **BRK** , **SHIFT** , **CAPS** y **F** . La pulsación de **SHIFT** o **CAPS** junto con otra tecla permite la entrada en el modo para caracteres especiales y mayúsculas. Sin embargo, no se toma en cuenta la entrada de mandos de una tecla.

EJEMPLO DE PROGRAMA

* Esta subrutina asigna un número determinado de caracteres a una variable de caracteres.

```

10 REM INKEY$ EXAMPLE
20 CLEAR 6000:AB$=""
30 FOR I=1 TO 10
40 K$=INKEY$
50 IF K$="" THEN 40
60 IF K$="*" THEN 90
70 AB$=AB$+K$
80 NEXT I
90 PRINT AB$

```

El número de caracteres que pueden asignarse a una variable de caracteres es limitado.

Si la longitud de la secuencia de caracteres entrada mediante una sentencia INPUT excede dicho límite, aparece entonces un error ST. No obstante, en este programa, una vez entrados 10 caracteres, no se acepta ninguna otra entrada y el programa avanza hacia el siguiente mando (línea 90).

Es más, la entrada puede interrumpirse antes de entrado el décimo carácter, mediante la entrada de un asterisco. Al hacerlo, el programa pasa a la línea siguiente (línea 90).

VER INPUT

DMS\$

| | |
|----------------|--|
| Función | Convierte valores decimales en sexagesimales y los expresa como una secuencia de caracteres. |
| Formato | DMS\$ (expresión numérica) |

Esta función convierte valores decimales en sexagesimales y los expresa como una secuencia de caracteres. Los límites para la expresión numérica son los siguientes:

$$|\text{Expresión numérica}| < 10^{100}$$

Por otro lado, cuando la expresión numérica $\geq 1E6$, los minutos y los segundos no aparecen en la pantalla (se convierte el valor entrado en una secuencia de caracteres).

EJEMPLO DE PROGRAMA

En el siguiente programa, se entran varios valores decimales para ver cómo se convierten.

```

10 REM DMS$ EXAMPLE
20 INPUT "NUMBER = ", A
30 PRINT "DMS$( "; A; )"
40 PRINT "= "; DMS$(A)
50 GOTO 20

```

DEG

HEX\$

| | |
|----------------|--|
| Función | Convierte un valor decimal en hexadecimal y lo expresa como una secuencia de caracteres. |
| Formato | HEX\$ (expresión numérica) |

Esta función convierte un valor decimal en hexadecimal y lo expresa como una secuencia de caracteres. La expresión numérica debe estar dentro de los siguientes límites. Se descarta toda parte decimal.

$$-32769 < \text{expresión numérica} < 65536$$

La secuencia de caracteres obtenida es un valor hexadecimal de 4 dígitos. Los números negativos se expresan como complementos de dos. Cuando un número es igual o mayor que 32768, se le resta 65536 y se convierte el resultado en un valor hexadecimal.

Uso

En el programa siguiente, los valores decimales entrados se convierten en hexadecimales y aparecen en la pantalla como tales. Entre algunos valores para probar que el programa funciona correctamente.

```

10 REM HEX$ EXAMPLE
20 INPUT "NUMBER= ", A
30 PRINT "HEX$(" ; A ; ")";
40 PRINT "= &H"; HEX$(A)
50 GOTO 20

```

VER

&H

4-5 FUNCIONES DE VISUALIZACION

TAB

| | |
|---------|---|
| Función | Desplaza el cursor un número especificado de posiciones hacia la derecha. |
| Formato | TAB (expresión numérica) $0 \leq \text{expresión numérica} < 80$ |

Esta función se utiliza con las sentencias PRINT y LPRINT para desplazar el cursor horizontalmente hasta la posición deseada.

Ejemplo

```
10 FOR X=1 TO 5
20 PRINT X; "^2:";
30 PRINT TAB(10);X^2
40 NEXT X
```

La ejecución de este programa visualiza lo siguiente:

△1^2: _____△1

△2^2: _____△4

⋮

TAB(10)

△ indica la posición donde debería estar el signo +.

TAB (10), por ejemplo, haría que el cursor avance 10 posiciones en la pantalla antes de visualizar la información. Esta función sirve para la visualización de variables y expresiones numéricas, y puede especificarse de 0 hasta 79. Los números a continuación del punto decimal se omiten.

EJEMPLO DE PROGRAMA

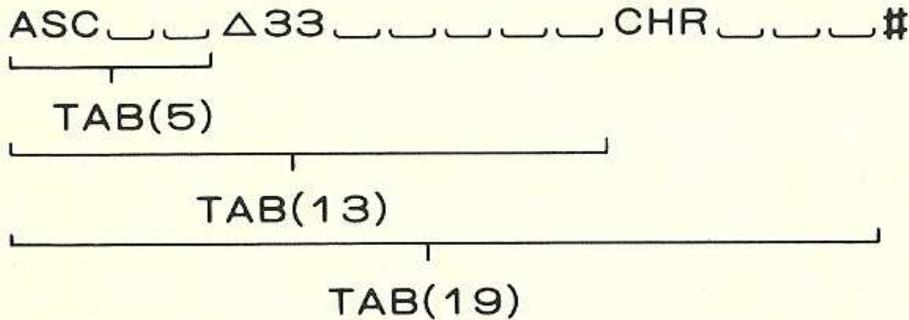
Visualiza el carácter correspondiente a un código ASCII dado en la posición designada.

```
10 FOR I=33 TO 254
20 PRINT "ASC";
30 PRINT TAB(5);I;
```

```

40 PRINT TAB(13); "CHR";
50 PRINT TAB(19); CHR$(I)
60 NEXT I
70 END
    
```

La ejecución de este programa visualiza lo siguiente:



La pulsación de la tecla  hace que se visualice el siguiente código y el carácter que le corresponde en la misma posición.

Posiciones en la pantalla

1. A partir de la primera posición de la izquierda de la primera línea, cuente 1 por cada espacio que se avanza hacia la derecha.
2. Cuando se especifica una posición de tabulado hacia la izquierda de la posición en curso, la posición especificada se determina contando a partir de la primera posición de la línea siguiente.
3. Cuando se especifica una posición que excede la capacidad de la línea, la posición especificada se determina contando el excedente a partir del comienzo de la línea en curso.

NOTA:

Cuando se utiliza TAB junto con el mando LPRINT, la impresora FA-10 ó FA-11 puede ejecutar la función TAB normalmente; ello no sucede, sin embargo, cuando se utilizan impresoras CENTRONICS conectadas a través de la interface FA-4. Cuando se ejecuta una función TAB, se emiten a la impresora los siguientes datos:

(1B)_H + (54) + (valor de expresión numérica) + (0D)_H

* ()_H representa un valor hexadecimal.

Cuando se usan impresoras especiales donde no pueda utilizarse TAB directamente, use el código de función de tabuladores horizontales correspondiente a la misma.

VER USING, PRINT, LPRINT, LOCATE

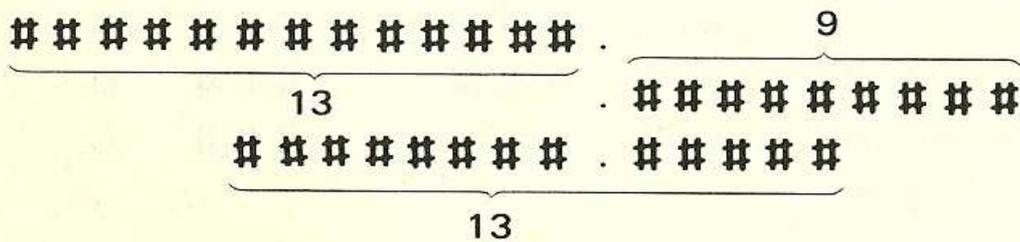
La ejecución de este programa visualiza lo siguiente:

CASIO└COMPUTER└└!!

El número de caracteres en la frase "CASIO└COMPUTER" es 14, pero como el número de caracteres de formato es de 16, los dos caracteres que restan aparecen como espacios en blanco.

En el uso de la sentencia USING, deben tenerse en cuenta las siguientes precauciones:

- (1) El formato no puede especificarse por caracteres que no sean #, ., ^ o &.
 - (2) #, ., ^ y & no pueden usarse juntos.
 - (3) Especificación del formato numérico:
 - # Número de dígitos
 - Posición del punto decimal
 - ^ Exponente
- (a) # puede especificar hasta 13 dígitos en la parte entera y 9 dígitos en la parte decimal. Los dos juntos pueden especificar hasta 13 dígitos.

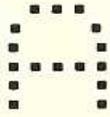


- (b) La posición del signo menos debe también designarse por medio de #.
- (c) ^ se especifica al final de la secuencia de caracteres para formato.
- (d) Cuando la parte decimal excede el límite, la visualización se lleva a cabo redondeando el dígito que sigue al especificado.
 PRINT USING "###.###"; 12.3456 → 12.346
- (e) Cuando la parte entera excede el límite, se coloca al comienzo el símbolo % para permitir la visualización sin seguir el formato especificado.
 PRINT USING "##.##"; 1234.56 → %└1234.56
- (f) La visualización de los valores numéricos es alineada con el borde derecho de la pantalla.

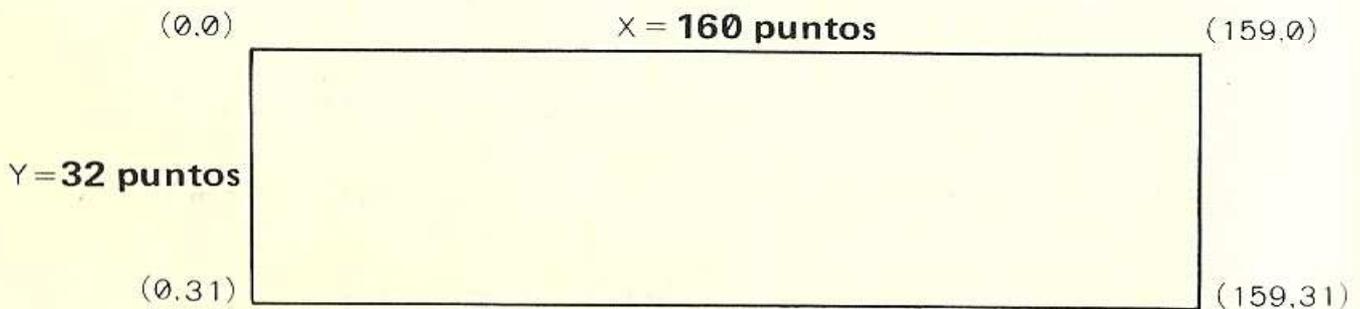
POINT

| | |
|----------------|---|
| Función | Verifica si está encendido o no un punto determinado de la pantalla. |
| Formato | POINT (X, Y) $0 \leq X \leq 159$ (posiciones horizontales) $0 \leq Y \leq 31$ (posiciones verticales) |

La pantalla consiste en pequeños puntos, por medio de cuya combinación se forman los caracteres. La letra A, por ejemplo, sería del siguiente modo:



La pantalla se compone, así, de 5.120 puntos.



Se adoptan de este modo dos ejes de coordenadas, uno vertical (Y) y otro horizontal (X), para especificar la posición de cada punto:

Coordenada (X, Y)

La función POINT verifica entonces si está encendido o no el punto en la posición (X, Y), devolviendo un 0 (cero) cuando sí lo está y un 1 (uno) en caso contrario.

Ejemplo

```
10 X=0:Y=0
20 A=POINT(X,Y)
30 PRINT A
```

Al ejecutarse este programa, si el punto en $(0, 0)$ está encendido, se visualiza un "1"; de lo contrario, aparece en la pantalla un "0".

En el uso de esta función, deberán considerarse las siguientes precauciones:

- (1) Las coordenadas X e Y no aceptan decimales. Todos los decimales son redondeados al número entero más cercano.
- (2) Siempre que el valor de X o Y exceda los límites especificados, aparece un error BS.

EJEMPLO DE PROGRAMA

* Programa para simular gráficamente un rayo laser.

```

10 100 CLS : AA$="2499DBFFFFDB9924": BB$="
      00808080C1E3E3F7"
20 110 X=INT(RND*7)+7
30 120 LOCATE X,0:PRINT $AA$;:LOCATE 10,3
      :PRINT $BB$;
40 130 N$=INKEY$
60 140 IF N$=" " THEN GOSUB 300
60 150 LOCATE X,0:PRINT " ";
70 160 GOTO 110
80 300 DRAW(83,24)-(83,7)
90 310 A=POINT(83,2)
100 320 IF A=1 THEN LOCATE 0,3:PRINT "BEE!"
      "":BEEP
110 330 DRAWC(83,24)-(83,7)
120 340 FOR I=0 TO 30:NEXT I
130 350 CLS :RETURN

```

4-6 MANDOS/FUNCIONES ESTADISTICAS

STAT

| | |
|----------------|---|
| Función | Permite la entrada de datos estadísticos. |
| Formato | STAT [valor del dato x] [, valor del dato y] [; frecuencia] |

El mando STAT se emplea del siguiente modo para la entrada de datos estadísticos.

(1) STAT $x ; n$

Este formato es para la entrada de datos con una sola variable; n es igual a 1 cuando no se especifica lo contrario.

(2) SATA $x, y ; n$

Este formato es para la entrada de pares de variables; n es igual a 1 cuando no se especifica lo contrario. Siempre que se omita el valor de cualquiera de las dos variables, x o y , se utiliza el valor entrado previamente para ellas (función de repetición). No obstante, no se permite la omisión de ambos valores a la vez.

Uso

Ver 3-18 "FUNCIONES ESTADISTICAS"

VER STAT CLEAR' STAT LIST

STAT CLEAR

| | |
|----------------|--|
| Función | Borra los registros utilizados para cálculos estadísticos. |
| Formato | STAT CLEAR |

Este mando borra los registros utilizados para cálculos estadísticos. Con el mismo, los contenidos de CNT, SUMX, SUMY, SUMX2 y SUMXY se transforman en 0.

Este mando debe usarse siempre antes de comenzar cálculos estadísticos nuevos.

 STAT

STAT LIST/STAT LLIST

| | |
|-----------------|---|
| Función | Visualiza/imprime estadísticas básicas. |
| Formatos | STAT LIST STAT LLIST |

(1) STAT LIST

Visualiza los nombres de los datos y los valores para las estadísticas básicas en el orden CNT, SUMX, SUMY, SUMXY, SUMX2 y SUMY2.

La presentación en la pantalla puede interrumpirse mediante la pulsación de la tecla . Una segunda pulsación la reanuda.

(2) STAT LLIST

Igual que STAT LIST, sólo que en lugar de la pantalla, se imprime todo en la impresora.

 STAT

CNT

| | |
|----------------|--|
| Función | Da el número de datos procesados estadísticamente (n). |
| Formato | CNT |

La función CNT da el número de datos procesados estadísticamente.

Referencia

Para mayor información sobre el uso de las siguientes funciones estadísticas, remítase a la sección 3-18.

COR

| | |
|----------------|--|
| Función | Determina el coeficiente de correlación (r). |
| Formato | COR |

COR determina el coeficiente de correlación (r) como una expresión numérica.

$$\text{COR} : \frac{n \cdot \sum xy - \sum x \cdot \sum y}{\sqrt{\{n \cdot \sum x^2 - (\sum x)^2\} \{n \cdot \sum y^2 - (\sum y)^2\}}}$$

(n: número de datos)

SUMX

SUMY

SUMX2

SUMY2

SUMXY

| | |
|------------------|--|
| Funciones | SUMX: Determina la suma de los datos x . SUMY: Determina la suma de los datos y . SUMX2: Determina la suma de los cuadrados de x . SUMY2: Determina la suma de los cuadrados de y . SUMXT: Determina la suma de los productos de x e y . |
| Formatos | SUMX SUMY SUMX2 SUMY2 SUMXY |

Estas funciones permiten obtener sumas, sumas de los cuadrados y sumas de los productos.

$$\text{SUM X} : \Sigma x$$

$$\text{SUM Y} : \Sigma y$$

$$\text{SUM X 2} : \Sigma x^2$$

$$\text{SUM Y 2} : \Sigma y^2$$

$$\text{SUM X Y} : \Sigma xy$$

MEANX MEANY

| | |
|------------------|--|
| Funciones | MEANX: Determina la media de los datos x . MEANY: Determina la suma de los productos de x e y . |
| Formatos | MEANX MEANY |

Estas funciones permiten obtener la media de los datos.

$$\text{MEANX} : \Sigma x/n$$

$$\text{MEANY} : \Sigma y/n$$

SDX SDY SDXN SDYN

| | |
|------------------|--|
| Funciones | SDX: Determina la desviación estándar por muestreo de los datos x . SDY: Determina la desviación estándar por muestreo de los datos y . SDXN: Determina la desviación estándar total de los datos x . SDNY: Determina la desviación estándar total de los datos y . |
| Formatos | SDX SDY SDXN SDYN |

Estas funciones permiten obtener desviaciones estándar por muestreo y total a modo de expresiones numéricas, del modo que se muestra a continuación.

$$\text{SDX} : \sqrt{\frac{n \cdot \sum x^2 - (\sum x)^2}{n(n-1)}} \quad (x\sigma_{n-1})$$

$$\text{SDY} : \sqrt{\frac{n \cdot \sum y^2 - (\sum y)^2}{n(n-1)}} \quad (y\sigma_{n-1})$$

$$\text{SDXN} : \sqrt{\frac{n \cdot \sum x^2 - (\sum x)^2}{n^2}} \quad (x\sigma_n)$$

$$\text{SDYN} : \sqrt{\frac{n \cdot \sum y^2 - (\sum y)^2}{n^2}} \quad (y\sigma_n)$$

(n : número de datos)

EOX EOY

| | |
|------------------|--|
| Funciones | EOX: Determina el valor estimado de x en términos de y . EOY: Determina el valor estimado de y en términos de x . |
| Formatos | EOX: expresión numérica EOY: expresión numérica |

Estas funciones permiten obtener valores estimados a modo de expresiones numéricas.

$$EOX(y) : \frac{y - LRA}{LRB} \quad (\hat{x})$$

$$EOY(x) : LRA + x \cdot LRB \quad (\hat{y})$$

LRA LRB

| | |
|------------------|--|
| Funciones | LRA: Determina el término constante de regresiones lineales. LRB: Determina el coeficiente de regresión lineal. |
| Formatos | LRA LRB |

Estas funciones permiten obtener el término constante de regresiones lineales y el coeficiente de regresión lineal.

$$LRA : \frac{\sum y - LRB \cdot \sum x}{n}$$

$$LRB : \frac{n \cdot \sum xy - \sum x \cdot \sum y}{n \cdot \sum x^2 - (\sum x)^2}$$

(n : número de datos)

4-7 MISCALEANOS

&H

| | |
|----------------|--|
| Función | Convierte un valor hexadecimal (hasta 4 dígitos) en decimal. |
| Formato | &H valor hexadecimal |

Esta función convierte valores hexadecimales de hasta cuatro dígitos en decimales. Veamos a continuación algunos ejemplos.

| Hexadecimal | | Decimal |
|-------------|--------------------------------------|---------|
| &H1 | <input type="button" value="ENTER"/> | 1 |
| &HA | <input type="button" value="ENTER"/> | 10 |
| &H000B | <input type="button" value="ENTER"/> | 11 |
| &HABCD | <input type="button" value="ENTER"/> | -21555 |
| &HG | <input type="button" value="ENTER"/> | Error |
| &H12345 | <input type="button" value="ENTER"/> | Error |

&H no es una función del lenguaje BASIC.

Uso

(1) Cálculos manuales

Entre un número hexadecimal luego de &H y pulse la tecla .

Ejemplo: &H1B7F → 7039

(2) Programa

Veamos a continuación un programa de aplicación. Como a continuación de &H no puede usarse una variable, se le adjunta una secuencia de caracteres hexadecimal, y luego se usa la función VAL para convertirla en un valor decimal.

```
10 REM &H EXAMPLE
20 INPUT "&H";A$
30 H=VAL("&H"+A$)
40 PRINT "&H ";A$;"=";H
50 GOTO 10
```

 HEX\$, VAL

EOX

FTM

Handwritten text, possibly a name or title.

Handwritten text, possibly a signature or name.

CAPITULO 5

LISTA DE PROGRAMAS

ATENCIÓN:

Los programas de este capítulo pueden usarse libremente, sin necesidad de permiso alguno. Sin embargo, se da por entendido que la compañía no se hace responsable por cualquier daño o pérdida que pudiera resultar al usar estos programas de ejemplos.

En caso de que se ejecuten programas de este capítulo sin usar la impresora por coordenadas opcional (A-10 ó FA-11), oprima la tecla "N" cuando la unidad PB-770 pregunte si se harán o no impresiones, al aparecer en la pantalla "PRINTER ON? (Y/N)". (IMPRESORA ACTIVADA? (SI/NO)).

ADMINISTRACION DE ACCIONES Y PRECIOS ADECUADOS DE COMPRA/VENTA

Este programa almacena los precios de acciones de las últimas 53 semanas. Después de entrados los datos de las 53 semanas, cada vez que se introduzcan los datos de una nueva semana, se eliminarán los datos de la semana más remota. En base a los datos de los precios de las acciones, el programa proporciona el valor de desviación en curso e información útil para evaluar las condiciones de compra o venta. El programa también exhibe en la pantalla el valor de desviación y del promedio móvil, así como las fluctuaciones del precio de las acciones en gráficas.

Explicación

Primero, comience el programa en P0 y el menú aparecerá en la pantalla. Luego, introduzca un número adecuado del 1 al 7 que ofrece el menú. Si introduce otro número que no sea del 1 al 7, el menú aparecerá nuevamente en la pantalla.

(1) Introducción de datos

Para introducir datos, oprima después de que se exhiba el menú en la pantalla, y entonces aparecerá "Initial? (Y/N)". Si usted entra datos por primera vez, oprima la tecla . Si usted ya había introducido datos antes de esta ocasión, entonces oprima la tecla . Cuando se oprime la tecla , "CLEAR OK? (Y/N)" aparece en la pantalla. Esto es para evitar que los datos se pierdan por alguna equivocación al introducirlos. Normalmente, oprima la tecla .

Para la introducción inicial de datos, "1) DATA =" aparece en la pantalla. Introduzca el precio de las acciones correspondientes y oprima la tecla . Para la segunda introducción de datos y las subsecuentes, "WEEK=" aparecerá en la pantalla. Al introducir datos para la 54ª semana y las subsecuentes, los datos más remotos se van eliminando subsecuentemente. Por tanto, el tiempo requerido para la introducción se hará un poco más largo. Para terminar la introducción de datos, introduzca un número negativo y entonces el menú aparecerá nuevamente en la pantalla.

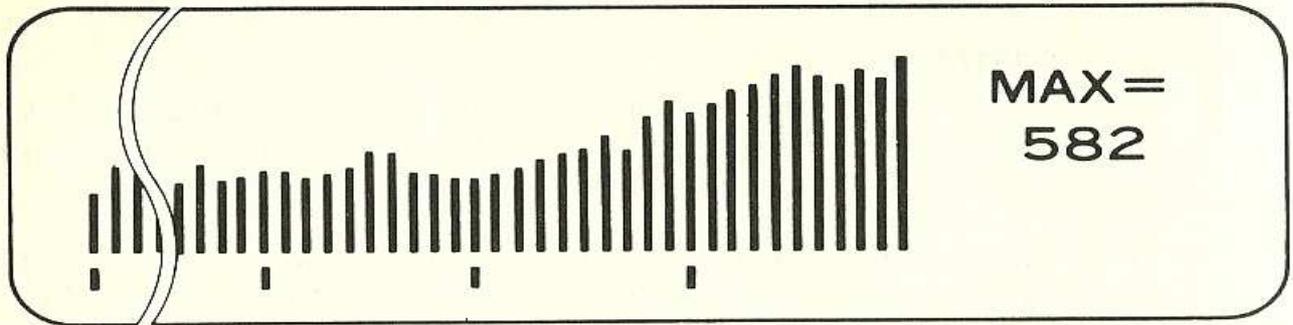
(2) Evaluación de las condiciones para compra o venta

Cuando se selecciona el menú 2, "Current Price?" aparece en la pantalla. En ese momento, introduzca el precio en curso de las acciones, entonces el programa comparará el precio de las acciones en curso con los datos anteriores y proporcionará el valor de la desviación. Para salirse de esta rutina, introduzca un valor negativo. El menú aparecerá nuevamente en la pantalla.

(3) Verificación del precio razonable de las acciones

Si se han introducido nuevos datos usando esta rutina, la rutina (3) deberá ejecutarse antes para que puedan obtenerse los valores correctos.

Ejemplo de visualización de una gráfica



Programa

P0

```

10 CLS
20 GOSUB 80
30 INPUT "INPUT NO. "; PR
40 IF PR > 7 THEN 10 ELSE IF PR < 1 THEN
10
50 IF PR = 1 THEN GOSUB PROG 1 ELSE IF
PR = 2 THEN GOTO PROG 2
60 IF PR = 3 THEN GOSUB PROG 3 ELSE IF
PR = 4 THEN GOTO PROG 4
70 IF PR = 5 THEN GOSUB PROG 5 ELSE IF
PR = 6 THEN GOTO PROG 6 ELSE GOTO PR
OG 7
80 PRINT :PRINT "DATA INPUT 1"
90 FOR I=1 TO 100:NEXT I
100 PRINT "PRICE CHECK 2"
110 FOR I=1 TO 100:NEXT I
120 PRINT "REASONABLE PRICE 3"
130 FOR I=1 TO 100:NEXT I
140 PRINT "DATA OUTPUT 4"
150 FOR I=1 TO 100:NEXT I
160 PRINT "MOVING AVE. 5"
170 FOR I=1 TO 100:NEXT I
180 PRINT "PAST MOVEMENT 6"
190 FOR I=1 TO 100:NEXT I
200 PRINT "GRAPH DISPLAY 7"
210 RETURN

```

P1

```

10 CLS
20 PRINT " ** DATA INPUT **"
30 INPUT "Initial?(Y/N)",P$: IF P$="Y"
    THEN GOSUB 200
40 IF P$="N" THEN A=A-1 ELSE IF P$(">")
    Y" THEN 30
50 A=A+1:GOSUB 300
60 IF A>=53 THEN 80
70 INPUT "DATA";Z(A): IF Z(A)<0 THEN Z
    (A)=0:GOTO PROG 0 ELSE GOTO 50
80 INPUT "DATA+";DZ:C=C+1:A=53: IF DZ<
    0 THEN GOTO PROG 0 ELSE Z(53)=DZ
90 FOR B=1 TO 53
100 Z(B-1)=Z(B)
110 NEXT B
120 GOTO 80
200 INPUT "CLEAR OK?(Y/N)",T$: IF T$="Y"
    " THEN 220 ELSE IF T$="N" THEN 260
210 GOTO 200
220 ERASE Z: DIM Z(53)
230 PRINT "STOCK PRICE "
240 FOR K=0 TO 50: NEXT K
250 INPUT "1)DATA";Z(1): A=1
260 P$="Y": RETURN
300 PRINT "WEEK="; A
310 FOR K=0 TO 10: NEXT K: RETURN

```

P2

```

10 CLS
20 PRINT " ** PRICE CHECK **"
30 S=0: Q=0
40 FOR D=0 TO 52
50 S=S+Z(D): Q=Q+Z(D)^2
60 NEXT D
70 IF A<54 THEN 90
80 E=S/53: U=Q-53*E*E: F=SQR(U/52)
90 E=S/A: U=Q-A*E*E: F=SQR(U/(A-1))
100 INPUT "Current Price";Y: IF Y<0 THE
    N 140
110 D=ROUND(50+10*(Y-E)/F,-3)

```

```

120 PRINT "Deviation=";D
130 GOTO 100
140 GOTO PROG 0

```

P3

```

10 CLS
20 PRINT "**REASONABLE PRICE**"
30 INPUT "Deviation=";D:Y=ROUND((D-50
    )*(F/10+E,-2):IF D<0 THEN 50
40 PRINT " PRICE=";Y:GOTO 30
50 GOTO PROG 0

```

P4

```

10 CLS
20 PRINT " ** DATA OUTPUT **"
30 FOR U=0 TO A-2
40 PRINT "DATA";U+1;"=";Z(U+1)
50 FOR K=0 TO 50
60 NEXT K:NEXT U
70 PRINT "DATA END"
80 FOR K=1 TO 200:NEXT K
90 GOTO PROG 0

```

P5

```

10 CLS
20 PRINT " ** MOVING AVE. **"
30 X=0
40 INPUT "No. of movements";N
50 IF A<=53 THEN 100
60 FOR L=53-N TO 52
70 X=X+Z(L)
80 NEXT L
90 GOTO 130
100 FOR K=A-N TO A
110 X=X+Z(K)
120 NEXT K
130 M=X/N
140 PRINT "MOVING AVERAGE";M
150 FOR K=0 TO 300:NEXT K
160 GOTO PROG 0

```

P6

```

10 CLS
20 PRINT "** PAST MOVEMENT **"

```

```
30 INPUT "No. of movements"; I
40 INPUT "FROM WHEN "; O
50 X=0
60 IF A<=53 THEN 110
70 FOR J=53-O-I TO A-O-I
80 X=X+Z(J)
90 NEXT J
100 GOTO 140
110 FOR J=A-O-I TO A-O-1: IF A<=J THEN
180
120 X=X+Z(J)
130 NEXT J
140 M=X/I
150 FOR K=0 TO 100:NEXT K
160 PRINT "Movins Ave. "; M
170 O=O-1: X=0: FOR K=0 TO 10:NEXT K: GOT
O 60
180 PRINT " END"
190 FOR K=0 TO 300:NEXT K
200 GOTO PROG 0
```

```
P7
10 CLS
20 PRINT "** GRAPH DISPLAY **"
30 MX=0: FOR MD=1 TO 53
40 IF Z(MD)>MX THEN MX=Z(MD)
50 NEXT MD
60 CLS : LOCATE 15, 1: PRINT "MAX=": LOC
ATE 14, 2: PRINT INT(MX)
70 FOR K=1 TO 5
80 PO=130-K*20: DRAW(PO, 27)-(PO, 30)
90 NEXT K
100 FOR K=1 TO A-1
110 J1=K*2+10: J2=25-25/MX*Z(K)
120 DRAW(J1, J2)-(J1, 25)
130 NEXT K
140 IF INKEY$="" THEN 140 ELSE GOTO PR
OG 0
```

Número de bitios utilizado: 1,872

CAPITULO 5 LISTA DE PROGRAMAS

| Datos de ejemplo | | | | | | | |
|------------------|-----|-----------------|-----|----------------|-----|------------------|-----|
| Hace 19 semanas | 584 | Hace 14 semanas | 545 | Hace 9 semanas | 635 | Hace 4 semanas | 686 |
| Hace 18 semanas | 580 | Hace 13 semanas | 550 | Hace 8 semanas | 652 | Hace 3 semanas | 697 |
| Hace 17 semanas | 579 | Hace 12 semanas | 563 | Hace 7 semanas | 673 | Hace 2 semanas | 686 |
| Hace 16 semanas | 570 | Hace 11 semanas | 589 | Hace 6 semanas | 701 | La semana pasada | 672 |
| Hace 15 semanas | 562 | Hace 10 semanas | 620 | Hace 5 semanas | 692 | Esta semana | 689 |

| Contenido de las variables | | | | | |
|----------------------------|--|-----|--|-------|----------------------------|
| A | Contador de datos | J 2 | Eje de Y en la gráfica | PR | Selección del programa |
| B ~ D | Contadores | MD | Contador | Q | Suma de cuadrados de datos |
| DA | Datos en caracteres | MX | Dato máx. | S | Suma de datos |
| DZ | Precio de acciones | N | No. de movimientos (semanas) | T\$ | CLEAR OK? (Eliminación) |
| E | Promedio | O | Semana inicial para calcular el promedio móvil | U | Contador |
| H | Valor de desviación | P\$ | Y o N (Initial?) (Sí o No) | V | Variación |
| J ~ L | Contadores (de períodos para calcular el promedio móvil en P5) | P0 | Escala de la gráfica | X | Contador |
| J 1 | Eje de X en la gráfica | | | Z (X) | Precio de acciones |

Operación

| | Paso | Operación de las teclas | En pantalla |
|---|------|-------------------------|--|
| Exhibición del menú → | | SHIFT P0 | DATA INPUT 1 PRICE CHECK 2 REASONABLE PRICE 3 DATA OUTPUT 4 MOVING AVE. 5 PAST MOVEMENT 6 GRAPH DISPLAY 7 INPUT NO. ? _ |
| Selección de la introducción de datos. → | 1 | 1 ↵ | ** DATA INPUT ** Initial ? (Y/N) _ |
| Oprima [Y] para introducción inicial de datos y [N] para introducción adicional de datos. → | 2 | Y ↵ | CLEAR OK ? (Y/N) _ |
| Oprima [Y] para eliminar los datos almacenados y [N] para no eliminarlos. → | 3 | Y ↵ | STOCK PRICE 1) DATA ? _ |
| (1) Introducción de datos | 4 | 584 ↵ ⋮ | WEEK=2 DATA ? _ ⋮ |
| Introduzca un número negativo para terminar la introducción de datos. → | 5 | -1 ↵ | } INPUT NO. ? _ |
| (2) Verificación del precio de acciones | 6 | 2 ↵ | ** PRICE CHECK ** Current Price ? _ |
| Introduzca el precio de acciones en curso y entonces el valor de desviación aparece en la pantalla, en base a los datos anteriores. → | 7 | 585 ↵ | Deviation = 49.23 Current Price ? _ |
| Si introduce un número negativo, el menú se vuelve a exhibir. → | 8 | -1 ↵ | } INPUT NO. ? _ |
| (3) Precio de acciones razonable | 9 | 3 ↵ | ** REASONABLE PRICE ** Deviation = ? _ |
| Introduzca un valor de desviación. → | 10 | 55 ↵ | PRICE = 670.1 Deviation = ? _ |
| Si introduce un número negativo, el precio de las acciones se exhibe. → | 11 | -1 ↵ | } INPUT NO. ? _ |

CAPITULO 5 LISTA DE PROGRAMAS

| | Paso | Operación de las teclas | En pantalla |
|---|------|-------------------------|--|
| (4) Obtención de datos Todos los datos aparecen en la pantalla y luego vuelve a aparecer el menú. | 12 | 4 | **DATA OUTPUT** DATA 1=584 DATA 2=580 DATA 3=579 : DATA END } INPUT NO. ?_ |
| Exhibición del menú | | | |
| (5) Promedio móvil | 13 | 5 | **MOVING AVE.** No. of movements ? _ |
| Introduzca el número de semanas (movements). | 14 | 15 | MOVING AVERAGE 643.2 } INPUT NO. ?_ |
| (6) Promedio móvil en el pasado. Introduzca el número de semanas (movements) para las cuales se va a calcular el promedio móvil. | 15 | 6 | **PAST MOVEMENT** No. of movements ? _ |
| Introduzca la semana inicial. | 16 | 3 | FROM WHEN ? _ |
| Cada promedio móvil aparecerá en la pantalla y después aparecerá nuevamente el menú. | 17 | 2 | Moving Ave. 689 Moving Ave. 684.666..... Moving Ave. 682 END } INPUT NO. ?_ |
| (7) Gráfica | 18 | 7 | Una gráfica aparece en la pantalla |

DIRECTORIO TELEFONICO

Este programa le permite obtener inmediatamente un número telefónico requerido con la simple introducción de un nombre previamente almacenado. También puede obtener el número telefónico usando solamente la inicial de un nombre. Los nombres se pueden clasificar por orden alfabético.

Explicación

Este programa puede usarse para almacenar y visualizar números de teléfono de amigos y conocidos. Se puede inmediatamente obtener el número de teléfono deseado mediante este programa. Se pueden almacenar hasta 150 nombres cada vez. Una vez que los nombres y números de teléfono han sido almacenados, es posible obtener el número deseado con sólo introducir la letra inicial o las primeras dos o tres letras del nombres. Cuando se amplía la memoria a 16 o más kbytes mediante el uso de módulos RAM, se pueden almacenar hasta 255 nombres. En dicho caso, no obstante, es necesario hacer las siguientes modificaciones en el programa almacenado en P1.

(8KB)

```
30 IF N = 181 THEN 90
```

```
40 IF N = 1 THEN DIM A$(180), B$(180) * 12
```

(16KBo más)

```
30 IF N = 256 THEN 90
```

```
40 IF N = 1 THEN DIM A$(255), B$(255) * 12
```

Ejecute el programa en P0 y entonces el menú aparecerá en la pantalla. Introduzca un número adecuado, del 1 al 4. No necesita oprimir la tecla . En caso de aparecer un error OM (superación de la capacidad de la memoria) en la línea 40 de P1, borre otros programas o datos o amplíe la capacidad de la memoria usando módulos RAM.

(1) INTRODUCCION (INPUT)

Almacena los nombres y números telefónicos. Introduzca los datos de la siguiente manera. Los datos que se van a introducir aparecen subrayados.

NAME? (NOMBRE) CASIO 

TEL NO.? 123-4567 

Introduzca todos los nombres y números telefónicos repitiendo estos pasos. Después de introducir el último nombre y número telefónico, introduzca END y oprima la tecla . El menú vuelve a aparecer en la pantalla.

Programa

```

P0
  10 CLS
  20 PRINT "1-INPUT  2-SORTING"
  30 PRINT "3-LOOK FOR"
  40 PRINT "4-DELETE"
  50 K$=INKEY$:IF K$="" THEN 50
  60 IF K$="1" THEN GOTO PROG 1
  70 IF K$="2" THEN GOTO PROG 2
  80 IF K$="3" THEN GOTO PROG 3
  90 IF K$="4" THEN GOTO PROG 4
 100 GOTO 50
P1
  10 CLS
  20 N=N+1
  30 IF N=181 THEN 90
  40 IF N=1 THEN DIM A$(180),B$(180)*12
  50 INPUT "NAME  ":A$(N)
  60 IF A$(N)="END" THEN 100
  70 INPUT "TEL NO. ":B$(N)
  80 GOTO 10
  90 PRINT "FULL":BEEP 1
 100 N=N-1
 110 GOTO PROG 0
P2
  5  CLS :PRINT "SORTING..."
 10  FOR I=1 TO N
 20  MM$=A$(I):X=I
 30  FOR J=I TO N
 40  KK$=A$(J)
 50  GOSUB 200
 60  NEXT J
 70  A$(X)=A$(I):A$(I)=MM$
 80  MM$=B$(X)
 90  B$(X)=B$(I):B$(I)=MM$
100  NEXT I

```

```

110 GOTO PROG 5
200 KU=0
210 KU=KU+1
230 O1=LEN(MM$):O2=LEN(KK$)
240 IF KU>O1 THEN RETURN ELSE IF KU>O
    2 THEN 300
250 MI$=MID$(MM$,KU,1):KI$=MID$(KK$,KU
    ,1)
260 IF ASC(MI$)=ASC(KI$) THEN 210
270 IF ASC(MI$)>ASC(KI$) THEN X=J:MM$=
    KK$:RETURN
280 RETURN
300 X=J:MM$=KK$:RETURN

```

P3

```

100 10 CLS
120 20 INPUT "NAME ";MM$
140 30 X=LEN(MM$)
200 40 I=0
210 50 I=I+1
220 60 IF I=N+1 THEN IF F=1 THEN 120 ELSE
    130
230 70 IF MM$=LEFT$(A$(I),X) THEN 90
240 80 GOTO 50
250 90 F=1:PRINT A$(I)
260 100 PRINT B$(I)
270 110 K$=INKEY$:IF K$="" THEN 110 ELSE 5
    0
280 120 F=0:GOTO PROG 0
290 130 PRINT "NO DATA"
300 140 K$=INKEY$:IF K$="" THEN 140 ELSE 1
    20

```

P4

```

10 CLS
20 INPUT "NAME ";MM$
30 X=LEN(MM$)
40 I=0
50 I=I+1
60 IF I=N+1 THEN 180

```

```
70 IF MM$=LEFT$(A$(I),X) THEN 100
90 GOTO 50
100 PRINT A$(I)
110 PRINT B$(I); " Y/N ";
120 INPUT K$
130 IF K$="Y" THEN 150
140 GOTO 50
150 A$(I)=A$(N)
160 B$(I)=B$(N)
170 N=N-1
180 GOTO PROG 0
P5
10 FOR I=1 TO N
20 CLS
30 PRINT A$(I)
40 PRINT B$(I)
50 K$=INKEY$: IF K$="" THEN 50
60 NEXT I
70 GOTO PROG 0
```

Número de bitios utilizado: 1,003

Operación

| | Paso | Operación de las teclas | En pantalla |
|---|------|-------------------------|--|
| Exhibición del menú. → | | SHIFT PO 0 | 1-INPUT 2-SORTING 3-LOOK FOR 4-DELETE |
| (1) INTRODUCCION (INPUT) Seleccione el menú 1. → | 1 | 1 | NAME ? _ |
| Introduzca el nombre de la primera persona. → | 2 | SMITH, JOHN ↵ | NAME ? SMITH, JOHN TEL NO. ? _ |
| Introduzca el número telefónico. → | 3 | 03-583-4111 ↵ | NAME ? _ |
| Introduzca el nombre de la segunda persona. → | 4 | BROWN, MARY ↵ | NAME ? BROWN, MARY TEL NO. ? _ |
| Introduzca el número telefónico. → | 5 | 052-264-1453 ↵ | NAME ? _ ⋮ |
| Después de introducir todos los datos, introduzca END y oprima la tecla ↵; entonces el menú aparecerá en la pantalla. → | 6 | END ↵ | 1-INPUT 2-SORTING 3-LOOK FOR 4-DELETE |
| (3) BUSQUEDA (LOOK FOR) ¿El número telefónico de quién desea usted saber? → | 7 | 3 | NAME ? _ |
| Solamente puede introducir un apellido. → | 8 | SMITH, JOHN ↵ | NAME ? SMITH, JOHN SMITH, JOHN 03-583-4111 |
| El menú aparece en la pantalla. → | 9 | ↵ | 1-INPUT 2-SORTING 3-LOOK FOR 4-DELETE |
| (2) CLASIFICACION (SORTING) Orden alfabético → | 10 | 2 | SORTING... |
| Después de que "SORTING ..." desaparezca → | 11 | ↵ | ALLEN, ROBERT 06-314-2681 |
| Los nombres ordenados pueden aparecer en secuencia. → | 12 | ↵ | BROWN, MARY 052-264-1453 ⋮ |

Después de que todos los nombres y números telefónicos hayan aparecido en la pantalla, el menú aparecerá nuevamente.

(4) ELIMINACION (DELETE)

¿Qué nombre desea usted eliminar?

Si realmente desea eliminarlo, oprima **[Y]**; de no ser así, oprima **[N]**.

El menú aparece en la pantalla.

Verifique que los datos han sido eliminados usando el menú 3.

"NO DATA" indica que el nombre no ha sido almacenado.

| Paso | Operación de las teclas | En pantalla |
|------|---|--|
| 13 |  | 1-INPUT 2-SORTING 3-LOOK FOR 4-DELETE |
| 14 | [4] | NAME ? _ |
| 15 | SMITH, JOHN  | NAME ? SMITH, JOHN SMITH, JOHN 03-583-4111 Y/N ? _ |
| 16 | [Y] | 1-INPUT 2-SORTING 3-LOOK FOR 4-DELETE |
| 17 | [3] | NAME ? _ |
| 18 | SMITH, JOHN  | NAME ? SMITH, JOHN NO DATA |

| Contenido de las variables | |
|----------------------------|---|
| N | (Número de nombres y no. de tel) - 1 |
| X | Longitud de la serie de caracteres que se va a buscar |
| A\$ (1) A\$ (180) | Nombres |
| B\$ (1) B\$ (180) | Números telefónicos |
| MM\$ | Serie de caracteres que se va a buscar |
| KK\$ KU 01 02 | Para la clasificación |

Este programa obtiene sumas de datos horizontales (X) y verticales (Y), u ordena los datos para determinar el porcentaje de cada elemento de los datos. Por ejemplo, el ítem X puede ser cierto producto y el ítem Y cierto mes.

Explicación

Primero, ejecute el programa en P0 y entonces el siguiente menú aparecerá en la pantalla:

- | | | |
|---|----------------|---|
| 1 | DATA INPUT | ←Para introducir datos |
| 2 | TOTAL | ←Para obtener totales horizontales y verticales |
| 3 | SORT | ←Para ordenar los datos |
| 4 | DATA OUTPUT ?_ | ←Para verificar todos los datos |

(1) Introducción de datos

Cuando el menú 1 (DATA INPUT) se selecciona, "CLEAR (Y/N)?" aparece en la pantalla. Para introducir datos por primera vez, oprima . Entonces el programa le pedirá que introduzca valores para X y Y (el valor de X es el número de elementos de los datos en dirección horizontal, y el valor de Y es el número de elementos de los datos en dirección vertical). Introduzca los valores correspondientes.

Una vez que haya introducido los valores de X e Y, se le pedirá que introduzca los datos. Introduzca los datos de acuerdo con los números de elementos exhibidos en la pantalla:

X = 1, Y = 1 → X = 2, Y = 1 → X = 3, Y = 1 ...

X = 1, Y = 2 → X = 2, Y = 2 → X = 3, Y = 2 ...

| | X1 → | 2 | 3 | 4..... |
|----|------|---|---|--------|
| Y1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| ⋮ | | | | |

Una vez que todos los datos hayan sido introducidos, aparecerá en la pantalla primero el gran total y después el menú.

Si hay un error en los datos introducidos, vuelva a introducir 1. Cuando "CLEAR (Y/N)?" aparezca en la pantalla, introduzca N. Entonces el programa le preguntará sobre el número de elemento cuyos datos asociados se vayan a corregir. Introduzca el número del elemento correspondiente y enseguida "DATA?" aparecerá en la pantalla. Entonces introduzca los datos correctos. El gran total correcto aparecerá en la pantalla y luego se exhibirá el menú.

(2) Suma de los datos en X o Y

Cuando se selecciona el menú 2 (TOTAL), "PRINTER ON? (Y/N)" aparece en la pantalla. Los subtotales se imprimen al introducir Y. El programa le preguntará entonces si desea obtener la suma de X o de Y. Introduzca X o Y según sus requerimientos.

Cuando se introduce X, el programa proporciona los subtotales de X desde 1 hasta el valor establecido, y regresa a la exhibición del menú después de haber proporcionado el total. Cuando se introduce Y, el programa proporciona los subtotales de Y desde 1 hasta el valor establecido, y regresa a la exhibición del menú después de haber proporcionado el total.

(3) Ordenación

Cuando se selecciona el menú 3 (SORT), el programa le pregunta si los subtotales de X o de Y se van a ordenar después de que aparezca "PRINTER ON? (Y/N)" en la pantalla. Cuando se introduzca X o Y, "SORTING NOW" aparecerá en la pantalla y la operación de ordenación se iniciará. Los datos son arreglados en orden descendente y se exhiben junto con la clasificación, número de ítem, subtotal y porcentaje de cada elemento; luego se exhibe el menú. La operación de ordenación requiere cierto tiempo. Por ejemplo, requiere aproximadamente un minuto y 10 segundos para ordenar 20 elementos de datos. Cuando la operación de ordenación se termina, se produce un zumbido y empiezan a aparecer los datos. Una vez que este programa ha sido ejecutado, el menú vuelve a aparecer en la pantalla.

Como este programa usa muchas variables de precisión media, puede manejar un volumen relativamente grande de datos. No obstante, el número máximo de dígitos de los datos introducidos es cinco.

Para revisar el resultado de la ordenación de los datos, oprima la tecla **BRK** y ejecute la línea 140 del programa P4.

(4) Obtención de datos

Cuando se selecciona el menú 4 (DATA OUTPUT), aparecerá en la pantalla "X=1 Y=1 DATA=233", por ejemplo, después de aparecer "PRINTER ON? (Y/N)". Una vez que todos los datos hayan aparecido, el menú volverá a exhibirse en la pantalla.

Programa

P0

```

10 PRINT "1 DATA INPUT", "2 TOTAL", "3
    SORT", "4 DATA OUTPUT ";
20 INPUT R
30 IF R=1 THEN GOTO PROG 1
40 INPUT "PRINTER ON?(Y/N)", F$
50 IF R=2 THEN GOTO PROG 2
60 IF R=3 THEN GOTO PROG 3 ELSE IF R=
    4 THEN GOTO PROG 5

```

P1

```

10 INPUT "CLEAR (Y/N)?", S$
20 IF S$="Y" THEN CLEAR :GOTO 30 ELSE
    IF S$<>"N" THEN 10 ELSE 110
30 INPUT "X"; X, "Y"; Y
40 DIM D!(X,Y), X!(X), Y!(Y)
50 FOR J=1 TO Y:Y!(J)=0:FOR I=1 TO X
60 PRINT "INPUT DATA X="; I; " Y="; J,
70 INPUT D!(I,J)
80 Y!(J)=Y!(J)+D!(I,J)
90 NEXT I:NEXT J
100 GOSUB 200:GOTO PROG 0
110 PRINT "CORRECTION(X,Y)";
120 INPUT I,J
130 INPUT "DATA";D!(I,J)
140 FOR J=1 TO Y:Y!(J)=0:FOR I=1 TO X
150 Y!(J)=Y!(J)+D!(I,J)
160 NEXT I:NEXT J
170 GOSUB 200:GOTO PROG 0
200 S=0:FOR I=1 TO X:X!(I)=0:FOR J=1 T
    O Y
210 X!(I)=X!(I)+D!(I,J):S=S+D!(I,J)
220 NEXT J:NEXT I
230 PRINT "GRAND T. ";S:FOR K=0 TO 100:
    NEXT K

```

```
240 RETURN
P2
 10 INPUT "X-SUM OR Y-SUM";P$
 20 IF P$="Y" THEN 160 ELSE IF P$="X"
    THEN 80 ELSE 10
 80 FOR K=1 TO X
 90 PRINT "X=";K;" SUM=";X!(K)
 95 IF F$="Y" THEN GOSUB 300
 97 IF INKEY$="" THEN 97 ELSE 100
100 NEXT K
110 PRINT "GRAND T.=";S
115 IF F$="Y" THEN GOSUB 340
120 IF INKEY$="" THEN 120 ELSE 130
130 GOTO PROG 0
160 FOR K=1 TO Y
170 PRINT "Y=";K;" SUM=";Y!(K)
173 IF INKEY$="" THEN 173 ELSE 175
175 IF F$="Y" THEN GOSUB 320
180 NEXT K
190 PRINT "GRAND T.=";S
195 IF F$="Y" THEN GOSUB 340
200 IF INKEY$="" THEN 200 ELSE 210
210 GOTO PROG 0
300 LPRINT "X=";K;" SUM=";X!(K):RETURN
320 LPRINT "Y=";K;" SUM=";Y!(K):RETURN
340 LPRINT "GRAND T. ";S
350 GOTO PROG 0
P3
 10 INPUT "SORT X OR Y?",P$
 20 IF P$="Y" THEN GOSUB 100 ELSE IF P
    $="X" THEN GOSUB 200 ELSE 10
 30 GOTO PROG 4
100 ERASE A!
110 DIM A!(Y,2)
120 FOR J=1 TO Y
```

```

130 A!(J,1)=Y!(J):A!(J,2)=J
140 NEXT J
150 N=Y:RETURN
200 ERASE A!
210 DIM A!(X,2)
220 FOR I=1 TO X
230 A!(I,1)=X!(I):A!(I,2)=I:NEXT I
240 N=X:RETURN

```

P4

```

10 CLS
20 PRINT "SORTING NOW"
30 REM SORT
40 FOR K=N-1 TO 1 STEP -1
50 FOR L=1 TO K
60 IF A!(L,1)>A!(L+1,1) THEN 100
65 FOR M=1 TO 2
70 T=A!(L,M)
80 A!(L,M)=A!(L+1,M)
90 A!(L+1,M)=T
95 NEXT M
100 NEXT L
110 NEXT K
120 REM PRINT
130 FOR K=1 TO 10:BEEP :NEXT K: CLS
140 FOR K=1 TO N:GOSUB 220
150 PRINT USING"##";K;" ";P$;"=";USING
"##";A!(K,2);USING"#####";A!(K,
1);
160 PRINT USING"###";A;"%"
170 IF F$="Y" THEN GOSUB 300:NEXT K:GO
SUB 320:GOTO PROG 0
180 FOR L=1 TO 300:NEXT L:NEXT K
190 PRINT "GRAND T. ";S
200 FOR K=0 TO 300:NEXT K
210 GOTO PROG 0
220 REM RATIO
230 A=ROUND(A!(K,1)/S,-3)*100

```

```
240 RETURN
300 LPRINT USING "##"; K; " "; P$; "="; USING
    G"##"; A!(K, 2); USING "#####"; A!(K
    , 1);
310 LPRINT USING "###"; A; "%": RETURN
320 PRINT "GRAND T."; S
330 LPRINT "GRNAD T."; S
340 RETURN
P5
5 CLS
10 FOR J=1 TO Y:FOR I=1 TO X
20 PRINT "X="; I; " Y="; J; " DATA="; D!(I
    , J)
30 IF F$="Y" THEN 50
40 IF INKEY$="" THEN 40 ELSE 60
50 LPRINT "X="; I; " Y="; J; " DATA="; D!
    (I, J)
60 NEXT I:NEXT J
70 IF F$="Y" THEN LPRINT :LPRINT :LPR
    INT
80 GOTO PROG 0
```

Número de bitios utilizado: 1,737

| Contenido de las variables | | | | | |
|----------------------------|---|------|---|--------|--------------------------------|
| A | Porcentaje | I, J | Valores de los subíndices en la matriz. | P\$ | X o Y. |
| AI () | Para almacenar datos durante su ordenación. | | | R | Selección del menú. |
| | | K~M | Contadores. | S | Gran total. |
| DI () | Serie de datos. | N | Número de elementos de los datos durante la ordenación. | S\$ | Y o N (sí o no) |
| F\$ | Determina si el impresor se usará o no. | | | T | Variable para cambio de datos. |
| | | | | XI () | Matriz de sumas de X |
| | | | | YI () | Matriz de sumas de Y |

* Si desea manejar datos de más de 5 dígitos, cambie las variables AI (), DI (), XI () y YI () de la siguiente manera: A(), D(), X() y Y().

Datos de prueba

X= 1 Y= 1 DATA= 321
 X= 2 Y= 1 DATA= 369
 X= 3 Y= 1 DATA= 357
 X= 1 Y= 2 DATA= 159
 X= 2 Y= 2 DATA= 147
 X= 3 Y= 2 DATA= 123
 X= 1 Y= 3 DATA= 842
 X= 2 Y= 3 DATA= 862
 X= 3 Y= 3 DATA= 579

Operación

| | Paso | Operación de las teclas | En pantalla |
|---|------|---|--|
| Exhibición del menú → | 1 |   | 1 DATA INPUT 2 TOTAL 3 SORT 4 DATA OUTPUT ? _ |
| (1) Introducción de datos ¿Desea eliminar los datos almacenados? → | 2 |   | CLEAR(Y/N)? _ |
| ¿Cuántos ítems horizontales? → | 3 |   | X? _ |
| ¿Cuántos ítems verticales? → | 4 | 3  | Y? _ |
| Introduzca los datos (X=1, Y=1) → | 5 | 3  | INPUT DATA X= 1 Y= 1 ? _ |

| Paso | Operación de las teclas | En pantalla |
|---|--|---|
| Introduzca los datos (X=2, Y=1) → | 3 2 1  | INPUT DATA X = 2 Y = 1 ? _ |
| Exhibición del gran total → | ⋮ Introduzca todos los datos repitiendo el paso anterior. | GRAND T. 3759 |
| Exhibición del menú → | | 1 DATA INPUT 2 TOTAL 3 SORT 4 DATA OUTPUT ? _ |
| (2) Suma de los datos en X ¿Se va a usar el impresor? → | 2  N  | PRINTER ON? (Y/N) _ X-SUM OR Y-SUM? _ |
| Introduzca X y oprima  para los subtotales de X, Y y  para los subtotales de Y. Obtención de cada subtotal (SUM) y del gran total. → | X     | X=1 SUM=1322 X=2 SUM=1378 X=3 SUM=1059 GRAND T. 3759 |
| Exhibición del menú → |  | 1 DATA INPUT 2 TOTAL 3 SORT 4 DATA OUTPUT ? _ |
| (2) Suma de los datos en Y ¿Se va a usar el impresor? → | 2  N  | PRINTER ON? (Y/N) _ X-SUM OR Y-SUM? _ |
| Introduzca Y y oprima  para los subtotales de Y. → | Y     | Y=1 SUM=1047 Y=2 SUM=429 Y=3 SUM=2283 GRAND T. 3759 |
| Exhibición del menú → |  | 1 DATA INPUT 2 TOTAL 3 SORT 4 DATA OUTPUT ? _ |

CAPITULO 5 LISTA DE PROGRAMAS

| | Paso | Operación de las teclas | En pantalla |
|---|------|--|--|
| (3) Ordenación | | | |
| Si el impresor está conectado, introduzca Y y oprima  . | 14 |   | PRINTER ON ? (Y/N) _ |
| Ordenación de cada subtotal (X o Y) | 15 |   | SORT X OR Y ? _ |
| La ordenación se está ejecutando. | 16 | X  | SORTING NOW |
| Se obtienen los datos arreglados en orden descendente, junto con su clasificación, nombre del ítem, subtotal y porcentaje. | 17 | | 1 X=2 1378 37% 2 X=1 1322 35% 3 X=3 1059 28% GRAND T. 3759 |
| Exhibición del menú | 18 | | 1 DATA INPUT 2 TOTAL 3 SORT 4 DATA OUTPUT ? _ |
| Si el impresor está conectado, introduzca Y y oprima  . | 19 | 3  | PRINTER ON ? (Y/N) _ |
| Cuando se vayan a ordenar los datos de Y, introduzca Y y oprima  . | 20 |   | SORT X OR Y |
| | 21 | Y  | SORTING NOW |
| Los datos arreglados en orden descendente aparecen en la pantalla. | 22 | | 1 Y=3 2283 61% 2 Y=1 1047 28% 3 Y=2 429 11% GRAND T. 3759 |
| | 23 | | 1 DATA INPUT 2 TOTAL 3 SORT 4 DATA OUTPUT ? _ |
| (4) Obtención de los datos | | | |
| Si el impresor está conectado, introduzca Y y oprima  . | 24 |   | PRINTER ON ? (Y/N) _ |
| | 25 |      | X=1 Y=1 DATA=321 X=1 Y=1 DATA=369 X=1 Y=1 DATA=357 X=1 Y=2 DATA=159 ⋮ |

* La suma de los porcentajes no siempre es 100%; ello depende de los valores de los datos.

PROGRAMA PARA TRAZAR GRAFICAS

Este programa traza diversos tipos de gráficas con la impresora por coordenadas (FA-10 o FA-11). Se pueden introducir hasta 12 ítems de datos. El límite de los datos es el siguiente:

$$| \text{Valor de los datos} | \leq 1E90$$

El programa puede trazar gráficas de bandas, barras y líneas, aprovechando los 4 colores que tiene la impresora por coordenadas.

* Este programa se encuentra almacenado en el cassette que viene junto con la impresora por coordenadas opcional FA-11.

Explicación

Cuando este programa se ejecuta, el menú aparece en la pantalla. Primero, introduzca todos los datos oprimiendo [1].

El límite de los datos es el indicado anteriormente. Los datos pueden ser números negativos. Se pueden introducir hasta 12 ítems de datos. Si se introducen más de 12 datos, el menú volverá a aparecer en la pantalla automáticamente. Para suspender la introducción de datos, oprima la tecla [↵] sin introducir ningún dato numérico.

El menú 2 se usa para corregir los datos introducidos. Al oprimir [2] los primeros datos introducidos aparecerán en la pantalla y al oprimir [↵] aparecerán los siguientes datos. Si se oprimen las teclas [SHIFT] y [↵] aparecerán los datos previos. Cuando los datos que desee corregir aparezcan en la pantalla, introduzca los datos correctos.

El menú 3 es la rutina para hacer gráficas. Al presionar [3] aparecen en la pantalla tres tipos de nombres de gráficas. Seleccione el tipo de gráfica deseada oprimiendo [1], [2] ó [3]. (Si oprime la tecla [4], el menú su exhibirá nuevamente).

El tipo 1 es una gráfica de bandas. La longitud total de la banda representa el 100% y el porcentaje de cada elemento de datos está representado por un segmento dentro de esta banda. Cada segmento de datos se muestra en diferentes colores y rayas para que se pueda reconocer fácilmente. Si se introduce un dato negativo en este tipo de gráfica, automáticamente volverá a aparecer el menú en la pantalla.

El tipo 2 es una gráfica de barras. La escala se determina automáticamente de acuerdo con las dimensiones de los datos introducidos. En esta gráfica, los valores positivos se proporcionan en color verde y los negativos en rojo.

El tipo 3 es una gráfica lineal. Cuando se oprime la tecla [3], "Over previous graph?" aparece en la pantalla. Si se acaba de trazar una gráfica de barras, es posible trazar sobre ella una gráfica lineal oprimiendo en ese momento la tecla [Y]. Si no se ha trazado ninguna gráfica de barras, o no desea trazar sobre ella una gráfica lineal, oprima [N]. En este caso, se determina automáticamente la escala adecuada y se traza la gráfica lineal.

Independientemente del tipo de gráfica seleccionado, el menú volverá a aparecer en la pantalla después de haberse trazado la gráfica. El menú también aparecerá cuando no se introduzca ningún dato. Recuerde que cuando el programa se vuelve a activar, se eliminan los datos existentes. El menú 4 finaliza la ejecución del programa.

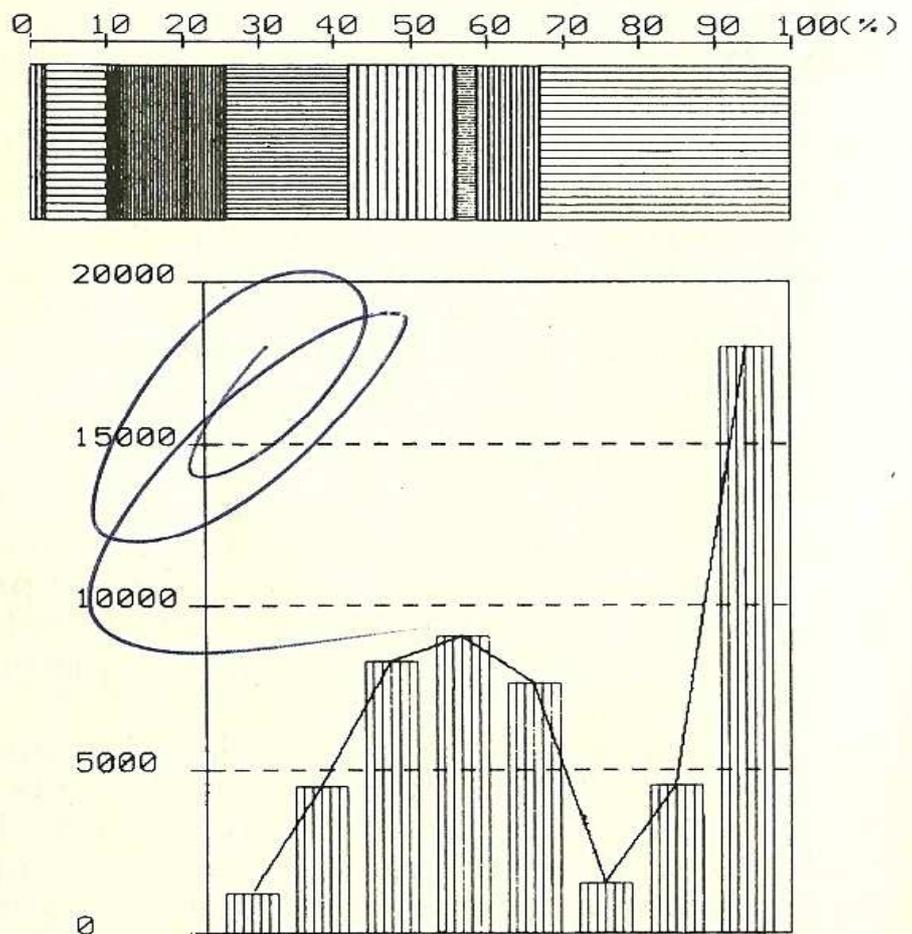
El menú 5 transfiere los datos a la impresora por coordenadas. También se puede usar para obtener el total de los datos.

Ejemplo de impresión

Print Data

D(1)= 1200
 D(2)= 4500
 D(3)= 8383
 D(4)= 9102
 D(5)= 7701
 D(6)= 1532
 D(7)= 4562
 D(8)= 18020

Total= 55000



Programa

```
P0
10 CLEAR
20 CLS :PRINT " ---- DATA ----"
30 PRINT TAB(2);"1:Input 2:Correct",TAB(2);"3:Graph 4:END",TAB(2);"5:Print Data";
40 K=VAL(INKEY$); IF K<1 THEN 40 ELSE IF K>5 THEN 40
```

PROGRAM PARA TRAZAR GRAFICAS

```

50 BEEP :GOTO K*100
100 CLS :ERASE D: DIM D(13):Z=1
110 LOCATE 2,3:PRINT "<RTEURN> : END";
120 LOCATE 6,0:PRINT " ";
130 LOCATE 0,0:PRINT "D(";Z;")=";:INPU
    T "",AB$
140 IF AB$<>"" THEN D(Z)=VAL(AB$):IF Z
    <12 THEN Z=Z+1:GOTO 120 ELSE 20 EL
    SE Z=Z-1:GOTO 20
200 IF Z<1 THEN 20
210 CLS :I=1:PRINT TAB(46);":Shift RE
    TURN",TAB(6);":RETURN ";
220 LOCATE 4,2:PRINT CHR$(228);CHR$(22
    9):LOCATE 4,3:PRINT CHR$(230);CHR$
    (231);
230 DRAWC(32,23)-(47,23)-(47,24)-(32,2
    4):LOCATE 0,0:PRINT "D(";
240 LOCATE 2,0:PRINT I;")=";D(I);"
    ":LOCATE 6,1:PRINT "
    "
250 LOCATE 6,1:INPUT AB$:K$=INKEY$:IF
    AB$<>"" THEN D(I)=VAL(AB$)
260 IF K$=CHR$(24) THEN I=I-1:IF I<1 T
    HEN 20 ELSE 240
270 IF K$=CHR$(13) THEN I=I+1:IF I>Z T
    HEN 20 ELSE 240
280 IF K$=CHR$(23) THEN I=I+1:IF I>Z T
    HEN 20 ELSE 240
290 IF K$="" THEN I=I+1:IF I>Z THEN 20
    ELSE 240
300 IF Z<1 THEN 20 ELSE CLS
310 PRINT TAB(5);"1:Band",TAB(5);"2:Ba
    r"

```

```
320 PRINT TAB(5); "3:Line", TAB(5); "4:ME
    NU";
330 K=VAL(INKEY$); IF K<1 THEN 330 ELSE
    IF K>4 THEN 330
340 BEEP :GOTO K*1000
400 LPRINT CHR$(28); CHR$(46):END
500 IF Z<1 THEN 20 ELSE GOSUB 6000:LPR
    INT "Q1":U=0
510 LPRINT "M93,0", "PPrint Data"
520 FOR I=1 TO Z
530 LPRINT "M"; 90-4*I; ",0", "PD("; MID$(
    STR$(I), 2); ")="; D(I)
540 U=U+D(I):NEXT I
550 LPRINT "M"; 90-I*4-5; ",0", "PTotol="
    :U:GOSUB 6000:GOTO 20
1000 GOSUB 6000:A=0
1010 FOR I=1 TO Z:IF D(I)<0 THEN ERASE
    I:GOTO 20 ELSE A=A+D(I):NEXT I
1020 IF A<=0 THEN 20
1030 LPRINT "05,0", "X1,8,10", "M85,1", "P
    ("; CHR$(37); ") "
1040 FOR I=100 TO 0 STEP -10:LPRINT "M"
    :-4+8*I/10; ",1", "P"; I:NEXT I
1050 B=0:C=0
1060 FOR I=1 TO Z
1070 C=ROUND(D(I)/A*80+C, -2)
1080 LPRINT "J"; I MOD 4, "A"; B; ", -3, "; C;
    ", -23"
1090 LPRINT "G"; I MOD 2+1; ", "; C-B; ", -20
    , "; (I MOD 3)/4+.5
1100 B=C:NEXT I
1110 LPRINT "H30":GOTO 20
2000 GOSUB 6000:GOSUB 7000:GOSUB 8000
```

```
2010 FOR I=1 TO Z
2020 IF D(I)>=0 THEN J=2 ELSE J=3
2030 LPRINT "J";J
2040 LPRINT "A";0;",";6-8*I;",";ROUND(D
(I)/A*90/N+0,-2);",";-8*I
2050 LPRINT "M";0;",";6-8*I,"G1",";ROUND
(D(I)/A*90/N,-2);",";-6"
2060 NEXT I
2070 LPRINT "M0,";-8*(Z+1):GOTO 20
3000 IF Z<2 THEN 20 ELSE CLS :PRINT "O
ver previous graph?",TAB(8);"Y/N"
3010 K$=INKEY$:IF K$="Y" THEN BEEP :GOT
O 3080 ELSE IF K$<>"N" THEN 3010 E
LSE BEEP
3020 GOSUB 6000
3030 GOSUB 7000:GOSUB 8000
3040 LPRINT "L1","J0"
3050 FOR I=1 TO Z
3060 F=3-8*I:IF I MOD 2=1 THEN LPRINT "
D0,";F;",";90,";F ELSE LPRINT "D90,"
;F;",";0,";F
3070 NEXT I:LPRINT "L0","M0,0"
3080 S=S+1:IF S>3 THEN S=0
3090 T=T+.25:IF T>3.9 THEN T=0
3100 IF T<2 THEN LPRINT "B1.6" ELSE LPR
INT "B6.4"
3110 LPRINT "J";S,"L";T
3120 G=ROUND(D(1)/A*90/N+0,-2):H=-5
3130 FOR I=2 TO Z
3140 U=ROUND(D(I)/A*90/N+0,-2):U=3-8*I
3150 LPRINT "D";G;",";H;",";U;",";U
3160 G=U:H=U
3170 NEXT I
```

```
3180 LPRINT "B3.2", "M0, "; -8*(Z+1):GOTO
    20
4000 GOTO 20
6000 LPRINT CHR$(28);CHR$(37), "00,0", "J
    0", "L0", "S1", "Q0", "Y0", "B3.2", "H20
    "
6010 S=0:T=0:RETURN
7000 Y=-9E99:B=9E99
7010 FOR I=1 TO Z
7020 IF D(I)>Y THEN Y=D(I)
7030 IF D(I)<B THEN B=D(I)
7040 NEXT I
7050 IF Y>=0 THEN D(0)=Y ELSE D(0)=0
7060 IF B>0 THEN D(Z+1)=0 ELSE D(Z+1)=B
7070 RETURN
8000 IF SGND(0)*SGND(Z+1)<=0 THEN M=ABS
    (D(0)-D(Z+1)) ELSE M=D(0):IF M<0 T
    HEN M=ABSD(Z+1)
8010 IF M<0 THEN 20 ELSE R=INTLGTM:A=1
    0^R
8020 IF A*INT(M/A)*.75<M THEN A=A*.5
8030 D=LEN(STR$(A))*2.4+5
8040 IF SGND(0)*SGND(Z+1)<0 THEN N=INT(
    M/A)+2 ELSE N=INT(M/A)+1
8050 C=ABSINT(D(0)/A)+SGND(0)
8060 FOR I=N TO 0 STEP -1:IF C=0 THEN 0
    =I*90/N
8070 C=C-1:NEXT I
8080 LPRINT "D0,0,90,0", "Q1":W=18/N:U=0
8090 U=U+5:IF U*W<18 THEN 8090
8100 IF D(0)<=0 THEN 8150 ELSE X=0
8110 K=ROUND(X,-2):LPRINT "D";K:";2,";K
    ;",-2"
```


Operación

| | Paso | Operación de las teclas | En pantalla |
|---|------|---|--|
| Exhibición del menú → | |   | ----- DATA ----- 1 : Input 2 : Correct 3 : Graph 4 : END 5 : Print Data |
| (1) Introducción de datos | 1 |  | D(1) = _ <RETURN> : END |
| Después de introducir el 12º dato, el menú volverá a aparecer automáticamente en la pantalla. → | 2 | 2112  ⋮ | D(2) = _ ⋮ |
| (2) Corrección de datos Después de hacerse la última corrección, el menú vuelve a exhibirse. → | 3 |  1200  | D(1) = 1200 ? _ ▲ Shift RETURN (Datos previos) ▼ RETURN (Datos siguientes) |
| (3) Elaboración de gráficas | 4 |  | 1 : Band 2 : Bar 3 : Line 4 : MENU |
| La gráfica de bandas se imprime. → | 5 |  | (Después de imprimirse la gráfica, el menú se vuelve a exhibir). |
| La gráfica de barras se imprime. → | |  | (Después de imprimirse la gráfica, el menú se vuelve a exhibir). |
| Determine si se va a sobreponer una gráfica lineal en la gráfica de barras. → | |  | Over previous graph? Y/N |
| La gráfica lineal se imprime → | |  o  | (Después de imprimirse la gráfica, el menú se vuelve a exhibir). |
| (5) Obtención de datos. Los datos se imprimen. → | 6 |  | (Después de imprimirse los datos el menú se vuelve a exhibir). |
| (4) Finaliza la ejecución. | 7 |  | |

CAPITULO 6

MATERIAL DE REFERENCIA

6-1 TABLA DE MANDOS DE LA UNIDAD PB-770

6-1-1 Símbolos de operadores

1 Operadores aritméticos

| Nombre | Formato general | Formato de PB-770 | Significado | Prioridad |
|----------------|-----------------|--------------------|---|-----------|
| Potenciación | x^y | $X \wedge Y$ | Eleva x a la potencia y . | 1 |
| Multiplicación | $x \times y$ | $X * Y$ | Multiplica x por y . | 2 |
| División | $x \div y$ | X / Y | Divide x entre y . | 2 |
| Remanencia | $x \div y = z$ | $X \text{ MOD } Y$ | Lo que queda sin dividir cuando se divide x entre y . | 3 |
| Suma | $x + y$ | $X + Y$ | Añade y a x . | 4 |
| Resta | $x - y$ | $X - Y$ | Resta y de x . | 4 |
| Asignación | $x = y + 5$ | $X = Y + 5$ | Asigna $y + 5$ a x . | 5 |

2 Operadores relacionales (expresiones condicionales)

| Formato general | Formato de PB-770 | Significado |
|-----------------|---------------------|--------------------------------|
| $x = y$ | $X = Y$ | x es igual a y . |
| $x \neq y$ | $X <> Y, X > < Y$ | x no es igual a y . |
| $x < y$ | $X < Y,$ | x es menor que y . |
| $x > y$ | $X > Y$ | x es mayor que y . |
| $x \leq y$ | $X \leq Y, X = < Y$ | x es menor o igual que y . |
| $x \geq y$ | $X \geq Y, X = > Y$ | x es mayor o igual que y . |

- Los operadores relacionales son válidos solamente en las órdenes IF.
- Se puede hacer una comparación entre constantes numéricas, variables numéricas y expresiones numéricas; también se puede hacer una comparación entre constantes de caracteres y variables de caracteres.

3 Operadores de expresión de caracteres

+ Dos o más series de caracteres se pueden encadenar con un + (signo de suma).

6-1-2 Caracteres especiales

| Formato general | Formato PB-770 | Significado |
|-----------------|----------------|--|
| $x \times 10^y$ | XEY | Introducción de exponente (Multiplica el número x por la potencia y de 10). |

- Si el valor absoluto del resultado de la operación es igual o mayor que 10^{10} o menor que 10^{-10} (0,001), es automáticamente indicado por una notación exponencial.

6-1-3 Funciones incorporadas

| Nombre | Formato general | Formato PB-700 | Significado/observaciones | Página |
|--------------------------|--|-------------------------------------|---|-------------------|
| Trigonométricas | $\sin x$ | SIN X | Da el seno de x | 214 217 218 |
| | $\cos x$ | COS X | Da el coseno de x | |
| | $\tan x$ | TAN X | Da la tangente de x | |
| Trigonométricas inversas | $\sin^{-1} x$ $\cos^{-1} x$ $\tan^{-1} x$ | ASN X ACS X ATN X | Da el arcoseno de x Da el arcocoseno de x Da el arcotangente de x | 219 |
| Hiperbólicas | $\sinh x$ $\cosh x$ $\tanh x$ | HYP SINX HYP COSX HYP TANX | Determina el $\sinh x$ Determina el $\cosh x$ Determina el $\tanh x$ | 221 |
| Hiperbólicas inversas | $\sinh^{-1} x$ $\cosh^{-1} x$ $\tanh^{-1} x$ | HYP ASN X HYP ACS X HYP ATN X | Determina el $\sinh^{-1} x$ Determina el $\cosh^{-1} x$ Determina el $\tanh^{-1} x$ | 221 |
| Logarítmicas | $\log x$ $\ln x$ | LGT X LOG X | $\log_{10} x$ (logaritmo común) $\log_e x$ (logaritmo natural) | 223 |
| Exponenciales | e^x | EXP X | Potencia x de la base del logaritmo natural (e) | 226 |
| Potenciación | x^y | X ^ Y | Potencia y de x | 26 |
| Raíz cuadrada | \sqrt{x} | SQR X | Da la raíz cuadrada ($x \geq 0$) | 222 |
| Valor absoluto | $ x $ | ABS X | Da el valor absoluto de x | 228 |
| Enteros | | INT X | Cuando $x > 0$, la fracción de x se descarta. Cuando $x < 0$, la fracción de $ x $ se redondea, y un - (signo menos) aparece x . INT 1.2 \rightarrow 1 INT -1.2 \rightarrow -2 | 230 |
| Fracción | | FRAC X | Elimina la porción de enteros para obtener sólo la fracción. | 232 |
| Constante circular | π | PI | Da el radio aproximado de la circunferencia de un círculo en relación a su diámetro en 11 dígitos: 3,1415926536. | 238 |
| Número aleatorio | | RND | Genera un número pseudoaleatorio de 10 dígitos ($0 < \text{RND} < 1$). | 239 |
| Signo | | SGN X | Verifica el signo de un argumento: $x < 0 \rightarrow -1$ $x = 0 \rightarrow 0$ $x > 0 \rightarrow 1$ | 234 |

| Nombre | Formato general | Formato PB-770 | Significado/observaciones | Página |
|-------------------------------------|-----------------------|--|--|--------|
| Redondeo | | ROUND (X,Y) | Redondea por incremento o por desprecio el valor de x a 10^Y posiciones. | 236 |
| Grados, minutos y segundos | Sexagesimal → Decimal | DEG(d[,m[,s]]) (d, m, s: expresiones numéricas) | Determina el equivalent decimal de una valor hexadecimal | 241 |
| Lectura del contenido de la memoria | | PEEK X | Da el contenido de la dirección de memoria x . | 242 |

6-1-4 Funciones de caracteres

| Uso | Función | Ejemplo | Significado | Página |
|--|----------|------------------------|---|--------|
| Da el código de caracteres del primer carácter en una secuencia. | ASC | PRINT ASC ("E") | Exhibe el código de caracteres del carácter E. | 243 |
| Da un caracter designado por un código de caracteres. | CHR \$ | PRINT CHR \$ (69) | Exhibe el carácter E, correspondiente al código de caracteres 69. | 245 |
| Convierte el numeral en una serie de caracteres a un valor numérico. | VAL | A = VAL (X\$) | Convierte una serie de caracteres de numerales almacenados en una variable de caracteres X\$ a un valor numérico. | 247 |
| Convierte un valor numérico a una serie de caracteres. | STR \$ | C\$ = STR \$(X) | Convierte un valor numérico almacenado en una variable numérica X a una serie de caracteres. | 250 |
| Toma el número de caracteres especificado a partir de la izquierda de una serie de caracteres. | LEFT \$ | C\$ = LEFT \$(X\$, 3) | Toma los tres caracteres de la izquierda de una serie de caracteres almacenada en X\$ y los asigna a C\$. | 252 |
| Toma el número especificado de caracteres a partir de la derecha de una serie de caracteres. | RIGHT \$ | C\$ = RIGHT \$(X\$, 3) | Toma los tres caracteres de la derecha de una serie de caracteres almacenada en X\$ y los asigna a C\$. | 253 |

| Uso | Función | Ejemplo | Significado | Página |
|---|----------|-----------------------------|--|--------|
| Toma el número especificado de caracteres a partir de la posición especificada. | MID \$ | C\$ = MID \$ (X\$, 3, 5) | Toma los cinco caracteres a partir del tercer carácter de una serie de caracteres almacenada en X\$ y los asigna a C\$. | 254 |
| Cuenta el número de caracteres en una serie de caracteres. | LEN | A = LEN (X\$) | Asigna a A el número de caracteres en una serie de caracteres almacenados en X\$. | 256 |
| Introduce un caracter del teclado. | INKEY \$ | A\$ = INKEY \$ | Cuando INKEY\$ se ejecuta, la tecla que se oprima mediante el teclado le será asignada a A\$. Únicamente se puede asignar un caracter a A\$. | 257 |
| Convierte un valor decimal en sexagesimal | DMS\$ | C\$=DMS\$ (x) | Convierte el valor numérico asignado a x en una secuencia de caracteres que representa el valor sexagesimal de X. | 259 |
| Convierte un valor decimal en hexadecimal. | HEX\$ | C\$=HEX\$(x) | Convierte el valor numérico asignado a X en una secuencia de caracteres que representa el valor hexadecimal de X. | 260 |

6-1-5 Funciones de visualización

| Uso | Función | Ejemplo | Significado | Página |
|--|---------|---------------------------|--|--------|
| Verifica si se exhibe o no un punto en la pantalla. | POINT | POINT (10, 20) | Verifica si el punto de las ;las coordenadas (10, 20) está encendido o no da "1" si el punto está encendido y "0" si no lo está. | 266 |
| Mueve el indicador el número especificado de posiciones. | TAB | PRINT TAB(10) | Desplaza el indicador hasta la posición 10 en la pantalla. | 261 |
| Especifica el formato de la información de salida. | USING | PRINT USING "###,##";A | Exhibe un valor numérico almacenado en la variable numérica A de acuerdo al formato "###.##". | 263 |

6-1-6 Funciones estadísticas

| Uso | Formato común | Significado | Página |
|-------|-----------------|---|--------|
| CNT | n | Número de datos estadísticos procesados. | 270 |
| SUMX | Σx | Sumatoria de los datos x . | 271 |
| SUMY | Σy | Sumatoria de los datos y . | |
| SUMX2 | Σx^2 | Sumatoria de los cuadrados de los datos x . | |
| SUMY2 | Σy^2 | Sumatoria de los cuadrados de los datos y . | |
| SUMXY | Σxy | Sumatoria de los productos de los datos x e y . | |
| MEANX | \bar{x} | Media de los datos x . | 272 |
| MEANY | \bar{y} | Media de los datos y . | |
| SDX | $x\sigma_{n-1}$ | Desviación estándar por muestreo de los datos x . | 273 |
| SDY | $y\sigma_{n-1}$ | Desviación estándar por muestreo de los datos y . | |
| SDXN | $x\sigma_n$ | Desviación estándar total de los datos x . | |
| SDYN | $y\sigma_n$ | Desviación estándar total de los datos y . | |
| LRA | a | Término constante de regresión lineal. | 274 |
| LRB | b | Coefficiente de regresión lineal. | |
| COR | r | Coefficiente de correlación. | 270 |
| EOX | \hat{x} | Valor estimado de x para y . | 274 |
| EOY | \hat{y} | Valor estimado de y para x . | |

6-1-7 Mandos manuales

| Uso | Mando | Ejemplo | Significado | Página |
|--|--------|---------------------|--|--------|
| Genera números de línea automáticamente. | AUTO | AUTO | Números de línea a partir de 10 con incrementos de 10. | 128 |
| | | AUTO 100 | Números de línea a partir de 100 con incrementos de 10. | |
| | | AUTO 50, 20 | Números de líneas a partir de 50 con incrementos de 20. | |
| Reanuda la ejecución de un programa. | CONT | CONT | Reanuda la ejecución de un programa que ha sido detenido mediante la sentencia STOP o la tecla $\frac{STOP}{(AB)}$. | 129 |
| Elimina un programa (o parte de él) | DELETE | DELETE 50 | Elimina la línea 50. | 130 |
| | | DELETE 30 - | Elimina desde la línea 30 hasta la última. | |
| | | DELETE - 100 | Elimina desde la primera hasta la 100. | |
| | | DELETE 150 - 200 | Elimina de la línea 150 a la 200. | |
| Modifica un programa. | EDIT | EDIT | Exhibe la primera línea y especifica la modalidad EDIT. | 132 |
| | | EDIT 30 | Exhibe la línea 30 y especifica la modalidad EDIT. | |
| Visualiza la lista de un | LIST | LIST | Exhibe el programa en el área del programa que se haya especificado. | 135 |
| | | LIST 50 | Exhibe la línea 50. | |
| | | LIST 30 - | Exhibe desde la línea 30 hasta la última; | |
| | | LIST - 50 | Exhibe hasta la línea 50. | |
| | | LIST 30 - 50 | Exhibe las líneas 30 a 50. | |
| | | LIST ALL | Exhibe un todos los programas en la áreas. | |
| | | LIST V | Exhibe los nombres de las variables registradas. | |

| Uso | Mando | Ejemplo | Significado | Página |
|--|-------|--------------------|--|------------|
| Imprime la lista de un programa. | LLIST | LLIST | Imprime el programa en el área del programa que se haya especificado. | 135 |
| | | LLIST 50 | Imprime la línea 50. | |
| | | LLIST 30 – | Imprime desde la línea 30 hasta la última. | |
| | | LLIST – 50 | Imprime desde la primera hasta la 50. | |
| | | LLIST 30 – 50 | Imprime las líneas 30 hasta 50. | |
| | | LLIST ALL | Imprime todos los programas. | |
| | | LLIST V | Imprime los nombres de las variables registradas. | |
| Recupera programas desde cintas de casette. | LOAD | LOAD | Recupera un programa en el formato de código interno y lo almacena en el área de programa en curso. | 138 |
| | | LOAD ALL | Recupera todos los programas en el formato de código interno y los almacena en las áreas de programa correspondientes. | |
| | | LOAD, A | Recupera un programa en formato ASCII y lo almacena en el área de programa en curso. | |
| | | LOAD, M | Encadena el programa almacenado en el área de programa en curso con el programa recuperado grabado en formato ASCII. | |
| | | LOAD, D, 4096 | Recupera datos en el formato de código interno y los almacena en la parte de la memoria a partir de la dirección 4096 (decimal). | |
| | | LOAD "ABC" | Lleva a cabo funciones similares a las mencionadas arriba para los programas cuyos nombres de archivo se especifican. | |
| | | LOAD ALL "CASIO" | | |
| | | LOAD "TEST", A | | |
| | | LOAD "TEST", M | | |
| | | LOAD "PB", D, 4096 | | |
| Borra el programa. | NEW | NEW | Borra un programa en el área del programa que se haya especificado. | 142 |
| | | NEW ALL | Borra toda la RAM. | |

6-1 TABLA DE MANDOS DE LA UNIDAD PB-770

| Uso | Mando | Ejemplo | Significado | Página |
|--|--------|--------------------------|--|--------|
| Protege un programa. | PASS | PASS "KEY" | Determina una clave llamada "KEY". | 143 |
| Especifica el área de un programa. | PROG | PROG 2 | Especifica el área 2 de un programa. | 145 |
| Inicia la ejecución de un programa. | RUN | RUN | Inicia la ejecución de un programa desde el principio del área de programa que se haya especificado. | 146 |
| | | RUN 100 | Inicia la ejecución de un programa desde la línea 100. | |
| Graba programas en cintas de cassette | SAVE | SAVE | Graba en cinta de cassette con formato de código interno el programa almacenado en el área de programa en curso. | 147 |
| | | SAVE ALL | Graba en cinta de cassette con formato de código interno los programas almacenados en las áreas de programa. | |
| | | SAVE, A | Graba en cinta de cassette con formato ASCII el programa almacenado en el área de programa en curso. | |
| | | SAVE, D, 3000, 3999 | Graba en cinta de cassette los datos con formato de código interno que se encuentran en las posiciones de memoria de 3000 a 3999 (decimal). | |
| | | SAVE "ABC" | Realiza funciones similares a la mencionadas arriba con los programas cuyos nombres de archivo se especifican. | |
| | | SAVE ALL "CASIO" | | |
| | | SAVE "TEST", A | | |
| | | SAVE "PB", D, 3000, 3999 | | |
| Exhibe el estado de las áreas del programa. | SYSTEM | SYSTEM | Visualiza el estado de las áreas de programa, la unidad angular especificada, la capacidad de memoria, el número remanente de bytes y la dirección donde comienza la memoria para datos. | 150 |
| Verifica los programas almacenados en una cinta de cassette. | VERIFY | VERIFY | Realiza una verificación o prueba de paridad del archivo del programa que aparezca primero. | 152 |
| | | VERIFY "ABC" | Realiza una verificación o prueba de paridad del programa que tenga el nombre del archivo especificado. | |

6-1-8 Mandos de programa

| Uso | Mando | Ejemplo | Significado | Página |
|--|-------|------------------|--|--------|
| Especifica la unidad angular. | ANGLE | ANGLE 0 | Especifica los grados. | 153 |
| | | ANGLE 1 | Especifica los radianes. | |
| | | ANGLE 2 | Especifica los gradientes. | |
| Genera un zumbido. | BEEP | BEEP | Igual que BEEP 0. | 154 |
| | | BEEP 0 | Produce un zumbido grave. | |
| | | BEEP 1 | Produce un zumbido agudo. | |
| Extrae y ejecuta un programa. | CHAIN | CHAIN | Extrae y ejecuta el PF B que aparezca primero | 155 |
| | | CHAIN "XYZ" | Extrae y ejecuta el programa con el nombre del archivo especificado. | |
| Borra todas las variables. Especifica la dirección a partir de la cual comienza la memoria para datos. | CLEAR | CLEAR | Borra todas las variables. | 157 |
| | | CLEAR 4000 | Borra todas las variables y reserva la memoria para datos a partir de la dirección 4000 (decimal). | |
| Borra lo exhibido en la pantalla. | CLS | CLS | Borra todo lo exhibido en la pantalla y hace que el indicador se desplace hasta la posición inicial. | 161 |
| Almacena los datos. | DATA | DATA 1, 2, 3 | Almacena los datos que sirven de referencia a la orden READ. | 205 |
| Especifica una tabla o lista. | DIM | DIM A (3) | Especifica una matriz numérica de precisión simple (hasta 8 dígitos). | 162 |
| | | DIM B (2, 3) | Especifica una matriz numérica de precisión simple. | |
| | | DIM C! (4) | Especifica una matriz numérica de precisión doble (más de 8 dígitos). | |
| | | DIM D! (3, 4) | Especifica una matriz numérica de precisión doble. | |
| | | DIM E\$ (5) | Especifica una matriz de caracteres de longitud fija. | |
| | | DIM F\$ (4, 5) | Especifica una tabla de caracteres de longitud fija. | |
| | | DIM G\$ (2)*3 | Especifica una matriz de caracteres de longitud definida (3 caracteres por elemento en este caso). | |
| | | DIM H\$ (4, 5)*6 | Especifica una matriz de caracteres de longitud definida (6 caracteres por elemento en este caso). | |

| Uso | Mando | Ejemplo | Significado | Página |
|--|---------------------------|--|---|--------|
| Traza puntos y líneas rectas. | DRAW | DRAW (0, 0) | Traza un punto en las coordenadas (0, 0). | 167 |
| | | DRAW (1,0) – (5,10) | Traza una línea recta de las coordenadas (1, 0) a las (5, 10). | |
| Borra puntos y líneas rectas. | DRAWC | DRAWC (0, 0) | Borra el punto en las coordenadas (0, 0) | 167 |
| | | DRAWC (1, 0) – (5, 10) | Borra la línea recta entre las coordenadas (1, 0) y (5, 10). | |
| Finaliza la ejecución de un programa | END | END | Finaliza la ejecución de un programa. | 170 |
| Libera un nombre de table. | ERASE | ERASE A | Libera la especificación de la variable A registrada o de matriz. | 171 |
| Lazo (repetición). | FOR TO STEP NEXT | FOR I = 5 TO 20 STEP 0.5 NEXT I | Realiza repetidamente el procedimiento entre FOR y NEXT a la vez que va incrementando en 0.5 el valor de la variable I, desde 5 hasta 20. | 172 |
| Extrae los datos de variables de la cinta de cassette. | GET | GET A | Extrae los datos de variables que aparezcan primero. | 177 |
| | | GET "MAX" B | Extrae los datos de variables que tengan el nombre de archivo "MAX". | |
| Se salta hasta una subrutina. | GOSUB | GOSUB 100 | Se salta hasta la subrutina en la línea 100. | 180 |
| | | GOSUB PROG 3 | Se salta hasta la subrutina en el área 3 del programa. | |
| Fin de la subrutina. | RETURN | RETURN | Regresa a la línea o mando que esté después de la orden GOSUB. | 180 |
| Salto incondicional. | GOTO | GOTO 500 | Se salta hasta la línea 500. | 184 |
| | | GOTO PROG 5 | Se salta hasta el área 5 del programa. | |
| Salto condicional. | IF THEN ~ ELSE ~ | IF I > 9 THEN 50 ELSE 80 | Se salta hasta la línea 50 si el valor es mayor de 9; de no ser así, se salta hasta la línea 80. | 186 |
| Introducción de datos mediante el teclado. | INPUT | INPUT S | Se exhibe un ? , y luego espera a que se introduzcan los datos en S. | 189 |
| | | INPUT "NAME"; T\$ | Se exhibe NAME, y luego espera a que se introduzcan los datos en T\$. | |
| | | INPUT "NAME"; U\$ | Se exhibe NAME? , y luego espera a que se introduzcan los datos en U\$. | |

| Uso | Mando | Ejemplo | Significado | Página |
|---|------------|---------------|--|--------|
| Asigna datos a variables. | LET | LET A = B | Asigna B a A. | 195 |
| Especifica la posición de indicador. | LOCATE | LOCATE 2, 3 | Especifica las coordenadas (2, 3) como la posición del indicador. | 196 |
| Escrube datos en la dirección de memoria especificada | POKE | POKE 3500, 10 | Escribe el valor 10 (decimal) en la dirección 3500 (decimal). | 197 |
| Exhibe los datos en la pantalla. | PRINT | PRINT C, D | Exhibe los valores de C y D en líneas separadas. | 198 |
| | | PRINT C; D | Exhibe los valores de C y D uno a continuación del otro. | |
| | | PRINT \$AB\$ | Visualiza el patrón definido por AB\$ | |
| Imprime los datos. | LPRINT | LPRINT C, D | Imprime los valores de C y D en líneas separadas. | 198 |
| | | LPRINT C; D | Imprime los valores de C y D uno a continuación del otro. | |
| Almacena los datos de variables en la cinta de cassette. | PUT | PUT A | Almacena los datos de la variable A en cinta de cassette. | 203 |
| | | PUT "DATA" A | Almacena los datos de la variable A con un nombre de archivo en cinta de cassette. | |
| Extrae los datos almacenados. | READ | READ X | Extrae los datos almacenados en la variable X mediante la orden DATA. | 205 |
| Comentarios u observaciones. | REM | REM *** | Proporciona espacio para comentarios en un programa. | 209 |
| Especifica la secuencia de ejecución de los datos en la sentencia DATA. | RESTORE | RESTORE | Extrae los primeros datos de la orden DATA cuando se ejecuta la siguiente orden READ. | 205 |
| | | RESTORE 100 | Extrae datos de la orden DATA en la línea 100 cuando se ejecuta la siguiente orden READ. | |
| Entrada de datos estadísticos | STAT | STAT 1, 2; 2 | Entrada de los datos x e y, y de la frecuencia. | 268 |
| Borrado de los registros estadísticos | STAT CLEAR | STAT CLEAR | Asignación de cero a CNT, SUMX, SUMY, SUMX2, SUMY2, y SUMXY | 269 |

| Uso | Mando | Ejemplo | Significado | Página |
|---|------------|------------|---|--------|
| Visualización del contenido de los registros estadísticos | STAT LIST | STAT LIST | Visualización del contenido de CNT, SUMX, SUMY, SUMX2, SUMY2 y SUMXY | 269 |
| Imprime el contenido de los registros para cálculos estadísticos. | STAT LLIST | STAT LLIST | Imprime en la impresora los contenidos de CNT, SUMX, SUMY, SUMX2, SUMY2 y SUMXY. | 269 |
| Detiene la ejecución de un programa. | STOP | STOP | Detiene la ejecución de un programa. | 210 |
| Rastrea la ejecución de un programa | TRON | TRON | Especifica el estado de rastreo y rastrea el estado de la ejecución de un programa. | 212 |
| Suspende el rastreo de la ejecución de un programa. | TROFF | TROFF | Suspende la modalidad de rastreo. | 212 |

6-2 TABLA DE MENSAJES DE ERROR

| Mensaje | Significado | Medida para corregirlo |
|---|---|---|
| BS error (valor de ubicación (subscript) incorrecto) | <ul style="list-style-type: none"> El valor de ubicación de una variable de una matriz es un valor negativo o el valor excede 255. Ejemplo: DIM A (256) El valor numérico especificado está fuera de la gama del argumento. Ejemplo: El argumento de la función POINT tiene los siguientes límites: $0 \leq X \leq 159$ y $0 \leq Y \leq 31$ Pero el valor numérico especificado está fuera de este límite. | <ul style="list-style-type: none"> Cambie el valor de ubicación por un valor dentro del límite especificado. Si el valor de ubicación es una variable, verifique el valor asignado. Vuelva a especificar el valor de ubicación dentro del límite del argumento especificado. |
| BV error (Saturación de la memoria provisional) | <ul style="list-style-type: none"> Saturación de la memoria provisional de entrada o salida. | <ul style="list-style-type: none"> Cada operación u orden de un programa no puede excederse de 79 caracteres de longitud. |
| DA error (error en los datos) | <ul style="list-style-type: none"> Se ejecutó una sentencia READ y/o GET a pesar de no haber datos. | <ul style="list-style-type: none"> Revise la relación entre las sentencias READ y DATA. Asegúrese que haya datos para cada READ. |
| DD error (Definición duplicada) | <ul style="list-style-type: none"> Se definieron dos veces matrices con el mismo nombre pero con un subíndice diferente. Ejemplo: Las siguientes definiciones de variables de matriz mediante un mando DIM causan un error DD DIM A (1), A (2, 3) DIM A! (1), A! (2, 3) DIM A\$ (1), A\$ (2, 3) DIM A\$ (1)*20 A\$ (2, 3) *20 DIM A\$ (1)*20 A\$ (2, 3)*20 | <ul style="list-style-type: none"> Verifique las variables en la línea en la cual ocurrió el error. También verifique los subíndices de las matrices mismo nombre. Cambie cualquiera de los nombres de matrices. Entre CLEAR o ERASE antes que la sentencia DIM o ejecútelo manualmente para borrar la matriz. |
| FC error (se ha pedido una función inadecuadamente) | <ul style="list-style-type: none"> Se intentó ejecutar cualquiera de los siguientes mandos manuales en un programa: CONT, PASS, RUN, EDIT, DELETE Se intentó ejecutar cualquiera de los siguientes mandos del programa en forma manual: END, LET, REM, STOP, LOCATE, DRAW, DRAWC, GOTO, GOSUB, RETURN, INPUT, DATA, READ, RESTORE, FOR~NEXT, IF~THEN~ELSE~ Se usó el mando CONT cuando el programa no se podía reanudar. | <ul style="list-style-type: none"> Elimine el mando incorrecto del programa. Ejecute el mando con un número de línea incluido. Oprima la tecla BRK. Después vuelva a ejecutar el programa desde el principio o, si sabe en que línea se interrumpió el programa, desde la línea anterior a la línea interrumpida mediante el mando RUN y el número de la línea. |

| Mensaje | Significado | Medida para corregirlo |
|---|--|--|
| FO error (orden NEXT sin FOR) | <ul style="list-style-type: none"> ● Sentencia FOR sin NEXT. | <ul style="list-style-type: none"> ● Revise la estructura de los circuitos. |
| GS error (RETURN sin GOSUB) | <ul style="list-style-type: none"> ● Sentencia RETURN sin un GOSUB que le corresponda. | <ul style="list-style-type: none"> ● Revise la estructura del circuito en la sentencia GOSUB y distinga claramente la rutina principal y la subrutina. |
| MA error (error matemático) | <ul style="list-style-type: none"> ● Una operación aritmética que contiene valores numéricos o funciones numéricas no es adecuada o es imposible. Ejemplo: División entre 0. | <ul style="list-style-type: none"> ● Verifique la expresión numérica en la línea en la cual ocurrió el error. También verifique el valor de la variable asociada. |
| NO error (saturación de circuitos dentro de otros circuitos-“nest”) | <ul style="list-style-type: none"> ● El número de niveles de lazos dentro de otros excede el límite especificado. Ejemplo: GOSUB~RETURN Máx. 12 niveles FOR~NEXT Máx. 6 niveles | <ul style="list-style-type: none"> ● Verifique la estructura de los circuitos y reduzca los niveles al rango permitido. |
| NR error (aparato no listo) | <ul style="list-style-type: none"> ● Aparato de entrada/salida no está conectado correctamente. Ejemplo: No está conectada una grabadora de cinta magnética. | <ul style="list-style-type: none"> ● Verifique que el aparato correspondiente esté adecuadamente conectado y activado. |
| OM error (superación de la capacidad de la memoria) | <ul style="list-style-type: none"> ● Capacidad de RAM insuficiente. ● No alcanzan los bytes remanentes del área de trabajo para la ejecución de PRINT (LPRINT). Ejemplo) 5 bytes remanentes. ● Dirección de memoria equivocada en la sentencia CLEAR. | <ul style="list-style-type: none"> ● Borre los programas que no necesite. Reduzca el tamaño de la memoria para datos usando el mando CLEAR. Amplíe la capacidad de la memoria usando un módulo RAM. Confirme el número de bytes remanentes usando el mando SYSTEM. ● Use el mando CLEAR con una dirección de memoria mayor y reduzca el tamaño de la memoria para datos. |
| OV error (error de capacidad excedida) | <ul style="list-style-type: none"> ● El resultado de una operación o el valor numérico introducido se excede de 10^{100}. | <ul style="list-style-type: none"> ● Revise la expresión numérica en la línea en la cual ocurrió el error. Inserte una sentencia PRINT en el programa para verificar el valor de la variable. |

| Mensaje | Significado | Medida para corregirlo |
|--|---|--|
| PR error (error de protección) | <ul style="list-style-type: none"> ● Se intentó ejecutar cualquiera de los siguientes mandos, los cuales no se pueden usar cuando un programa tiene una clave: DELETE, LIST, LLIST, NEW, EDIT ● Se intentó agregar una nueva línea o eliminar una línea de un programa que tiene una clave. ● Se introdujo una clave diferente. ● Se intentó extraer un programa cuya clave es diferente de la clave de la PB-770. | <ul style="list-style-type: none"> ● Vuelva a introducir la clave. Elimine la condición de bloqueo y ejecute el programa. ● Introduzca la clave correcta. ● Elimine la clave de la PB-770 antes de cargar el programa. en este caso, la clave nueva cargada con el programa será la clave de la PB-770. |
| RW error (error al extraer o introducir) | <ul style="list-style-type: none"> ● Error de paridad durante la ejecución de un mando LOAD o VERIFY. ● No se ha activado la función de impresión. | <ul style="list-style-type: none"> ● Vuelva a almacenar el programa mediante el mando SAVE. ● Active la función de impresión. |
| SN error (error de sintáxis) | <ul style="list-style-type: none"> ● El formato del mando tiene un error. ● El número de línea tiene una fracción. ● Se especificó una tabla conteniendo 3 o más dimensiones. | <ul style="list-style-type: none"> ● Use el mando EDIT para que aparezca la línea en la cual ocurrió el error y luego corríjalo. ● Corrija el número de línea. ● Solamente se pueden especificar 2 dimensiones. |
| SO error (saturación de la memoria de retención temporal) | <ul style="list-style-type: none"> ● La memoria temporal de valores numéricos rebasó los 8 niveles. ● La memoria temporal de operadores rebasó los 20 niveles. ● La memoria temporal de caracteres rebasó los 10 niveles. | <ul style="list-style-type: none"> ● Simplifique o divida la expresión numérica para que el nivel de la memoria temporal quede dentro del límite especificado. ● Simplifique o divida la expresión de caracteres para que el nivel de la memoria temporal quede dentro del límite especificado. |
| ST error (error en la matriz de caracteres) | <ul style="list-style-type: none"> ● Se intentó asignarle a una variable de caracteres una serie de caracteres con una longitud mayor que la permisible. La longitud de las variables de caracteres permisible es la siguiente: <ul style="list-style-type: none"> – Variables de caracteres fijas Máx. 7 caracteres – Variables de caracteres registradas Máx. 16 caracteres – Variables de caracteres matriciales de longitud fija Máx. 16 caracteres – Variables de caracteres matriciales de longitud definida Máx. 79 caracteres | <ul style="list-style-type: none"> ● Cambie la variable por otra que pueda contener más caracteres. Disminuya el número de caracteres de la serie de caracteres que se vaya a asignar a una variable. Ponga atención cuando vaya a concatenar series de caracteres. |

| Mensaje | Significado | Medida para corregirlo |
|--|--|---|
| TM error (tipos de variables que no se corresponden) | <ul style="list-style-type: none"> ● En una sentencia para asignar valores, la parte izquierda y la derecha tienen un tipo de variable diferente. ● Los tipos de argumentos no se corresponden al asignar valores. | <ul style="list-style-type: none"> ● La parte izquierda ya la derecha de una sentencia para asignar valores deben ser o bien variables numéricas o bien variables de caracteres. ● Asigne un valor numérico a una variable numérica y una serie de caracteres a una variable de caracteres. |
| UL error (número de línea no definido) | <ul style="list-style-type: none"> ● El número de línea especificado no existe en una sentencia IF~THEN, GOTO o GOSUB. ● No hay ningún programa en el área del programa especificada mediante la sentencia GOTO o la GOSUB. | <ul style="list-style-type: none"> ● Introduzca un número de línea a donde se deba hacer un salto o cambie el número de línea especificado a donde se vaya a hacer un salto. ● Introduzca un área de programa a donde se vaya a hacer un salto o cambie la posición a donde se vaya a hacer un salto. |
| UV error (variable no definida) | <ul style="list-style-type: none"> ● Se usó una variable no definida. ● Se usó una variable de una tabla o lista sin especificarla mediante el mando DIM. ● El valor de ubicación (subscript) de una variable de tabla o lista excede el rango especificado mediante DIM. | <ul style="list-style-type: none"> ● Verifique el valor inicial de la variable. ● Especifique la matriz con el mando DIM al principio del programa. ● Cambie el valor de ubicación por un dentro del límite especificado. |
| VA error (error en las variables) | <ul style="list-style-type: none"> ● Se intentó registrar más de 40 variables. | <ul style="list-style-type: none"> ● Sólo se pueden usar hasta 40 variables registradas y de matriz. Revise los nombres de las variables usando LISTV y elimine las variables innecesarias usando CLEAR o ERASE. |

6-3 TABLA DE CODIGOS DE CARACTERES

| Hexa-decimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|--------------|------------|---------|---------|---|---|---|---|---|---|---|---------|---|---|---|----|---|
| 0 | 0 | 16 | Espacio | 0 | @ | P | ' | p | — | ┌ | Espacio | — | タ | ミ | 二 | × |
| 1 | 1 | (DEL) | ! | 1 | A | Q | a | q | — | ┐ | 。 | ア | チ | ム | 三 | 円 |
| 2 | (LINE TOP) | (INS) | " | 2 | B | R | b | r | — | └ | 「 | イ | ツ | メ | 十 | 年 |
| 3 | 3 | | # | 3 | C | S | c | s | ■ | ┌ | 」 | ウ | テ | モ | 十一 | 月 |
| 4 | 4 | | \$ | 4 | D | T | d | t | ■ | — | 、 | エ | ト | ヤ | ▲ | 日 |
| 5 | 5 | | % | 5 | E | U | e | u | ■ | — | ・ | オ | ナ | ユ | ▲ | 時 |
| 6 | (LINE END) | | & | 6 | F | V | f | v | ■ | — | ヲ | カ | ニ | ヨ | ▼ | 分 |
| 7 | (BEL) | (ENTER) | ' | 7 | G | W | g | w | ■ | — | ア | キ | ヌ | ラ | ▼ | 秒 |
| 8 | | (▲) | (| 8 | H | X | h | x | — | ┌ | イ | ク | ネ | リ | ♠ | 千 |
| 9 | (TAB) | |) | 9 | I | Y | i | y | — | ┐ | ウ | ケ | ノ | ル | ♥ | 市 |
| A | 10 | | * | : | J | Z | j | z | ■ | ┌ | エ | コ | ハ | レ | ◆ | 区 |
| B | (HOME) | | + | ; | K | [| k |] | ■ | ┐ | オ | サ | ヒ | ロ | ♣ | 町 |
| C | (CLS) | (→) | , | < | L | ¥ | l | ; | ■ | ┐ | ヤ | シ | フ | ワ | ● | 村 |
| D | (CR) | (←) | — | = | M |] | m | { | ■ | ┐ | ユ | ス | ヘ | ン | ○ | 人 |
| E | | (↑) | · | > | N | ^ | n | ~ | ■ | ┐ | ヨ | セ | ホ | " | / | ■ |
| F | | (↓) | / | ? | O | — | o | — | ┌ | ┐ | ツ | ソ | マ | 。 | \ | |

ESPECIFICACIONES

■ Modelo

PB-770

■ Funciones básicas de cálculo

Números negativos, exponentes, suma/resta/multiplicación/división con paréntesis (con una función que determina la prioridad de la secuencia-lógica algebraica verdadera). MOD.

■ Mandos

AUTO, CONT, DELETE, EDIT, LIST, LLIST, LOAD, NEW, NEW ALL, PASS, PROG, RUN, SAVE, SYSTEM, VERITY, ANGLE, BEEP, CHAIN, CLEAR, CLS, DIM, DRAW, DRAWC, END, ERASE, FOT-TO-STEP, NEXT, GET, GOSUB, RETURN, GOTO, IF-THEN-ELSE, INPUT, LET, LOCATE, POKE, PRINT, LPRINT, PUT, READ, DATA, RESTORE, REM, STOP, TRON, TROFF.

Mandos para cálculos estadísticos – STAT, STAT CLEAR, STAT LIST, STAT LLIST.

■ Funciones

SIN, COS, TAN, ASN, ACS, ATN, HYP SIN, HYPCOS, HYPTAN, HYPASN, HYPACS, HYPATN, SQR, LOG, LGT, EXP, ABS, INT, FRAC, SGN, ROUND, PI, RND, DEG, PEEK, ASC, CHR\$, VAL, STR\$, LEFT\$, RIGHT\$, MID\$, LEN, INKEY\$, DMS\$, HEX\$, TAB, USING, POINT, &H.

Funciones para cálculos estadísticos – CNT, COR, SUMX, SUMY, SUMX2, SUMY2, SUMXY, MEANX, MEANY, SDX, SDY, SDXN, SDYN, EOX, EOY, LRA, LRB.

■ Gama de cálculo

$\pm 1 \times 10^{-99} \sim \pm 9.999999999 \times 10^{99}$

(Los cálculos internos usan una mantisa de 12 dígitos)

■ Lenguaje de programación

BASIC

■ Capacidad de la memoria para programas

RAM: 8K bytes original, ampliable hasta 32K bytes (incluyendo los 1321 bytes del área de sistema)

ROM: Aproximadamente 32K bytes

■ Número de áreas de programas

Máximo 10 (P0–P9)

■ Número de memorias temporales

Subrutinas 12 niveles

Lazos FOR-NEXT 6 niveles

Valores numéricos 8 niveles

Operadores 20 niveles

- **Pantalla**
De cristal líquido
- **Capacidad de visualización**
20 dígitos x 4 líneas/32 x 160 puntos
- **Contenido de la exhibición**
Mantisa de 10 dígitos + exponente de 2 dígitos
- **Componente principal**
LSI
- **Consumo de energía**
0,1 W
- **Fuentes de alimentación**
Principal: 4 pilas tamaño AA.
Secundaria (para proteger la RAM): 1 pila de litio (CR1220).
- **Duración de las pilas**
Principal: Aproximadamente 100 horas en el tipo SUM-3 (operación continua).
Secundaria: Vea la página 12.
- **Interrupción automática de la corriente**
La corriente se apaga automáticamente unos 6 minutos después de la última operación.
- **Temperature ambiente**
De 0° C a 40° C
- **Dimensiones**
23 mm (alto) x 200 mm (ancho) x 88 mm (profundidad)
- **Peso**
315 grs. incluyendo las pilas.

| | | | |
|-------------------------------------|----------|---|--------------|
| A | | Coordenadas para los caracteres | 31 |
| Area para datos | 158 | | |
| ABS | 228 | D | |
| ACS | 219 | DATA | 205 |
| Alimentación principal | 15 | DEG | 241 |
| ANGLE | 151, 214 | DELETE | 130 |
| Apagado automático | 16 | Depuración | 40 |
| Areas de programa | 40 | DIM | 77, 162 |
| Asignación | 37 | Dimensión | 62 |
| ASC | 243 | Dirección inicial | 159 |
| ASN | 219 | DMS\$ | 259 |
| AUTO | 128 | DRAW | 99, 167 |
| ATN | 219 | DRAWC | 99, 167 |
| Atributos de archivo | 149 | | |
| B | | E | |
| BASIC | 34 | E | 67, 312 |
| BEEP | 154 | EDIT | 132 |
| C | | END | 47, 170 |
| CAPS | 23 | EOX | 85, 274 |
| Caracteres gráficos | 91 | EOY | 85, 274 |
| CHAIN | 155 | ERASE | 171 |
| CHR\$ | 91, 245 | Especificación del formato de secuencias de caracteres | 265 |
| Clave | 141 | Especificación del formato numérico | 264 |
| CLEAR | 157 | Especificación del modo de caracteres | 120 |
| CLS | 55, 161 | Especificación del modo gráfico | 120 |
| CNT | 85, 270 | EXP | 226 |
| Código ASCII | 243 | Expresión condicional | 46, 126, 186 |
| Códigos de caracteres | 328 | F | |
| Códigos para caracteres | 243 | FOR-NEXT | 55 |
| Coefficiente de correlación | 88 | FOR~TO~STEP/NEXT | 172 |
| Coefficiente de regresión lineal | 85, 274 | FRAC | 232 |
| CONT | 129 | Funciones especiales del mando PRINT | 202 |
| COR | 85, 270 | Funciones incorporadas | 313 |
| COS | 217 | G | |
| Con interface para cassette | 98 | GET | 177 |
| Control de contraste de la pantalla | 18 | GOSUB | 58, 180 |
| Control de la pantalla | 52 | GOTO | 47, 184 |
| Coordenadas gráficas | 31, 99 | | |

INDICE

GRADIENTES 153, 215
GRADOS 153, 215

H

HEX\$ 260
HYPACS 221
HYPASN 221
HYPCOS 221
HYPSIN 221
HYPATN 221
HYPTAN 221

I

IF~THEN~ELSE 46, 186
Inclusión de subrutinas 181
Incremento 172
Inicialización 44
INKEY\$ 257
INPUT 42, 45, 189
INT 230

L

Lazos 172
Lazos de esperá 175
LEFT\$ 252
LEN 256
LET 195
LGT 223
LIST 135
LLIST 135
LOAD 138
LOCATE 196
LOG 223
Logaritmo común 223
Logaritmo natural 223
LPRINT 120, 198
LRA 85, 274
LRB 274
LRG 85

M

Mandos para la impresora por
coordenadas 121
Matrices bidimensionales 63, 64
Matrices de variable numéricas 65
Matrices numéricas 65
Matrices unidimensionales 63
Matriz de caracteres 60
MEANX 85, 272
MEANY 85, 272
Precisión media 28
Mensajes 43
Mensajes de error 324
MID\$ 254
MOD 26, 27, 312
Modo de rastreo 212
Modificación del programa 51
Modo directo 22
Modo para funciones 24
Módulo de expansión RAM 17

N

NEW 142
NEW ALL 142
Nombre de archivo 126
Números aleatorios 239
Numéricas 29
Número de bytes 32

O

Operadores comparativos 28

P

PASS 143
PEEK 242
PI 238
POINT 99, 115, 266
POKE 197
Precisión de cálculos 26
Precisión media 67

| | | | |
|---|--------------|--|---------|
| Precisión simple | 28, 67 | SGN | 234 |
| PRINT | 42, 47, 198 | SIN | 214 |
| Prioridad de cálculo | 26 | SQR | 222 |
| Precisión media | 67 | STAT | 268 |
| Precisión simple | 28, 67 | STAT CLEAR | 86, 269 |
| PRINT | 42, 47, 198 | STAT LIST | 85, 269 |
| Prioridad de cálculo | 26 | STAT LLIST | 85, 269 |
| PROG | 145 | STOP | 210 |
| Programa principal | 180 | STR\$ | 250 |
| Protección de la memoria | 15 | Subrutina | 58, 180 |
| Punto y coma | 45, 47 | SUMX | 85, 271 |
| PUT | 203 | SUMX2 | 85, 271 |
| R | | SUMXY | 85, 271 |
| RADIANES | 153, 215 | SUMY | 85, 271 |
| READ | 205 | SUMY2 | 85, 271 |
| Regresión exponencial | 87 | SYSTEM | 60, 150 |
| Regresión logarítmica | 87 | T | |
| Regresión potencial | 88 | TAB | 52, 261 |
| REM | 44, 209 | TAN | 218 |
| RESTORE | 205 | Término constante de regresiones lineales | 85, 274 |
| RETURN | 180 | Tecla | 38 |
| RIGHT\$ | 253 | Tecla ENTER | 18, 38 |
| RND | 239 | Tecla RETURN | 18 |
| ROUND | 236 | Teclado numérico | 35 |
| RUN | 146 | Teclas de compaginación | 24 |
| Rutina principal | 58 | Teclas para corrección | 132 |
| S | | TROFF | 212 |
| Saltos | 46, 180, 184 | TRON | 212 |
| Salto condicionado | 186 | U | |
| Salto incondicional | 184 | USING | 53, 263 |
| SAVE | 147 | Uso del cursor | 97 |
| SDX | 85, 27 | V | |
| SDXN | 85, 273 | VAL | 247 |
| SDY | 85, 273 | Valor absoluto | 228 |
| SDYN | 85, 273 | Valor inicial | 172 |
| Secuencia de caracteres para el formato | 263 | Variable registrada | 190 |
| Sentencia múltiple | 53 | | |
| Serie sin caracteres | 164 | | |

INDICE

| | |
|--|--------|
| Variables de caracteres | 192 |
| Variables de caracteres matriciales | 29 |
| Variables de caracteres registradas | 29 |
| Variables de control | 76 |
| Variables fijas | 27 |
| Variables matriciales | 65 |
| Variables matriciales de caracteres | 77 |
| Variables numéricas | 60 |
| Variables numéricas | 192 |
| Variables registradas | 29, 30 |
| VERIFY | 152 |
| Visualización del número de dígitos | 68 |
| Visualización de patrones gráficos | 93 |
| Y | |
| &H | 275 |

