

Deseamos, antes que nada, agradecerle por la compra de uno de nuestros productos. La FX-702P es una calculadora programable de alto rendimiento, la cual incorpora microcircuitos para su uso en cálculos complejos. La más destacada característica de este instrumento es el uso del lenguaje para programas BASIC. Este lenguaje es del tipo conversacional y se usa para resolver problemas. La operación es muy simple, aún para principiantes. Por otro lado, la programación es bastante fácil gracias al uso de los comandos de una tecla, los cuales permiten obtener un alta eficiencia en el uso del teclado.

Las funciones para cálculos que este instrumento posee se dividen en dos grandes grupos:

1. Cálculos Manuales
2. Cálculos por Programa

Esta calculadora no sólo realiza cálculos por programa de alto nivel como una computadora, sino que también fue diseñada para usarse como una calculadora científica de fácil operación.

INDICE

Antes del Uso	3
Precauciones en el Uso	
Fuente de Alimentación y Reemplazo de las Baterías	
Capítulo 1 Nomenclatura y Operación de Cada Sección	4
1-1 Nomenclatura de Cada Sección	4
1-2 Lectura de la Pantalla	7
1-3 Ajuste del Contraste	7
Capítulo 2 Antes de Comenzar con los Cálculos	8
2-1 Secuencia de Prioridades de Cálculo	8
2-2 Número de Unidades de Entrada/Salida y de Dígitos de Cálculo	8
2-3 Cálculos Básicos	9
2-3-1 Símbolos de Cálculos y Comandos de Funciones	9
2-3-2 Obtención del Resultado del Cálculo Previo	10
2-3-3 Mensajes de Error	10
2-4 Operación de las Teclas	11
2-5 Expansión de la Memoria	12
2-6 Apagado Automático (Auto Power Off)	13
2-7 Exceso de Números	13

Este manual de instrucciones explica como usar la calculadora. Se ruega leerlo cuidadosa y minuciosamente antes de hacer uso del instrumento. Dominar a fondo el uso de cada una de las funciones y usarlas cuidadosamente para darle a la computadora una larga vida de servicio.

Capítulo 3 Cálculos Manuales	14
3-1 Explicación Sobre los Cálculos Manuales	14
3-2 Operación en Cálculos Manuales	14
3-3 Ejemplos de Cálculos Manuales	15
3-3-1 Cálculos Fundamentales	15
3-3-2 Cálculos de Funciones	17
3-3-3 Cálculos Estadísticos	21
Capítulo 4 Cálculos por Programa	25
4-1 Generalidades de un Programa	25
4-2 Principios de un Programa	28
4-3 Escritura y Ejecución de Programas	29
4-4 Compaginación de Programas	32
4-5 Comandos de Programa	39
4-5-1 Saltos y Lazos	39
4-5-2 Matrices	53
4-5-3 Comandos de Entrada/Salida	57
4-5-4 Funciones de Caracteres	63
4-5-5 Subrutinas	64
4-5-6 Funciones Generales	68
4-5-7 Procesos Estadísticos	69
4-5-8 Contraseñas	74
4-5-9 Especificaciones Opcionales	75
Lista de Mensajes de Error	79
Lista de Comandos de Programa	80
Especificaciones	83

Antes del Uso

Esta calculadora llega a sus manos como resultado de nuestra tecnología electrónica altamente desarrollada. Se han empleado en su fabricación un estricto control de calidad y minuciosos procesos de inspección. Se ruega seguir al pie de la letra las siguientes instrucciones y advertencias para asegurar una larga vida útil del aparato.

■ Precauciones en el Uso

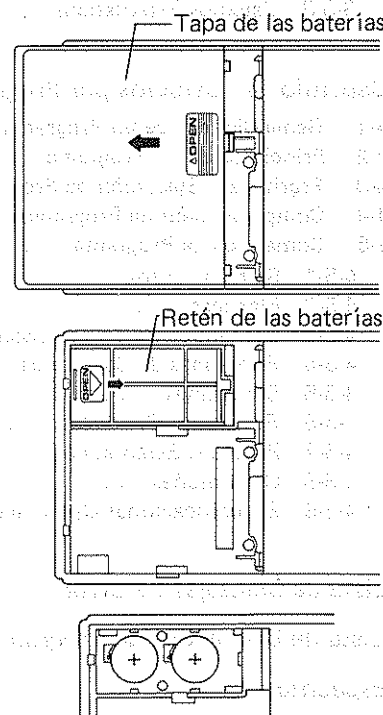
- Esta calculadora está compuesta por piezas electrónicas de alta precisión. Se recomienda no intentar desarmarla. Evítense también todo tipo de golpes. No deberá permitirse que la calculadora sufra cambios bruscos de temperatura. No se deberá almacenarla ni dejarla en lugares húmedos, con alta temperatura o excesivo polvo. En épocas invernales o de baja temperatura, la respuesta de la pantalla puede ser más lenta que lo normal o, en el peor de los casos, puede que ésta no funcione. Para solucionar este problema, llevar la calculadora donde la temperatura sea más alta.
- No usar otros equipos que no sean los opcionales.
- Mientras la calculadora está realizando cálculos, se visualizará en la pantalla un guión ("—"). En este caso, la operación de las teclas será efectiva sólo para una parte del teclado, por lo cual deberá prestarse en todo momento especial atención a lo visualizado en la pantalla y hacer uso de las teclas con sumo cuidado.
- En cuanto a las baterías, deberán cambiarse cada 2 años, aún si no se usa la calculadora con mucha frecuencia. Si se usan baterías gastadas, éstas podrán sulfatarse y dañar la calculadora. No dejar, entonces, baterías gastadas en la calculadora.
- Para limpiar la calculadora, úsese un paño suave y seco o, como alternativa, húmedo con agua y un poco de algún detergente suave. No deberá usarse ningún tipo de removedor como thinner o bencina.
- En caso que su aparato funcione mal, póngase en contacto con el comercio que le vendió su calculadora o con el distribuidor más cercano.
- Antes del mantenimiento, léase este manual de instrucciones nuevamente y verifíquese también la fuente de alimentación o si hay errores de operación o en el programa.

■ Fuente de Alimentación y Reemplazo de las Baterías

Esta calculadora usa 2 baterías de litio (CR2032). Si aún cuando el control de contraste está en su posición máxima la visualización en la pantalla es muy pobre, significa que las baterías están gastadas. En este caso, reemplazar las mismas tal como se ilustra en la figura de abajo. Existe la posibilidad de que se presenten errores en caso de usarse la calculadora con baterías gastadas. Aún cuando la calculadora esté funcionando normalmente, cambiar las baterías cada 2 años.

● Reemplazo de las baterías

- Después de poner la llave de encendido en OFF, deslizar la tapa de las baterías y sacarla desde la parte de atrás del aparato.
- Empujar el retén de las baterías en la dirección que indica la flecha y sacarlo.
- Extraer las dos baterías gastadas (esta operación será más fácil si se golpea suavemente la calculadora con el receptáculo de las baterías hacia abajo).
- Si la superficie de las baterías nuevas está cubierta con polvo, podrá producirse un falso contacto. Limpiar bien con un paño seco. Instalar las baterías con el positivo hacia arriba.
- Empujar las baterías hacia adentro con el retén de las mismas y deslizar la tapa para cerrarla.



Precaución

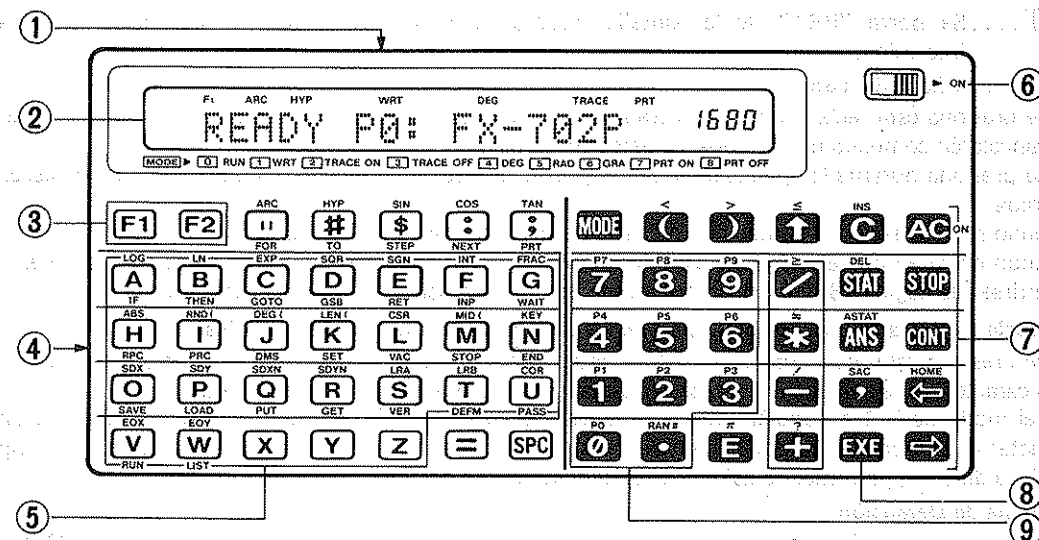
Asegurarse de llevar a cabo las siguientes operaciones después del reemplazo de las baterías.

Colocar la llave de encendido en ON.

MODE 1 CLR ALL EXE
MODE 0 F1 SAC F2 DEFN 0 EXE VAC EXE

Capítulo 1

Nomenclatura y Operación de Cada Sección



① Dial para el Contraste de la Pantalla

② Pantalla

③ Teclas de Funciones

④ Conector para el Adaptador

⑤ Teclas Alfabéticas

⑥ Llave de Encendido

⑦ Teclado

⑧ Tecla de Ejecución

⑨ Teclas Numéricas y Tecla del Punto Decimal

1-1 Nomenclatura de Cada Sección

Cada tecla posee 3 diferentes funciones.

Para la función impresa en la tecla, presionar ésta directamente. Para la función impresa por encima de la tecla, presionar [F1] y luego la tecla pertinente. En cuanto a la función impresa debajo de la tecla, ésta se hará efectiva si se presiona [F2] y luego la tecla correspondiente.

Ejemplo

LOG ... Modo de la 1ra. función

[A] ... Modo directo

[IF] ... Modo de la 2da. función

[F1] Tecla de la 1ra. Función

Si se presiona esta tecla, se selecciona la 1ra. función (se visualiza "F1"), permitiéndose el uso de la 1ra. función impresa por encima de cada tecla.

[F2] Tecla de la Segunda Función

Cuando se presiona esta tecla, se selecciona el modo de la 2da. función (se visualiza "F2"), permitiéndose el uso de la función impresa debajo de cada tecla.

MODE Tecla de Modo

La modalidad de la calculadora y la unidad de medición angular se designan presionando la tecla [MODE] y las teclas numéricas de [0] a [8].

MODE 0 ... Se visualiza "RUN" en la pantalla. Pueden realizarse cálculos manuales y por programa.

MODE 1 ... Se visualiza "WRT" en la pantalla. Se pueden escribir, verificar y compaginar programas.

MODE 2 ... Se visualiza "TRACE" en la pantalla. Se puede llevar a cabo una comprobación de la ejecución por seguimiento secuencial. (Para más detalles, ver la página 38).

- MODE** [3] Borra el "TRACE" visualizado en la pantalla. Se cancela la función de comprobación por seguimiento secuencial de la ejecución.
- MODE** [4] Se visualiza "DEG" en la pantalla. La unidad de medida angular designada son "grados".
- MODE** [5] Se visualiza "RAD" en la pantalla. Se designan "radianes" como la unidad de medida angular.
- MODE** [6] Se visualiza "GRA". Se designan "gradientes" como la unidad de medición angular.
- MODE** [7] Se visualiza "PRT" en la pantalla. Se puede imprimir en una impresora, si es que hay alguna conectada. (Por detalles, ver la página 78).
- MODE** [8] Se borra "PRT" de la pantalla (cuando este está visualizado) y se cancela la salida de impresión.

AC Tecla de Borrado Total

- Si se presiona esta tecla cuando se entran datos mediante el teclado, la pantalla se borrará totalmente y se cancelarán completamente todos los datos entrados hasta el momento.
- Si se presiona durante la operación de un programa, este se detendrá y retornará a la condición espera de entrada.
- Cuando se visualiza en la pantalla un mensaje de error, se usa para cancelar el mismo.
- Cuando entra en operación la función AUTO POWER OFF (función automática de ahorro de energía; remitirse a la página 13) y la pantalla se apaga, se presiona para encender nuevamente el aparato.

INS Tecla de Borrado (tecla de anulación de los caracteres de entrada)/Tecla de Inserción

- Si se presiona esta tecla cuando se entran datos por el teclado, se cancelará el carácter previo al cursor. Los caracteres a la derecha del cursor se desplazarán una posición hacia la izquierda.
- En el modo de la 1ra. función, esta tecla se transforma en la tecla de inserción. Cuando se la presiona, el carácter que se encuentra por debajo del cursor y aquellos a la derecha de este se desplazarán un espacio hacia la derecha, permitiendo así la inserción de un nuevo carácter.

STOP Tecla de Detención

Si se presiona esta tecla durante la operación de un programa, se visualizará "STOP" en la pantalla. La operación del programa se detendrá al final de una línea. Si se presiona durante la verificación de la ejecución por seguimiento secuencial, se visualizarán en pantalla el número de programa, el número de líneas y el contenido del programa.

CONT Tecla de Continuación

Si se presiona cuando se ha detenido un programa mediante la tecla **STOP** o una sentencia de detención, el programa comenzará a operar nuevamente desde la siguiente sentencia. Durante el seguimiento secuencial de ejecución, se presiona para avanzar a la siguiente sentencia.

DEL Tecla para Cálculos Estadísticos/Tecla de Anulación

- Esta tecla se utiliza para la entrada de datos usados en la realización de cálculos estadísticos.
- Ejemplo:** Desviación estándar . . . x [STAT]. Cálculos de regresión . . . x , y [STAT].
- En el modo de la 1ra. función, presiónese esta tecla para anular datos cuando se realizan cálculos estadísticos.

ANS Tecla de Resultados/Tecla de Resultados de Cálculos Estadísticos

- Esta tecla se usa para obtener el resultado del cálculo previo en cálculos manuales o por programa.
- En el modo de la 1ra. función, esta tecla se usa para visualizar el resultado de cálculos estadísticos (número de datos, sumatoria de x , sumatoria de y , sumatoria del cuadrado de x , sumatoria del cuadrado de y o producto de $x \cdot y$).

HOME Tecla del Cursor/Tecla para la Posición de Origen

- Pulsando esta tecla una vez, se desplaza el cursor una posición hacia la izquierda.
- En el modo de la 1ra. función, se presiona para llevar el cursor a la posición de origen (por debajo del primer carácter de entrada).

⇒ Tecla del Cursor

Esta tecla hace desplazar el cursor una posición hacia la derecha cada vez que se la pulsa.

EXE Tecla de Ejecución

- Esta tecla se presiona en lugar de "=" para obtener el resultado de un cálculo manual.
- En el modo "WRT", esta tecla se utiliza para escribir programas línea por línea. No se podrá escribir si no se presiona esta tecla.
- En el modo "RUN", esta tecla se presiona para poder entrar información durante la operación de un programa.

SAC Tecla de la Coma/Tecla de Borrado de la Memoria para Cálculos Estadísticos

- Esta tecla se presiona para escribir la coma.
- En el modo de la 1ra. función, se utiliza para borrar la memoria para cálculos estadísticos.

() Teclas de Paréntesis/Teclas de Comparación

- Estas teclas se utilizan para entrar paréntesis en cálculos donde éstos se utilizan.
- En el modo de la 1ra. función, estas teclas se usan para hacer comparaciones.

↑ ↓ × ÷ Teclas para Comandos de Cálculos/Teclas Comparativas/Teclas de Caracteres

- **+** **-** ***** **/** Símbolos para suma, resta, multiplicación y división. Presiónese el que corresponda a la función deseada.
- Presionar **↑** para obtener potencias.
- En el modo de la 1ra. función, **↑** **↓** ***** se presionan para hacer comparaciones. Para obtener resultados de factoriales, presiónese **!**.
- Para escribir un signo de pregunta, presionar **?**.

P7 P8 P9 Teclas Numéricas/Teclas de Número de Programa

- Se presionan para entrar números en los cálculos. Para entrar el punto decimal, presiónese **.**.
- En el modo de la 1ra. función, estas teclas son para designar el número de programa. Cuando se escribe, se puede iniciar un programa.

LOG A	LN B	EXP C	SQR D	SGN E	INT F	FRAC G
IF ABS H	THEN RNDI I	GOTO DEG J	GSB LEN K	RET CSR L	INP MID M	WAIT KEY N
RPC SDX O	PRC SDY P	DMS SDXN Q	SET SDYN R	VAC LRA S	STOP LRB T	END COR U
SAVE EOX V	LOAD EOY W	PUT X	GET Y	VER Z		
RUN	LIST					

Teclas Alfabéticas/Comandos de una Tecla

- Si se presionan estas teclas cuando se escriben programas, comandos o comandos de funciones, se visualizarán en la pantalla las letras correspondientes a las teclas pulsadas.
- En el modo de la 1ra. función, se visualizarán los comandos de función de una tecla.
- En el modo de la 2da. función, se visualizarán los comandos de programa de una tecla.

= Tecla de Igualdad

Presiónese esta tecla para escribir el símbolo de sustitución o el símbolo de igual.

SPC Tecla de Espacio

Presiónese esta tecla para incluir un espacio entre caracteres durante la entrada de información.

ARC Tecla de Secuencias Ordenadas/Comandos de una Tecla

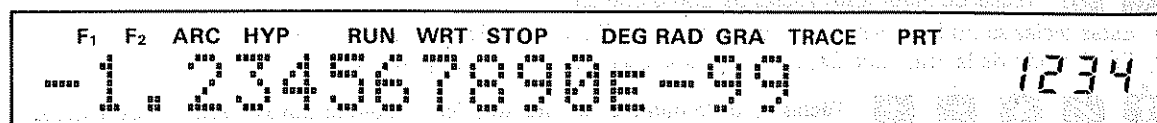
- Presionar esta tecla al comienzo y final de toda secuencia ordenada de caracteres cuando se entran o visualizan constantes de caracteres.
- En el modo de la 1ra. función, se visualizarán los comandos de función de una tecla.
- En el modo de la segunda función, se visualizarán los comandos de programa de una tecla.

HYP SIN COS TAN Teclas de Caracteres/Comandos de una Tecla

- Se presionan para usar los símbolos escritos sobre las teclas.
- En el modo de la 1ra. función, se visualizarán los comandos de función de una tecla.
- En el modo de la 2da. función, se visualizarán los comandos de programa de una tecla.

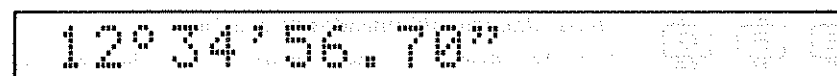
1-2 Lectura de la Pantalla

Pantalla con Patrón de Puntos

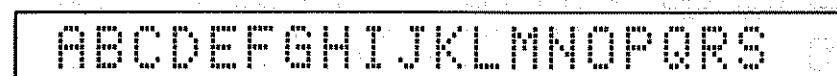


La pantalla visualiza los valores y resultados de los cálculos. Cada caracter está compuesto por 5 puntos de alto y 7 puntos de ancho. Se puede visualizar un máximo de 20 números o caracteres. (Cero: ). Si una fórmula o sentencia posee más de 20 caracteres, los números o letras se desplazarán hacia la izquierda. Se puede entrar un máximo de 62 caracteres. Las cuatro posiciones a la derecha de la pantalla muestran los pasos remanentes del programa. Durante la realización de cálculos, se visualizará un signo menos a la derecha de los 4 dígitos que se encuentran a la extrema derecha de la pantalla. Se visualizan también las unidades de medida angular "DEG", "RAD" y "GRA" o "F1", "F2" (cuando se presiona la tecla  o ). También se visualizan "ARC", "HYP" (cuando se presionan las teclas  o ), "RUN" (modo RUN), "WRT" (modo WRT), "TRACE" (modo TRACE), "PRT" (modo PRT), etc. Estos símbolos se visualizan según la situación. Por otro lado, las letras y símbolos sexagesimales o alfabéticos se visualizan de la siguiente manera:

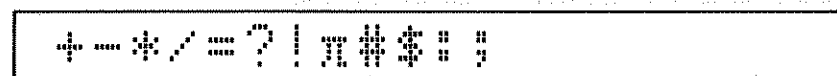
- **Ejemplo de sexagesimales**



● Ejemplo de letras

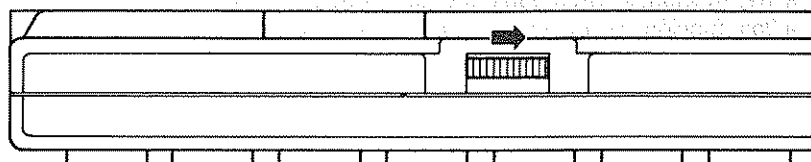


● Ejemplo de símbolos



1-3 Ajuste del Contraste

El ajuste del contraste se realiza haciendo uso de la perilla de control que se encuentra por encima de la calculadora.



Girar el control en la dirección que indica la flecha para aumentar el contraste de la pantalla y en sentido contrario para desvanecerlo. Este control de contraste es para compensar la descarga progresiva de las baterías y, además, para ajustar el contraste de la pantalla a gusto personal.

Capítulo 2

Antes de Comenzar con los Cálculos

Tanto los cálculos manuales como por programa se realizan en el modo "RUN" (visualización de "RUN" = ). Si el cálculo a realizarse no tiene ninguna relación con ángulos, se puede omitir la visualización de la unidad de medida angular "DEG", "RAD" o "GRA".

2-1 Secuencia de Prioridad del Cálculo (Lógica Algebraica de Verdad)

- Esta calculadora determina automáticamente la prioridad de cálculo y realiza los cálculos bajo dicha secuencia. La secuencia de prioridad de cálculo es la siguiente:

- ① Funciones (sen, cos, tan, etc.)
- ② Potencias, Factoriales
- ③ $\times$, $\div$ ($*$, $/$)
- ④ $+$, $-$

Quando el nivel de prioridad es el mismo, se comienza a calcular desde el principio (desde la izquierda). Si se usan paréntesis, tendrán mayor prioridad los cálculos que se encuentran dentro del mismo.

2-2 Número de Unidades de Entrada/Salida y de Dígitos de Cálculo

Las unidades de entrada/salida de esta calculadora están compuestas por 10 posiciones de mantisa y 2 posiciones de exponentes. El rango es de 1.0^{-99} hasta $\pm 9.999999999 \times 10^{+99}$.

El número de posiciones de entrada/salida es de 10 posiciones de mantisa y 2 posiciones de exponente. Sin embargo, el número de posiciones de cálculos internos y el número de posiciones almacenadas en la memoria es de 12 posiciones de mantisa y 2 posiciones de exponente. Cuando la entrada o el valor de cálculo es mayor de 10 posiciones, se escribirán hasta 12 posiciones. Se ignorará todo lo que sea mayor que esto. La salida son 10 posiciones de mantisa.

Ejemplo: $1.23456789123 \times 100 = 123.4567891$

Quando el resultado de un cálculo es mayor que 10^{10} ó menor que 10^{-3} , se lo visualizará automáticamente en forma exponencial.

Ejemplo: $1234567890 \times 10 =$ $1.23456789 \text{E } 10$

(La visualización exponencial aparece en la pantalla después de la visualización de la mantisa y con el signo exponencial "E").

Ejemplo: $1.234 \div 10000 =$ $1.234E-04$

2-3 Cálculos Básicos

2-3-1 Símbolos de Cálculos y Comandos de Funciones

Usando los símbolos para cálculos BASIC, los símbolos + y - son iguales que los símbolos comunes para suma y resta, pero para la multiplicación y división se usan los símbolos * y / respectivamente.

Por ejemplo:

$$2+3-4 \times 5 \div 6$$

se expresa de la siguiente manera

$$2+3-4*5/6$$

Este instrumento incorpora, además, las siguientes funciones.

Nombre de la Función

Función Trigonométrica

sen x

cos x

tan x

Función Trigonométrica Inversa

sen⁻¹ x

cos⁻¹ x

tan⁻¹ x

Función Hiperbólica

senh x

cosh x

tanh x

Función Hiperbólica Inversa

senh⁻¹ x

cosh⁻¹ x

tanh⁻¹ x

Raíz Cuadrada

√x

Función Exponencial

e^x

Logaritmo Natural

ln x

Logaritmo Común

log x

Factoriales

x!

Conversión a Enteros

INT x

Anulación de la Parte Entera

FRAC x

Conversión a Valor Absoluto

|x|

Cambio a Símbolo

Número Positivo → 1

0 → 0

Número Negativo → -1

Grados Minutos Segundos

(Sexagesimal → Decimal)

Grados Minutos Segundos

(Decimal → Sexagesimal)

Conversión de Coordenadas

(Rectangulares → Polares)

Conversión de Coordenadas

(Polares → Rectangulares)

* Las coordenadas x, y y r, θ pueden substituirse por las variables X, Y.

Redondeo

(se redondea 10^y de x)

Números Aleatorios

Cálculos Estadísticos

Números de datos n

Desviación estándar de x σ_{n-1}

Desviación estándar de y σ_{n-1}

Desviación estándar de x σ_n

Desviación estándar de y σ_n

Promedio de x $\bar{x}$

Promedio de y $\bar{y}$

Sumatoria de x Σx

Formato

SIN x [F1] SIN

COS x [F1] COS

TAN x [F1] TAN

ASN x [F1] ARC SIN

ACS x [F1] ARC COS

ATN x [F1] ARC TAN

HSN x [F1] HYP SIN

HCS x [F1] HYP COS

HTN x [F1] HYP TAN

AHS x [F1] ARC HYP SIN

AHC x [F1] ARC HYP COS

AHT x [F1] ARC HYP TAN

SQR x [F1] SQR

EXP x [F1] EXP

LN x [F1] LN

LOG x [F1] LOG

x! [F1] x!

INT x [F1] INT

FRAC x [F1] FRAC

ABS x [F1] ABS

SGN x [F1] SGN

DEG (x, y, z) [F1] DEG

DMS x [F2] DMS

RPC x, y [F2] CH RPC

PRC x, y [F2] PRC

RND (x, y) [F1] RND

RAN # [F1] RAN#

CNT -

SDX [F1] SDX

SDY [F1] SDY

SDXN [F1] SDXN

SDYN [F1] SDYN

MX -

MY -

SX -

Sumatoria de y Σy

Sumatoria de x al cuadrado Σx^2

Sumatoria de y al cuadrado Σy^2

Sumatoria del producto de datos Σxy

Término constante A

Coefficiente de regresión B

Coefficiente de correlación r

Valor estimado de x $\hat{x}$

Valor estimado de y $\hat{y}$

SY

SX2

SY2

SXY

LRA [F1] LRA

LRB [F1] LRB

COR [F1] COR

EOX y [F1] EOX

EOY x [F1] EOY

2-3-2 Obtención del Resultado del Cálculo Previo

El resultado de todo cálculo manual o por programa se almacena hasta que se haya ejecutado el siguiente cálculo. Este resultado puede visualizarse presionando la tecla [ANS].

Ejemplo: 741+852=1593
2431-1593=838

Operación:

7 4 1 + 8 5 2

[EXE]

2 4 3 1 - [ANS]

[EXE]

741+852

1593

2431- 1593

838

Además, el valor numérico visualizado después del cálculo puede usarse como está en el siguiente cálculo.

Ejemplo: (Continuación del anterior)
838×2=1676

Operación:

* 2

[EXE]

838*2

1676

2-3-3 Mensajes de Error

Si la fórmula o sentencia de substitución no se escribe correctamente de acuerdo al lenguaje BASIC, o si se excede el rango de cálculo, habrá un error en la operación y se visualizará en la pantalla un mensaje de error. Los siguientes son los mensajes de error que aparecen en los cálculos manuales.

ERR-2

(error en la estructura de la sentencia)

ERR-3

(error matemático)

Los siguientes mensajes de error se visualizan durante los cálculos por programa.

ERR-2 IN P0-10

↑ ↑
área de programa número de línea

(Significa que hay un error en la estructura de la sentencia que se encuentra en línea 10 del programa P0).

ERR-3 IN P2-20

(Significa que hay un error matemático en la línea 20 del área de programa P2).

En cuanto al significado de los mensajes de error, ver la Página 79 correspondiente a la lista de mensajes de error.

* Cuando se excede el rango de cálculo ($\pm 9.999999999E+99$), se produce una condición de desborde (superación de la capacidad), apareciendo en la pantalla un mensaje de error. Además, por debajo de $1.0E-99$, se presenta una condición de generación de un valor inferior al mínimo aceptable; en este caso, el resultado del cálculo será "0".

2-4 Operación de las Teclas

Antes que nada, encender la calculadora.

Se visualiza en la pantalla "READY P0", lo que significa que la calculadora está lista para recibir datos.

1. Entrada por Teclas

● Entrada de Letras

Ejemplo: Entrar ABC

Operación:

A B C

ABC

Ejemplo: Entrar SIN

Operación:

S I N (O F1 SIN)

SIN

* Pueden usarse ambas alternativas, ya sea el comando de una tecla o el comando alfabético.

● Entrada de Números

Ejemplo: Entrar 123

Operación:

1 2 3

123

Ejemplo: Entrar 96.3

Operación:

9 6 . 3

96.3

● Entrada de símbolos

Ejemplo: S#?

Operación:

\$ # ?

\$ # ?

● Entrada de números con exponentes

Ejemplo: Entrar 7.896×10^{15}

Operación:

7 . 8 9 6 E 1 5

7.896E15

Símbolo exponencial

Ejemplo: Entrar -2.369×10^{-45}

Operación:

- 2 . 3 6 9 E - 4 5

-2.369E-45

2. Modificación de la Entrada (Corrección, Anulación e Inserción)

● Corrección

Llevar el cursor a la posición donde se desea llevar a cabo la corrección (usar $\leftarrow$ $\rightarrow$). Presionar la tecla de la letra, número o símbolo deseado.

Ejemplo: Modificar A\$ a B\$.

Operación: Mover el cursor dos posiciones hacia la izquierda. $\leftarrow$ $\leftarrow$

Presionar la tecla B.

Ejemplo: Cambiar "LIST" a "RUN".

Operación: Mover el cursor 4 posiciones hacia la izquierda. $\leftarrow$ $\leftarrow$ $\leftarrow$ $\leftarrow$

Presionar RUN o F2 $\rightarrow$ RUN.

A\$

A\$

B\$

LIST

LIST

RUN

* En este ejemplo, para mover el cursor hasta la posición por debajo de la primer letra, presionar F1 $\rightarrow$ HOME.

● Anulación

Colocar el cursor un espacio a la derecha del carácter a anularse. Presionar la tecla $\rightarrow$. Cada vez que se pulse esta tecla, se anula el carácter que se encuentre a la izquierda del cursor.

Ejemplo: Para anular "I" de "SIIN".

SIIN

Operación: Llevar el cursor un espacio hacia la izquierda.

(Usar $\leftarrow$)

SIIN

Presionar la tecla $\rightarrow$.

SIN

Ejemplo: Anular la "X" de "INP X, Y".

INP X, Y

Operación: Mover el cursor un espacio hacia la izquierda.

(Usar $\leftarrow$)

INP X, Y

Presionar $\rightarrow$ $\rightarrow$. Se anulan, de este modo, los dos caracteres.

INP Y

Ejemplo: Anular "B" de "PRT A, B".

PRT A, B

Operación: Presionar $\rightarrow$ $\rightarrow$. De este modo se anulan ambos caracteres.

PRT A

● Inserción

Llevar el cursor a la derecha de la posición donde se desea insertar el nuevo carácter. Presionar F1 $\rightarrow$ INS. Por cada pulsación aparecerá un espacio. Presionar la tecla correspondiente al carácter que se desea insertar en el espacio abierto.

Ejemplo: Inserción de "\$" en "T=A\$" para que quede "T\$=A\$".

T=A\$

Operación: Mover el cursor 3 posiciones hacia la izquierda.

(Usar $\leftarrow$ $\leftarrow$ $\leftarrow$)

T=A\$

Presionar F1 $\rightarrow$ INS. Aparecerá debajo del cursor un espacio.

T_ =A\$

Presionar la tecla \$.

T\$=A\$

Ejemplo: Modificar "PRT X" a "PRT SIN X".

PRT X

Operación: Mover el cursor una posición hacia la izquierda. (Usar $\leftarrow$)

PRT X

Abrir 3 espacios (usar F1 $\rightarrow$ INS F1 $\rightarrow$ INS F1 $\rightarrow$ INS).

PRT _ X

Presionar SIN.

PRT SINX

2-5 Expansión de la Memoria

Se dispone, bajo condiciones normales, de 26 memorias (variables). Tendríamos, en este caso, 1680 pasos. La memoria puede extenderse hasta un máximo de 226 memorias. Esta expansión se lleva a la realidad haciendo una conversión a 10 unidades por memoria con 8 pasos para cada una de ellas.

Número de memorias	Número de pasos de programa
26 (A ~ Z)	1680 pasos
36 (A ~ Z, A0 ~ A9)	1600 pasos
46 (A ~ Z, A0 ~ A9, B0 ~ B9)	1520 pasos
56 (A ~ Z, A0 ~ A9, B0 ~ B9, C0 ~ C9)	1440 pasos
...	...
226 (A ~ A, A0 ~ A9, ..., T0 ~ T9)	80 pasos

Para expandir la memoria se usa el comando DEFM. Por otro lado, la expansión se designa en unidades de a 10, y cada una de estas 10 unidades formando un grupo. Para expandir a 10, usar "1". Para expandir a 50, usar "5".

Ejemplo: Para expandir la memoria a 56 incrementando 30.

Operación: Se lleva a cabo usando el modo RUN (**MODE** **0**) o el modo WRT (**MODE** **1**).

DEFM 3 **EXE** **VAR: 56 PRG: 1440**
 número de memorias número de pasos

* El DEFM puede entrarse presionando **D** **E** **F** **M** o **F2** **DEFM**.

Ejemplo: Para expandir la memoria a 166 incrementando 140.

Operación: **DEFM 14** **EXE** **VAR: 166 PRG: 560**

* Si ya se encuentran en uso un gran número de programas, y en caso que el área remanente sea insuficiente, aparecerá un mensaje para proteger a tales programs ya en uso.

ERR-1 Indica que el número de pasos es insuficiente.

* Como la variable de caracteres exclusiva (\$) es una memoria especial, no se la cuenta en la designación de la capacidad de la memoria.

2-6 Apagado Automático (AUTO POWER OFF)

Con el fin de ahorrar energía cuando la llave de encendido está en ON, la calculadora se apagará automáticamente después de aproximadamente 8 minutos. (excepto durante cálculos por programa). En este caso, vuelva a encender la calculadora o presione la tecla **AC** lo que hará que se vuelva a encender.

* El contenido de la memoria y el programa no se borrarán cuando se apaga la unidad (OFF). La designación del modo angular y la modalidad de la calculadora se borran ("WRT", "TRACE", "PRT", etc).

2-7 Exceso de Números

Cuando se visualizan resultados de cálculos manuales o por programa, se puede cancelar o redondear la 11ma. posición. Esto se lleva a cabo usando el comando "SET".

SET { 0 } ("0" para anular
 { 5 } "5" para redondear)

Ejemplo: $11 \div 3 = 3.6666 \dots$

Operación: Para anular **SET** **0** **EXE** **1** **1** **3** **EXE** **3.666666666**

* Esta anulación es efectiva sólo para la pantalla. Los cálculos internos reales se llevan a cabo con redondeo.

(Continuación) **ANS** **3.666666667**

• **Redondeo** **SET** **5** **EXE** **1** **1** **3** **EXE** **3.666666667**

Este comando SET se pone automáticamente en la condición de redondeo cuando se apaga la calculadora manualmente o por medio de la función automática de apagado. Todos los cálculos de este manual se realizarán usando "SET5".

Capítulo 3

Cálculos Manuales

3-1 Explicación Sobre los Cálculos Manuales

En los cálculos manuales, a diferencia de los cálculos por programa, las operaciones no se realizan automáticamente usando fórmulas previamente almacenadas. Se denominan cálculos manuales aquellas operaciones para desplazar números de derecha a izquierda, hacer cálculos manualmente y obtener el contenido de variables.

3-2 Operación en Cálculos Manuales

■ La suma, resta, multiplicación y división usan lógica algebraica de verdad. **+**, **-**, ***** (X), **÷** ($\div$) y **EXE** (=) son los símbolos usados. La tecla **EXE** se utiliza para obtener el resultado de los cálculos y cumple la misma función que el signo igual ("=").

Ejemplo: $12+36-9 \times 5 \div 4 = 36.75$

Operación: **1** **2** **+** **3** **6** **-** **9** ***** **5** **÷** **4** **EXE** **12+36-9*5/4**
36.75

■ Los cálculos de funciones incluyen suma, resta, multiplicación y división y poseen la misma lógica algebraica de verdad. Los datos se escriben después de los comandos de función.

Ejemplo: $\log 1.23 = 0.08990511144$

Operación: **LOG** **1** **.** **2** **3** **EXE** **LOG1.23**
0.08990511144

* En este manual se escriben las letras y números sin sus recuadros.

Ejemplo: **SIN** (**1** **5** **+** **8**) **EXE** $\rightarrow$ **SIN** (**15+8**) **EXE**

* Los comandos de funciones y programas se pueden escribir usando los comandos de una tecla o los comandos alfabéticos correspondientes. No obstante, en este manual, se lo hará usando los comandos alfabéticos.

■ Aquellos cálculos que emplean la memoria para almacenar valores numéricos o resultados de cálculos y para obtener totales harán uso de las variables. Estas variables se representan por letras (A-Z) o letras y números (0-9) combinados (cuando se expandieron las memorias a más de 26). Para colocar dentro de una variable un valor numérico o un resultado, úsense las substituciones.

Ejemplo: Almacenar 1234 como la variable A.

Operación: **A** **=** **1** **2** **3** **4** **EXE** **A=1234**

Ejemplo: Sumar a la variable K el resultado de 23×56 .

Operación: $K \oplus K + 23 \times 56$ $K = K + 23 \times 56$

Este es el método manual para realizar la misma operación que una sustitución de una sentencia en programas.

- Para realizar correcciones antes de presionar la tecla **EXE**, mover el cursor a la posición a corregirse y pulsar la tecla de corrección (Ver el Capítulo 2).
- Presionar la tecla **AC** para cancelar todo lo visualizado en la pantalla.

3-3 Ejemplos de Cálculos Manuales

3-3-1 Cálculos Fundamentales

■ Suma, resta, multiplicación y división

Ejemplo: $23 + 4.5 - 53 = -25.5$

Operación: $23 \oplus 4.5 \ominus 53$ **EXE** -25.5

Ejemplo: $56 \times (-12) \div (-2.5) = 268.8$

Operación: $56 \times (-12) \div (-2.5)$ **EXE** 268.8

(puede omitirse)

Ejemplo: $12369 \times 7532 \times 74103 = 6.903680613 \times 10^{12} (=6903680613000)$

Operación: $12369 \times 7532 \times 74103$ **EXE** $6.903680613E 12$

Ejemplo: $1.23 \div 90 \div 45.6 = 2.997076023 \times 10^{-4} (=0.0002997076023)$

Operación: $1 \div 23 \div 90 \div 45.6$ **EXE** $2.997076023E -04$

* Cuando el resultado es mayor que 10^9 o menor que 10^{-3} , éste se visualizará en forma exponencial.

Ejemplo: $7 \times 8 + 4 \times 5 = 76$

Operación: $7 \times 8 + 4 \times 5$ **EXE** 76

Ejemplo: $12 + (2.4 \times 10^5) \div 42.6 - 78 \times 36.9 = 2767.602817$

Operación: $12 + 2.4 \times 10^5 \div 42.6 - 78 \times 36.9$ **EXE** 2767.602817

■ Cálculos por Memoria

Ejemplo: $12 \times 45 = 540$
 $12 \times 31 = 372$
 $75 \div 12 = 6.25$

Operación: $A \oplus 12$ **EXE**
 $A \times 45$ **EXE** 540
 $A \times 31$ **EXE** 372
 $75 \div A$ **EXE** 6.25

Ejemplo: $23 + 9 = 32$
 $53 - 6 = 47$
 $\rightarrow 45 \times 2 = 90$
 $99 \div 3 = 33$
Total 22

Operación: $M \oplus 23$ **EXE**
 $M \oplus 53$ **EXE**
 $M \ominus 45$ **EXE**
 $M \oplus 99$ **EXE**
 M **EXE** 22

* Al usarse este método de operación, no se podrán obtener resultados de cálculos individuales. Para poder visualizar resultados individuales, sígase el siguiente procedimiento.

$23 \oplus 9$ **EXE** 32
 $M \oplus$ **ANS** **EXE**
 $53 \ominus 6$ **EXE** 47
 $M \oplus$ **ANS** **EXE**
 45×2 **EXE** 90
 $M \oplus$ **ANS** **EXE**
 $99 \div 3$ **EXE** 33
 $M \oplus$ **ANS** **EXE**
 M **EXE** 22

■ Funciones Hiperbólicas ($\sinh$, $\cosh$ y $\tanh$) y Funciones Hiperbólicas Inversas ($\sinh^{-1}$, $\cosh^{-1}$ y $\tanh^{-1}$).

Ejemplo: $\sinh 3.6 = 18.28545536$

Operación: HSN 3 6 EXE 18.28545536
(o F1 HYP SIN 3 6; igual para los ejemplos que siguen).

Ejemplo: $\tanh 2.5 = 0.9866142981$

Operación: HTN 2 5 EXE 0.9866142981

Ejemplo: $\cosh 1.5 - \sinh 1.5 = 0.2231301602$

Operación: HCS 1 5 HSN 1 5 EXE 0.2231301602

Ejemplo: $\sinh^{-1} 30 = 4.094622224$

Operación: AHS 30 EXE 4.094622224
(o F1 HYP SIN 30; igual para los ejemplos que siguen).

Ejemplo: ¿Cuál es el valor de x si $4x = 0.88$? $x = \frac{\tanh^{-1} 0.88}{4} = 0.3439419141$

Operación: AHT 0 88 4 EXE 0.3439419141

■ Otras Funciones ($\sqrt{\quad}$, $x!$, SGN, RAN#, RND, ABS, INT y FRAC).

Ejemplo: $\sqrt{2} + \sqrt{5} = 3.65028154$

Operación: SQR 2 SQR 5 EXE 3.65028154

Ejemplo: $8! (=1 \times 2 \times 3 \times \dots \times 7 \times 8) = 40320$

Operación: 8 F1 EXE 40320

Ejemplo: Asignar un "1" a un número positivo, un "-1" a un número negativo y un "0" a un cero.

Operación: SGN 6 EXE 1
SGN 0 EXE 0
SGN 2 EXE -1

Ejemplo: Generación de Números Aleatorios (número pseudoaleatorio de $0 < \text{RAN\#} < 1$).

Operación: RAN # EXE 0.904186914

Ejemplo: Redondeo a 10^{-2} del resultado de $12,3 \times 4,56$. $12.3 \times 4.56 = 56.088$

Operación: RND 12 3 4 56 2 EXE 56.1

Ejemplo: $|-78.9 \div 5.6| = 14.08928571$

Operación: ABS 78 9 5 6 EXE 14.08928571

Ejemplo: La parte entera de $\frac{7800}{96}$ es 81.

Operación: INT 7800 96 EXE 81

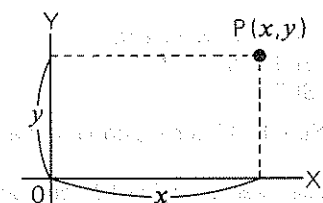
* Este comando no excederá el valor del número original.

Ejemplo: La parte decimal de $\frac{7800}{96}$ es 0,25.

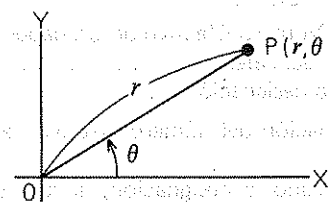
Operación: FRAC 7800 96 EXE 0.25

■ Conversión de Coordenadas (C $\rightarrow$ P y P $\rightarrow$ C).

• Coordenadas cartesianas



• Coordenadas polares



El θ de C $\rightarrow$ P puede obtenerse en la gama de $-180^\circ < \theta \leq 180^\circ$. ("RAD" y "GRAD" están dentro de la misma gama).

Ejemplo: ¿Cuánto valen r y θ cuando $x = 14$ y $y = 20,7$?

Operación: MODE 4 (designación de DEG).

RPC 14 20 7 EXE

(r) X EXE

(theta) DMS Y EXE

24.98979792

55°55'42.20"

Ejemplo: ¿Cuánto valen r y θ cuando $x = 7,5$ y $y = -10$?

Operación: MODE 5 (designación de RAD)

RPC 7 5 10 EXE

(r) X EXE

(theta) Y EXE

12.5

-0.927295218

Ejemplo: ¿Cuánto valen x e y cuando $r = 25$ y $\theta = 56$?

Operación: MODE 4

PRC 25 56 EXE

(x) X EXE

(y) Y EXE

13.97982259

20.72593931

Ejemplo: ¿Cuánto valen x e y cuando $r = 4,5$ y $\theta = 2/3\pi$ rad?

Operación: MODE 5

PRC 4 5 2 3 F1 EXE

(x) X EXE

(y) Y EXE

-2.25

3.897114317

■ Designación del Número Efectivo de Unidades y Designación de Decimales

La designación del número efectivo de unidades y la designación de decimales se llevan a cabo haciendo uso del comando "SET".

Designación del Número Efectivo de Unidades SET E n ($n=0$ a 9)

Designación de Decimales SET F n

Cancelación de la designación SET N

* Para la designación del número efectivo de unidades, "SET E 0" corresponde a una designación de 10 unidades.

* Si se lleva a cabo la designación, la función de desborde (Ver la página 13) provocará la anulación y redondeo en la pantalla de una unidad por debajo de la unidad designada. El valor del número original permanecerá intacto en la sección interna donde se está realizando el cálculo o en la memoria.

Ejemplo: $100 \div 6 = 16.66666666$

Operación: SET [E] 4 [EXE] (en este caso, se designan 4 unidades como unidades efectivas).

100 [2] 6 [EXE] 1.667E 01

Ejemplo: $123 \div 7 = 17.57142857$

Operación: SET [F] 2 [EXE] (2 unidades designadas para decimales).

123 [2] 7 [EXE] 17.57

Ejemplo: $1 \div 3 = 0.3333333333$

Operación: SET [N] [EXE] (cancelación de la designación).

1 [2] 3 [EXE] 0.3333333333

3-3-3 Cálculos Estadísticos

- En caso de llevarse a cabo cálculos estadísticos, asegurarse de presionar [F1] [SAC] con el fin de borrar la memoria para cálculos estadísticos; comenzar los cálculos después de esta operación.

■ Cálculo de la Desviación Estándar

- Use la tecla [STAT] y presionesela una vez por cada dato.

Dato [STAT]

- Cuando un mismo dato se repite, presionar la tecla [STAT] cuantas veces se repita dicho dato o entrar de la siguiente manera: "dato [2] cantidad [STAT]".

- Desviación Estándar

$$\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{\sum x^2 - (\sum x)^2 / n}{n-1}}$$

Usar datos de muestra en un grupo y designar la desviación estándar de dicho grupo.

- Promedio

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{\sum x}{n}$$

Ejemplo: Cuando los datos son: 55, 54, 51, 55, 53, 53, 54 y 52

Operación: [F1] [SAC]

55 [STAT] 54 [STAT] 51 [STAT] 55 [STAT]

53 [STAT] 54 [STAT] 52 [STAT]

(Desviación estándar σ_n) SDXN [EXE]

(Desviación estándar σ_{n-1}) SDX [EXE]

(Promedio $\bar{x}$) MX [EXE]

(Datos n) CNT [EXE]

(Sumatoria de x $\sum x$) SX [EXE]

(Sumatoria de x al cuadrado $\sum x^2$) SX2 [EXE]

STAT 52
1.316956719
1.407885953
53.375
8
427
22805

Ejemplo: ¿Cuánto valen σ_n y σ_{n-1} que aparecen en la siguiente figura?

Número de Grado	Valor del Grado	Grados
1	110	10
2	130	31
3	150	24
4	170	2
5	190	3

Operación:

[F1] [SAC]
110 [2] 10 [STAT]
130 [2] 31 [STAT]
150 [2] 24 [STAT]
170 [STAT] [STAT]
190 [STAT] [STAT] [STAT]

[F1] [ASTAT] [ANS]

SDXN [EXE]

SDX [EXE]

STAT 190
CNT= 70
SX=9640
SY=0
SX2=1351000
SY2=0
SXY=0
18.2968716
18.42898069

* Anulación y Corrección de Datos Equivocados (Operación de corrección: 51 [STAT]).

(1) 50 [STAT] y [F1] [DEL] ; operar luego en la forma correcta.

(2) 49 [STAT] (varios ítems antes) → 49 [F1] [DEL] ; operar luego en la forma correcta.

Las operaciones de anulación y corrección se pueden realizar de modo similar para cierto número de ítems.

49 [2] 12 [STAT] (varios ítems antes) → 49 [2] 12 [F1] [DEL] , operar luego en la forma correcta.

■ Cálculos de Regresión

- Presionar dato x [2] dato y [STAT].

- Cuando se dispone de un número de pares de datos similares, presionar la tecla [STAT] la cantidad de veces deseada o entrar de la siguiente manera: dato x [2] dato y [2] cantidad [STAT].

■ Cálculos de Regresión Lineal

- La ecuación de regresión es $y = A + Bx$

Los coeficientes A y B se calculan usando las siguientes fórmulas:

Coeficiente de regresión de la ecuación de regresión Término constante de la ecuación de regresión

$$B = \frac{n \cdot \sum xy - \sum x \cdot \sum y}{n \cdot \sum x^2 - (\sum x)^2}$$

$$A = \frac{\sum y - B \cdot \sum x}{n}$$

- El coeficiente de correlación r para los pares de datos de entrada se calcula usando la siguiente fórmula:

$$r = \frac{n \cdot \sum xy - \sum x \cdot \sum y}{\sqrt{\{n \cdot \sum x^2 - (\sum x)^2\} \{n \cdot \sum y^2 - (\sum y)^2\}}}$$

Ejemplo: • Temperatura y longitud de una barra de acero

Temperatura	Longitud
10°C	1003mm
15	1005
20	1010
25	1008
30	1014

Obtener la ecuación de regresión y el coeficiente de regresión de estos valores medidos. Calcular la longitud a una temperatura de 18°C y la temperatura para una longitud de 1000 mm.

Operación:

F1 **SAC**

10 **▢** 1003 **STAT**

15 **▢** 1005 **STAT**

20 **▢** 1010 **STAT**

25 **▢** 1008 **STAT**

30 **▢** 1014 **STAT**

(Término constante A) **LRA** **EXE**

(Coeficiente de regresión B) **LRB** **EXE**

(Coeficiente de correlación r) **COR** **EXE**

(Longitud a 18°C) **EOY** 18 **EXE**

(Temperatura para 1000mm) **EOX** 1000 **EXE**

STAT 10, 1003
STAT 15, 1005
STAT 20, 1010
STAT 25, 1008
STAT 30, 1014
998
0.5
0.9190182776
1007
4

* Para cancelar o corregir datos de entrada equivocados (operación de corrección: 10 **▢** 1003 **STAT**)

(1) 11 **▢** 1003 **STAT** → y **F1** **DEL** **STAT**, operar luego correctamente.

(2) 10 **▢** 1030 **STAT** → y **F1** **DEL** **STAT**, operar luego correctamente.

(3) 11 **▢** 1003 **STAT** (varios ítems antes) → 11 **▢** 1003 **F1** **DEL** **STAT**, operar luego correctamente.

La cantidad puede también anularse o corregirse de modo similar.

11 **▢** 1003 **▢** 10 **STAT** (varios ítems antes) → 11 **▢** 1003 **▢** 10 **F1** **DEL** **STAT**, operar luego correctamente.

■ Cálculos de Regresión Logarítmica

- La ecuación de regresión es $y = A + B \cdot \ln x$. El dato x se entra como el logaritmo ($\ln$) de x . El dato y se entra del mismo modo que en la regresión lineal.
- Para obtener el coeficiente de regresión o para corregir, úsen las mismas operaciones que en la regresión lineal; sin embargo, se requerirá $\sum \ln x$ para $\sum x$, $\sum (\ln x)^2$ para $\sum x^2$ y $\sum \ln x y$ para $\sum xy$.

Ejemplo:

x_i	y_i
29	1.6
50	23.5
74	38.0
103	46.4
118	48.9

Llevar a cabo la regresión logarítmica de estos datos y obtener el término constante, el coeficiente y coeficiente de correlación de la ecuación de la regresión y luego calcular el coeficiente determinante (r^2).

F1 **SAC**

Operación:

LN 29 **▢** 1 **▢** 6 **STAT**

LN 50 **▢** 23 **▢** 5 **STAT**

LN 74 **▢** 38 **▢** 0 **STAT**

LN 103 **▢** 46 **▢** 4 **STAT**

LN 118 **▢** 48 **▢** 9 **STAT**

(Término constante A de la ecuación de regresión) **LRA** **EXE**

(Coeficiente de regresión B) **LRB** **EXE**

(Coeficiente de correlación r) **COR** **EXE**

(Coeficiente determinante r^2) **2** **EXE**

TAT 3.36729583, 1.6
T 3.912023005, 23.5
TAT 4.304065093, 38
T 4.634728988, 46.4
T 4.770684624, 48.9
-111.1283976
34.02014749
0.9940139464
0.9880637256

■ Cálculos de Regresión Exponencial

- La ecuación de regresión es $y = A \cdot e^{B \cdot x}$ ($\ln y = \ln A + B \cdot x$). El dato y se entra como el logaritmo ($\ln$) de y . El dato x se entra del mismo modo que en la regresión lineal.
- El método de corrección es el mismo empleado en la regresión lineal, con la diferencia que se requieren $\ln A$ para el Término Constante A, $\sum \ln y$ para la Sumatoria de y (SUM Y), $\sum (\ln y)^2$ para la Sumatoria de y^2 y (SUM Y2) $\sum x \cdot \ln y$ para la Sumatoria de xy (SUM XY).

Ejemplo:

x_i	y_i
6.9	21.4
12.9	15.7
19.8	12.1
26.7	8.5
35.1	5.2

Llevar a cabo la regresión exponencial de estos datos y obtener la ecuación de regresión y el coeficiente de correlación.

Operación:

F1 **SAC**

6 **▢** 9 **▢** LN 21 **▢** 4 **STAT**

12 **▢** 9 **▢** LN 15 **▢** 7 **STAT**

19 **▢** 8 **▢** LN 12 **▢** 1 **STAT**

26 **▢** 7 **▢** LN 8 **▢** 5 **STAT**

35 **▢** 1 **▢** LN 5 **▢** 2 **STAT**

(Término constante A) **EXP LRA** **EXE**

(Coeficiente B) **LRB** **EXE**

(Coeficiente de correlación r) **COR** **EXE**

AT 6.9, 3.063390922
T 12.9, 2.753660712
T 19.8, 2.493205453
T 26.7, 2.140066163
T 35.1, 1.648658626
30.49758742
-0.04920370831
-0.9972473519

■ Cálculos de Regresión de Potencia

- La ecuación de regresión es $y = A \cdot x^B$ ($\ln y = \ln A + B \ln x$). Ambos datos x e y se entran en forma de logaritmos ($\ln$).
- El método de corrección es el mismo usado en la regresión lineal, con la diferencia que se requieren $\ln A$ para el Término Constante A, $\sum \ln x$ para $\sum x$, $\sum (\ln x)^2$ para $\sum x^2$, $\sum \ln y$ para $\sum y$, $\sum (\ln y)^2$ para $\sum y^2$ y $\sum (\ln x \cdot \ln y)$ para $\sum xy$.

Ejemplo:

x_i	y_i
28	2410
30	3033
33	3895
35	4491
38	5717

Llevar a cabo la regresión de potencia de estos datos y obtener la ecuación de regresión y el coeficiente de correlación.

Operación:

F1 **SAC**

LN 28 **▢** LN 2410 **STAT**

LN 30 **▢** LN 3033 **STAT**

LN 33 **▢** LN 3895 **STAT**

LN 35 **▢** LN 4491 **STAT**

LN 38 **▢** LN 5717 **STAT**

(Término constante A) **EXP LRA** **EXE**

(Coeficiente B) **LRB** **EXE**

(Coeficiente de correlación r) **COR** **EXE**

3220451, 7.787382026
1197382, 8.017307508
6507561, 8.267448958
5348061, 8.409830673
3758616, 8.651199471
0.2388010829
2.771866148
0.9989062562

Capítulo 4

Cálculos por Programa

Esta calculadora utiliza en sus programas el lenguaje BASIC. BASIC es una sigla de origen inglés cuyo equivalente en el idioma español es "Código de Instrucciones Simbólico Multipropósito para Principiantes". Este lenguaje se lo considera como un sistema de lenguaje fundamental de simple uso para novatos en el campo de las computadoras.

Características del Lenguaje BASIC

1. Es un lenguaje para la solución de problemas que mejora la eficacia de la programación.
2. Su uso no se limita a un campo en particular; es aplicable en muchos temas, como ser ciencias naturales, ciencias sociales y en el campo comercial.
3. Es un lenguaje del tipo conversacional, con el cual el usuario de la computadora puede componer programas mientras "conversa" con su máquina y hacer uso del mismo en las entradas.
4. Posee muchas de las características y facilidades del lenguaje FORTRAN (lenguaje de traducción de fórmulas) pero no tiene, en cambio, las reglas y limitaciones de este.
5. Se han preparado numerosas funciones fundamentales incorporadas y posee, por otro lado, una amplia gama de cálculos.

4-1 Generalidades de un Programa

Cálculos por programa **1)** Contenido del cálculo por programa a ejecutarse **2)** La calculadora almacena los programas y **3)** se hacen uso de estos. Entrar los datos y se podrán obtener automáticamente los resultados.

■ Principios para Programación

Hechemos un vistazo a las operaciones de programación necesarias para usar la computadora en un problema dado y a los conceptos y procedimientos de estas operaciones.

● Programas y Programación

Cuando el operador hace uso de la computadora para solucionar un problema, este deberá usar instrucciones, las cuales son palabras que la computadora misma puede comprender. A estas instrucciones se las denominan programas, mientras que se denomina programación a la composición de estas instrucciones.

● ¿Qué es un programa?

Existen numerosas reglas de sintaxis que se deben seguir cuando se compone un programa, de las cuales hablaremos más adelante. Antes de disertar sobre lo que es un programa, veamos un ejemplo de un programa fundamental.

	Comando	Operando	
10	INP	A, B	Instrucción general de entrada
20	C =	A + B	Instrucción general de operación
30	PRT	C	Instrucción general de salida
			Número de línea

Este es un programa fundamental y está compuesto por una instrucción general de entrada, una de operación, una de salida y los números de líneas. Digamos, entonces, que la instrucción general de entrada da entrada a los datos. De acuerdo a estos datos, la instrucción general operativa lleva a cabo varias operaciones, mientras que la instrucción general de salida da salida a los resultados de las operaciones ejecutadas. Además, cada línea tiene un número que le corresponde antes de esta. Las instrucciones generales operativas no se limitan a sólo una; por el contrario, se pueden llevar a cabo varias veces por medio de la inclusión de instrucciones generales de decisión, mediante las cuales el programa se haría largo y complicado. No obstante, el principio de un programa largo y complicado como el que se acaba de mencionar es el mismo.

Se puede observar también que cada línea tiene un número, a continuación del cual se encuentra un COMANDO; este comando le indica a la calculadora cuál es el siguiente paso a seguir. Este último está compuesto por caracteres alfabéticos y viene seguido por el OPERANDO, el cual da la información necesaria para el comando que le antecede.

Esto que se acaba de describir es, entonces, un PROGRAMA en su formato convencional.

● Número de pasos de programa

Los pasos de programa se cuentan del siguiente modo:

- 1) Comando de programa 1 paso/comando
- 2) Comando funcional 1 paso/comando
- 3) Número de línea 2 pasos/número de línea
- 4) Carácter 1 paso/carácter
- 5) Pulsación de la tecla **EXE** después de la entrada de cada línea para almacenarla en la calculadora 1 paso

Ejemplo:

1	INP	A	EXE		5 pasos
2	B	=	SIN	A	EXE
10	PRT	B	EXE		7 pasos
2	B	=	"	B	EXE
10	PRT	"	B	EXE	10 pasos
2	B	=	"	B	EXE
10	PRT	"	B	EXE	Total 22 pasos

Nota: Para la asignación de una contraseña se requieren 6 pasos (remitirse a la página 74).

■ Secuencia de Programación

A continuación se muestra la secuencia en el trabajo de programación:

- 1) Análisis del problema
- 2) Preparación del diagrama de flujo
- 3) Inscripción del programa en un formulario de codificación.
- 4) Ejecución de la depuración

Explicación

Paso 1: Consiste en el análisis del problema con el fin de determinar el número de pasos requeridos para la solución del mismo.

Paso 2: Composición de un diagrama de flujo para representar el progreso lógico de la solución del problema. Este diagrama se compone de símbolos, los cuales muestran el proceso y los elementos de toma de decisiones.

Paso 3: Consiste en la inscripción del programa en un formulario de codificación o similar usando el lenguaje BASIC y basándose en el diagrama de flujo preparado.

Paso 4: Verificación de posibles errores en el programa.

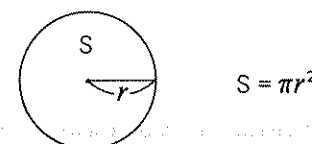
Estos cuatro pasos conforman el procedimiento para la realización de un programa.

■ Diagrama de Flujo

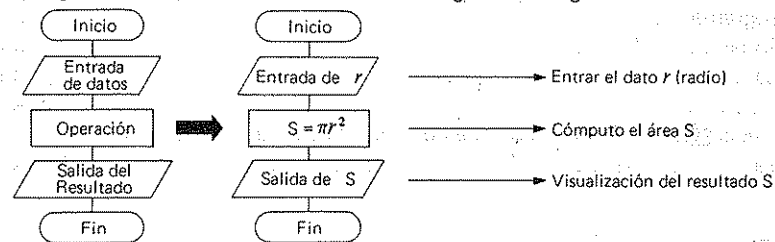
Símbolo	Nomenclatura	Significado
	Terminal	Inicio o fin de un proceso.
	Entrada/Salida	Para funciones de entrada/salida.
	Proceso	Para varias funciones de proceso.
	Proceso pre-establecido	Grupo de comandos definidos en otra parte (ej.: una subrutina).
	Decisión	Para la decisión de una ruta específica a elegirse entre varias alternativas.

● Ejemplo de un Diagrama de Flujo

Observemos un simple ejemplo de un programa para determinar el área de un círculo.



Se consideran, antes que nada, la entrada de datos, la operación y la salida del resultado tal como se vió en los principios de programación y como se muestra en el siguiente diagrama.



Es de esta manera como se prepara el diagrama de flujo, siguiendo el progreso del programa y formándolo teniendo en consideración al programa en forma global.

La preparación de este tipo de diagrama de flujo deberá hacerse hábito en toda programación. Gracias a estos diagramas se pueden evitar confusiones cuando se crean programas más complejos.

■ Codificación

La codificación consiste en el proceso de inscripción del programa en formularios de codificación o similares después del ensamblaje del mismo.

Cuando se ensamblan programas, se requiere el uso de ciertos símbolos fundamentales más simples.

Símbolos de las cuatro operaciones aritméticas fundamentales:

Los signos "+" y "-" se expresan del mismo modo.

Sin embargo, los signos $\times$ y $\div$ se expresan "X" y "/", respectivamente.

Las potencias, como ser x^2 y x^3 se expresan como "x↑2" y "x↑3", respectivamente.

El signo "=", usado en instrucciones generales de asignación, será explicado en detalle más adelante; daremos aquí sólo una breve descripción del mismo. Por ejemplo, el signo "=" en la expresión $S = \pi r^2$ significa que se le asigna a S el resultado de πr^2 ; en consecuencia, no tiene, como en matemáticas, un significado de igualdad.

Escribamos, entonces, un programa para determinar el área del círculo previamente especificado.

1) Escribir una instrucción general de entrada para dar entrada al dato r.

Son varios los comandos de entrada que existen; sin embargo, el más usado es el comando "INP", el cual se utiliza para entrar datos desde el teclado durante la ejecución de un programa. La instrucción general de entrada usada para entrar el dato r sería, entonces, INP R. Si a esta instrucción general se le agrega un número de línea, obtendríamos 10 INP R.

Estos números de línea pueden elejirse entre 1 y 9999; sin embargo, no se pueden usar números iguales en un mismo programa. De lo contrario, tendría prioridad el posterior. Se considera conveniente, generalmente, asignar estos números de línea de diez en diez, con el fin de facilitar posteriores inclusiones, correcciones o anulaciones en el programa. Cuando se inicia el programa, las instrucciones se ejecutan en una secuencia de menor a mayor (correspondiente al número de línea). Por lo tanto, asígnense los números de línea de acuerdo a la secuencia de ejecución del programa.

2) Dése a continuación una instrucción general de asignación para asignar a S el resultado del cálculo correspondiente a los datos entrados.

Puesto que πr^2 sería igual a $\pi \times r^2$, obtenemos $S = \pi \times R \uparrow 2$ (el signo X se respresenta por medio de X y R^2 mediante R↑2).

Si se le agrega el número de línea, tendríamos:

20 S = πX R↑2

3) Escribir una instrucción general de salida para dar salida (visualizar) el resultado de la operación.

En vista que este procedimiento es sólo efectivo para el cálculo en sí y no hace que se visualice en pantalla el resultado, para llevar a este último a la pantalla deberá usarse el comando "PRT".

La instrucción general de salida para dar salida (visualizar) el resultado S sería

PRT S

Agregándole el número de línea correspondiente, obtendríamos:

30 PRT S

En consecuencia, el programa completo para obtener el área del círculo es:

10 INP R

20 S = πX R↑2

30 PRT S

Para la codificación del programa no será necesario utilizar un formulario de codificación. Sin embargo, será más conveniente su uso si se tiene en cuenta que esto facilita las operaciones de análisis de problemas o de dibujo de diagramas de flujo posteriores.

4-2 Principios de un Programa

■ Constantes y Variables

Los caracteres usados en el lenguaje BASIC son letras mayúsculas del alfabeto (de la A a la Z), números (de 0 a 9) y algunos caracteres especiales (símbolos, por ejemplo).

● Constantes

Una constante es un valor fijo que puede escribirse en un programa directamente.

Ejemplo: En la expresión $S = \pi r^2$, al pasarla al BASIC obtenemos: $S = \pi \times R \uparrow 2$, de cuya expresión 2 sería la constante.

● Variables

Una variable es un valor usado en el programa que, a diferencia de las constantes, se entra desde el teclado durante la ejecución y se usa cuando inicialmente se desconoce el resultado de un cálculo ya que este se asigna durante la ejecución.

Una variable se puede respresentar por letras mayúsculas (de la A a la Z) o combinando una letra mayúscula con un número de una sola cifra (A0, A1 hasta A9 y de B0 hasta T9 cuando se expande la memoria). La asignación puede realizarse dentro de estos límites.

Ejemplo: En la expresión $S = \pi r^2$, que en BASIC se representa $S = \pi \times R \uparrow 2$, R es la variable.

Ejemplo: $Y = 2 \times X \uparrow 2 + 3 \times X + 4$

V
C
V
C
C
V
C

V: Variable
C: Constante

Como se puede observar, entonces, los símbolos algebraicos y las constantes que se usan en las matemáticas corresponden a las variables y constantes (respectivamente) que se acaban de mencionar. Tenemos, por otro lado, constantes de caracteres y variables de caracteres. Las constantes de caracteres son secuencias ordenadas de caracteres que se escriben directamente, como ser "ABC" y "END"; éstas se encuentran siempre entre comillas y se deletrean. Las variables de caracteres no corresponden a valores numéricos, sino que son variables que aceptan una secuencia ordenada de caracteres. Por lo tanto, su contenido cambia cada vez que se le asigna una secuencia ordenada de caracteres.

Una secuencia ordenada de caracteres se compone de caracteres entre comillas tales como "123" y no tienen valor numérico. En consecuencia, se considera a "123" como una secuencia de los números 1, 2 y 3 del mismo modo que, por ejemplo, "ABC".

Las variables de caracteres son variables generales (por ej.: A, B, X e Y) con el signo \$ adjunto. Las combinaciones posibles son las que permiten estas limitaciones.

Ejemplo: A\$, B\$, C\$, X\$, Y\$

Las variables de caracteres pueden sumarse o compararse entre sí; no sucede lo mismo, sin embargo, con las demás operaciones (resta, multiplicación y división).

Ejemplo: Si A\$ = "123" y B\$ = "456",
Usando C\$ = A\$ + B\$
Obtenemos que C\$ es igual a "123456"
(Si C\$ = B\$ + A\$, el resultado sería "456123")

En esta variable de caracteres se puede insertar una secuencia ordenada de hasta 7 caracteres.

Tenemos, además de las variables de caracteres, las variables de caracteres exclusivas. Estas se visualizan usando el signo \$. En la secuencia ordenada de caracteres se pueden insertar hasta 30 caracteres.

Ejemplo: \$ = "1234567890ABCDEF G"

Estas variables de caracteres exclusivas pueden usar funciones de caracteres (funciones MID), las cuales se explicarán más adelante. A estas variables se les puede dar un uso más conveniente que las demás variables de caracteres.

■ Instrucciones Generales de Asignación

Las instrucciones de asignación del BASIC tienen el siguiente formato:

Variable = Expresión Numérica

En las instrucciones generales del BASIC, toda expresión con un símbolo de operaciones aritméticas que se encuentra a la derecha se la considera o denomina una expresión numérica (+, -, *, /).

Ejemplo: Para $Y = 2 \times X + 3$, la expresión $2 \times X + 3$, que se encuentra a la derecha, es una expresión numérica. El signo "=" no indica igualdad, sino que significa "asignación".

Ejemplo: Para $Y = 2 \times X + 3$, la parte de la izquierda es la variable y la de la derecha es una expresión numérica. Esta expresión en forma global no significa, como significaría en la matemática convencional, que Y es igual a $2 \times X + 3$. Esta expresión indica que se asigna a Y el resultado de $2 \times X + 3$. (La expresión de arriba podrá entenderse mejor si se la tiene en mente de la siguiente manera: $Y \leftarrow 2 \times X + 3$).

Ejemplos de Instrucciones Generales de Asignación

A = B Asignar a A el valor de B (se anula el valor original de A).
N = 2 * M Asignar a N el doble del valor de M.
X = Y + Z Asignar a X el resultado de la suma de Y y Z.
I = I + 1 Asignar a I el valor de la expresión I + 1 (el valor numérico de I aumenta 1).

4-3 Escritura y Ejecución de Programas

■ Escritura de un Programa

Se denomina escritura de programa a la operación que consiste en almacenar un programa en la memoria de la calculadora.

Esta operación se lleva a cabo haciendo entradas desde el teclado.

1. Designación del modo WRT.
2. Designación del área de programa.
3. Entrada del programa por unidades de líneas (escritura).
4. El área de programa puede dividirse en diez partes, de P0 a P9. Los programas se escriben en cualquiera de estas secciones dentro del área de programa.

(1) Designación del Modo WRT

Todo programa se escribe con la calculadora en el modo WRT. Presiónese **MODE** **1** y "WRT" se visualizará en la pantalla.

(2) Designación del Área de Programa

Para designar un área de programa, presiónese la tecla **F1** y, al mismo tiempo, cualquier tecla numérica de **0** a **9**.

F1 P0 → P0	F1 P5 → P5
F1 P1 → P1	F1 P6 → P6
F1 P2 → P2	F1 P7 → P7
F1 P3 → P3	F1 P8 → P8
F1 P4 → P4	F1 P9 → P9

(3) Entrada del Programa (Escritura)

Todo programa se escribe línea por línea. Se pueden escribir hasta 62 caracteres, incluyendo el número de línea.

Presionar, por último, la tecla **EXE**.

* **Función de la Tecla EXE.**

Esta tecla se usa (presionándola) para escribir programas, entrar datos y obtener resultados de cálculos manuales. Para el caso cuando se escriben programas, la tecla **EXE** se presiona después de la entrada (desde el teclado) de cada número de línea, con el fin almacenar dicha línea en la calculadora. En toda operación de modificación de escritura de programas o de programas ya escritos (agregados y anulaciones), el paso final es la pulsación de esta tecla **EXE**. Si se ha cambiado el contenido de la pantalla y no se presiona la tecla **EXE**, el contenido de lo escrito no variará.

Ejemplo: Escribir el siguiente programa en P0.

```
10 INP A,B
20 V=A+B
30 W=A-B
40 PRT V,W
50 END
```

Operación:

1. Designación del Modo WRT.

Escrito en P6

MODE **1** **READY P6: 012345_7891228**

Area de programa actualmente designada Areas de programa no utilizadas Número de pasos

* El número de pasos cambiará según el número de memorias o la cantidad de programas escritos.

2. Designación del área de programa P0.

F1 **P0** **READY P0: 012345_7891228**

3. Si todavía queda algún programa previamente escrito, bórreselo.

CLR **EXE** **READY P0: 012345_7891228**

(Este paso se omite cuando no hay nada escrito).

4. Escribir la línea número 10.

10 **INP** **A,B** **EXE** **10 INP A,B 1221**

Correspondiente al espacio de un carácter (puede omitirse) * Asegurarse de presionar esta tecla cuando se cambian las líneas.

5. Escribir la línea número 20.

20 **V=A+B** **EXE** **20 V=A+B 1213**

6. Escribir la línea número 30.

30 **W=A-B** **EXE** **30 W=A-B 1205**

7. Escribir la línea número 40.

40 **PRT V,W** **EXE** **40 PRT V,W 1198**

8. Escribir la línea número 50.

50 **END** **EXE** **50 END 1194**

- Usar el comando "END" para dar por terminado el programa. El comando "END" podrá omitirse en aquellos programas como el de arriba; no obstante, deberá escribirse con el fin de clarificar la terminación de un programa cuando se utilicen en éste instrucciones generales GOTO o GSB.
- Para facilitar la lectura de lo visualizado en la pantalla, se abrirán espacios entre los números de línea, los comandos y los operandos. Estos espacios no tienen, en el lenguaje BASIC, ningún significado en especial, excepto en las instrucciones generales PRT.
- En este caso, los números de línea se entran en unidades de a 10. Estos números de línea se pueden elegir libremente entre 1 y 9999. No obstante, resultará más fácil realizar cualquier modificación posterior si estos números se entran de diez en diez. La ejecución del programa se lleva a cabo de acuerdo a los números de línea en una secuencia de menor a mayor; asígnense, entonces, los números de línea de acuerdo a la secuencia de ejecución del programa deseada.
- Usar el comando "CLR" tal como se muestra arriba con el fin de borrar el programa previamente escrito; para borrar todos los programas escritos previamente (de P0 a P9), úsese el comando "CLR ALL".

■ Ejecución del Programa

La ejecución del programa se lleva a cabo en el modo RUN (presiónense las teclas **MODE** **0** y "RUN" se visualizará en la pantalla).

- Existen dos métodos para la ejecución de un programa.

(1) Métodos de Ejecución de Programas

1. Ejecución por Designación del Área de Programa

En este método, la ejecución comienza al mismo tiempo que la designación del área de programa.

Operación: (Presionar después de **F1** cualquier tecla entre **P0** a **P9**).

Ejemplo: Comenzar el programa especificado en el ejemplo previo.

Operación: **F1** **P0** **?** **RUN** (omitido de aquí en más)

* Este signo "?" aparece por que la instrucción general INP está escrita al principio.

2. Ejecución por medio del comando RUN

RUN **EXE** (RUN puede entrarse presionando **RUN** o **F2** **RUN**).

* Continuando con el ejemplo previo, se visualiza un signo "?". En una situación o condición de espera de entrada, las cancelaciones por la tecla **AC** no tendrán efecto. Después de presionar **MODE** **0**, se lleva a cabo la operación 2.

En caso de desearse comenzar el programa desde una línea determinada, entrar el número de dicha línea y presiónense la tecla **EXE** después de la ejecución del comando RUN.

Ejemplo: Comienzo desde la línea 20.

Operación: **RUN** **20** **EXE**

* Para el método 1, no es necesario designar el área de programa a ejecutarse. Sin embargo, en el método 2, sí lo es (si el área de programa no es la correcta, se ejecutará el programa escrito en dicho área).

(2) Entradas por el Teclado Durante la Ejecución de un Programa

Para la entrada por el teclado durante la ejecución de un programa, se usan la instrucción general INP y la función KEY. En la entrada por la función KEY, esta se limita a una sola tecla; aún cuando no se hace ninguna entrada, la ejecución continúa.

Para las entradas por medio de teclas usando la instrucción general INP, se visualiza en la pantalla el signo "?" y se entra en una condición de pausa o de espera de entrada. Después de entrar los datos pertinentes, la ejecución se reanuda presionando la tecla **EXE**.

Ejemplo: Ejecutar el programa escrito en P0 del ejemplo previo.

Operación: Ejecución del programa **F1** **P0** **?**

• Para este programa se entran 2 variables. Entrar primero el valor de la variable A.

47 **EXE** **?**

• Se entra luego el valor de la variable B.

69 **EXE** **116**

CONT **-22**

De este modo, para la entrada por teclado durante la ejecución usando la instrucción general INP, éntrense los datos empleando la secuencia: Datos **EXE**. Por otro lado, en el caso de pulsarse la tecla **EXE** directamente (sin entrar ningún número) durante el período de espera de entrada de la instrucción general INP, aparecerá en pantalla la inscripción "STOP" y se entrará en un estado de detención. En este estado, se podrán realizar otras operaciones, como ser, manuales. Para reanudar el programa interrumpido, presiónense la tecla **CONT**. Además, en el estado de espera de entrada, la ejecución del programa se detendrá si se presionan **EXE** **AC** directamente.

4-4 Compaginación de Programas

- La compaginación de programas se emplea para hacer al programa lógicamente correcto, haciendo posible la realización de cambios, agregados y anulaciones del contenido del mismo, como así también el reordenamiento de los números de línea.
- La compaginación de programa se realiza usando el comando LIST, el cual es para llamar (recuperar) cada línea.
- El comando LIST es útil en ambos modos RUN y WRT. En el modo RUN, el comando LIST visualiza el contenido del programa. Si se lo emplea en el modo WRT, se podrá realizar la compaginación de programa.

(1) Visualización del listado de programa en el modo RUN.

(La visualización dura aproximadamente 2 segundos)

Operación: **LIST** **EXE**

(LIST puede ser **LIST** o **F2** **LIST**)

10	INP A,B
20	V=A+B
30	W=A-B
40	PRT V,W
50	END

Si no se necesita el listado desde un principio, se designa directamente el número de línea. Para comenzar desde la línea número 30:

Operación: **LIST** **30** **EXE**

30	W=A-B
40	PRT V,W
50	END

* Durante la ejecución del comando LIST, la visualización se hará en forma secuencial hasta el final. Para detener, presiónense la tecla **STOP**. Para reanudar un comando LIST detenido, presiónense la tecla **CONT**.

(2) Cambios, agregados y anulaciones en un programa en el modo WRT.

1. Cambios

Usándose el comando LIST, cada línea se visualizará secuencialmente a partir del número de línea asignado, cada vez que se presiona la tecla **EXE**. Si se omite la designación del número de línea, la visualización comenzará desde la primera.

a. Modificaciones Parciales

Ejemplo: Cambio de "X" por "+" en la línea 20 del ejemplo previo.

Operación:

- Designar el área de programa P0 en caso de estar designada un área diferente.

F1 **P0** **READY P0: _12345_7891194**

- Recuperar la línea número 20 usando el comando LIST.

LIST **20** **EXE** **20 V=A+B_ 1194**

- Llevar el cursor hasta la posición (por debajo de ella) donde se desea realizar la modificación (en este caso, por debajo de "+").

20 V=A±B 1194

* Podrá desplazarse el cursor rápidamente si se presiona la tecla de movimiento del mismo (← o →) por más de 1 segundo.

- Realizar la corrección.

*** EXE** **30 W=A-B_ 1194**

* Asegurarse de presionar la tecla **EXE**. Si no se la presiona, sólo cambiará el contenido de la pantalla, sin modificarse el contenido del programa escrito.

- En este instante, el número de línea cambia a 30. Presionar la tecla **AC** para borrar la pantalla. De este modo se concluye con la modificación.

AC - 1194

* Si se opera alguna tecla o teclas en alguna línea donde no se necesita ninguna modificación, se cambiará el contenido de la misma. Por lo tanto, téngase cuidado de no presionar ninguna otra tecla además de **EXE** **AC**.

- Usar el comando LIST para confirmar la modificación.

MODE **0**
LIST **EXE**

READY P0
10 INP A,B
20 V=A*B
30 W=A-B
40 PRT V,W
50 END

b. Cambio de Toda una Línea

Entrar el número de la línea a cambiarse. (De este modo, se borra el número de línea entrado previamente).

Ejemplo: Cambiar "W=V/2" por "W=A-B" en la línea 30.

Operación:

MODE **1**

READY P0: _12345_789¹¹⁹⁴

- Escribir la línea 30 nueva.

30 **W** **=** **V** **/** **2** **EXE**

30 W=V/2 1194

- Verificar el listado del programa.

MODE **0**
LIST **EXE**

READY P0
10 INP A,B
20 V=A*B
30 W=V/2
40 PRT V,W
50 END

2. Agregados

Cuando se desean agregar líneas, usar algún número de línea que se encuentre entre los números de línea donde se quiere realizar la inserción.

Ejemplo: Agregar "U=V*2" entre las líneas 30 y 40 del ejemplo previo y cambiar la línea 40 a "PRT V,W,U".

Operación:

MODE **1**

READY P0: _12345_789¹¹⁹⁴

- Entre el número de línea 35 para insertar entre las líneas 30 y 40.

35 **U** **=** **V** ***** **2** **EXE**

35 U=V*2 1186

* Para la entrada entre las líneas 30 y 40, elíjase un número de 31 hasta 39.

- Para modificar o cambiar la línea 40, llévase dicha línea a la pantalla usando la instrucción general LIST y agréguese "U".

LIST **40** **EXE**

U **EXE**

AC

40 PRT V,W 1186
50 END 1184
- 1184

- Verificar el listado para confirmar los cambios.

MODE **0**
LIST **EXE**

READY P0
10 INP A,B
20 V=A*B
30 W=V/2
35 U=V*2
40 PRT V,W,U
50 END

3. Anulaciones

a. Anulación Parcial

Ejemplo: Anular "V" de la línea 40 del ejemplo previo.

Operación:

MODE **1**

READY P0: _12345_789¹¹⁸⁴

- Del mismo modo que en el método de modificaciones parciales, recuperar la línea 40 usando la instrucción general LIST.

LIST **40** **EXE**

40 PRT V,W,U 1184

- Mover el cursor y colocarlo por debajo de la letra W (a la derecha de la posición a anularse).

← **←** **←**

40 PRT V, W, U 1184

- Usar la tecla **C** para anular "V,"

C **C**

40 PRT W,U 1184

EXE

50 END 1186

* Si no se presiona la tecla **EXE**, no se cambiará el contenido del programa en sí.

AC

- 1186

* Asegúrese de presionar la tecla **AC** con el fin de cancelar la situación de cambio de la línea número 50.

- Verificar el listado para asegurarse de que la anulación se hizo correctamente.

MODE **0**
LIST **EXE**

READY P0
10 INP A,B
20 V=A*B
30 W=V/2
35 U=V*2
40 PRT W,U
50 END

b. Anulación de toda una línea.

Entrar el número correspondiente a la línea a anularse con lo cual se la anulará en forma completa.

Ejemplo: Anular la línea 30.

Operación:

MODE **1**

READY P0: _12345_789¹¹⁸⁶

- Entrar el número de línea a anularse, en este caso: 30.

30 **EXE**

- 1194

- Confirmar la posición anulada.

MODE **0**
LIST **EXE**

READY P0
10 INP A,B
20 V=A*B
35 U=V*2
40 PRT W,U
50 END

4. Corrección del Número de Línea

Ejemplo: El siguiente programa se encuentra escrito en P2.

```
10 INP N
20 M=N+2
30 L=N+0.5
40 PRT M,L
50 END
```

Desplazar la línea número 20 a una posición entre las líneas 30 y 40.

Operación: **READY P2: _1_345_789¹¹⁶⁰**

- Recuperar la línea número 20 usando el comando LIST.

20 M=N+2_ ¹¹⁶⁰

- Llevar el cursor hasta la posición por debajo del 2 de la línea 20.

20 M=N+2 ¹¹⁶⁰

- Cambiar de 20 a 35 y dar entrada.

35 30 L=N+0.5 ¹¹⁵²

- Para dar por concluido el cambio, presiónese la tecla y cáncelse el comando de cambio.

_ ¹¹⁵²

- Usar el comando LIST para confirmar el cambio en el contenido del programa.

READY P2

10 INP N
20 M=N+2
30 L=N+0.5
35 M=N+2
40 PRT M,L
50 END

- En este estado, ya se ha llevado el contenido de la línea 20 a una posición entre las líneas 30 y 40; sin embargo, la línea 20 todavía permanece. Bórrase, entonces, esta línea innecesaria.

READY P2: _1_345_789¹¹⁵²

20 _ ¹¹⁶⁰

- Con esta operación se completa el desplazamiento del número de línea. Verificar el resultado usando el comando LIST.

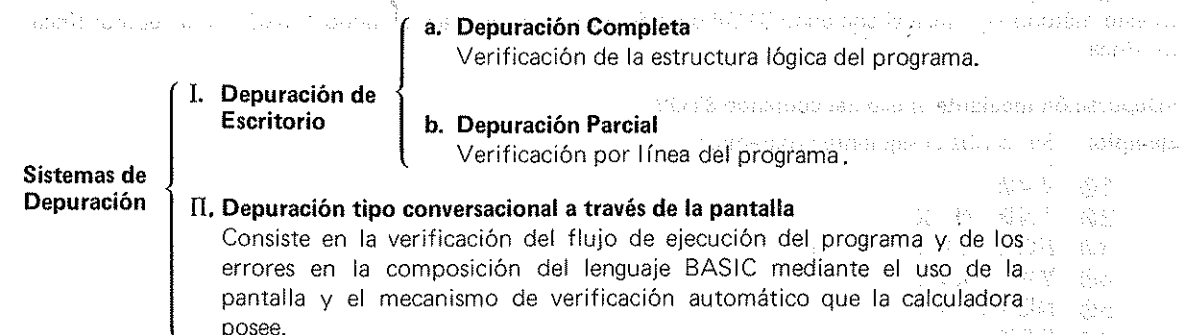
READY P2

10 INP N
30 L=N+0.5
35 M=N+2
40 PRT M,L
50 END

■ Depuración de Programas

(1) Sistema de Depuración de Programas

El sistema de depuración de programas de este equipo se divide en dos tipos: a. depuración de escritorio y b. depuración tipo conversacional a través de la pantalla.



La depuración de escritorio se lleva a cabo durante la programación. Hechemos un vistazo a la depuración de tipo conversacional a través de la pantalla.

(2) Depuración de Tipo Conversacional

Todo error cometido durante las etapas de ejecución del programa aparecerán en la pantalla usando un mensaje de error. Estos errores se visualizan en una unidad por línea y dan a conocer el tipo de error del lenguaje BASIC. Basándose en estos mensajes de error, se podrá llevar a cabo la depuración usando este manual.

En cuanto a el significado de los mensajes de error, remitirse a la lista de mensajes de error que se encuentra en la Página 79.

Ejemplo:

```
10 INP X
20 Y=X+2+3*X+15
30 PRT Y
40 END
```

En este programa, $Y = X + 2 + 3X + 15$ se entró en la línea 20 por error.

Operación: Si se ejecuta el programa, se visualizará el signo "?" usando la instrucción general 10 INP.

RUN ?

- Entrar entonces 45,

45 ERR-2 IN P0-20

- Este mensaje de error indica que ha ocurrido un error en la estructura de la instrucción general de la línea 20 y que deberá confirmarse el contenido del programa.

READY P0: _123456789

20 Y=X+2+3X+15_

- Puesto que falta el signo "X" entre el "3" y la "X" de la línea 20, se lleva a cabo la corrección usándose el método de compaginación de programas ya visto.

20 Y=X+2+3_X+15

30 PRT Y

(3) Depuración del Programa Durante la Ejecución del Mismo

La depuración de tipo conversacional es aquella que se realiza usando la información obtenida de los mensajes de error que la calculadora visualiza. Sin embargo, hay casos en que no aparece ningún mensaje de error pero el resultado del cálculo no es el esperado. Bajo estas circunstancias, se deberá repetir la ejecución del programa y llevar a cabo la depuración confirmando los cálculos parciales. En este método se utiliza el comando STOP para detener el programa y el modo TRACE para depurar línea por línea.

● Depuración mediante el uso del comando STOP

Ejemplo: Se escribe el siguiente programa.

```
10 Y=0
20 INP N,X
30 FOR I=1 TO N
40 Y=Y+X↑2
50 NEXT I
60 PRT Y
70 END
```

Usar la instrucción general STOP para ver el resultado de cada lazo con el fin de observar el valor de Y en el lazo FOR-NEXT.

Operación: La mejor posición para entrar la instrucción general STOP es a continuación de la fórmula del cálculo; escribase, por lo tanto, la instrucción general STOP entre las líneas 40 y 50.

MODE 1 READY P0: 123456789
45 STOP

● De este modo, el progreso del programa se detiene una vez terminado el cálculo de la línea 40, permitiéndose así la depuración.

MODE 0 RUN
4 EXE
87 EXE
?
?
-
El cursor parpadea

● ¿Cuál es el valor de Y en el momento de esta detención?

Y EXE 7569

● Cuando se reanuda el programa, este se volverá a detener en la próxima instrucción general STOP, donde se volverá a dar el valor de Y.

CONT
Y EXE
-
15138

● Repitiendo esta operación se puede, entonces, observar el proceso del cálculo.

Este ejemplo emplea un programa simple, pero cuando en la realidad se compone un programa complejo, resulta muy dificultoso verificar el proceso de los cálculos por medio de la depuración de escritorio. Si, en cambio, se verifican las variables usando este tipo de instrucción general STOP, podrán detectarse y corregirse errores.

● Depuración usando el modo TRACE

Si se realiza la ejecución del programa usando el modo TRACE (presionando MODE 2), la ejecución se hará línea por línea, permitiéndose la realización de la depuración por detención y durante la ejecución. Llevemos a cabo un ejemplo de depuración en el modo TRACE empleando el comando STOP ya visto.

Operación:

Los comandos "RUN", "STOP" y "TRACE" se omiten de aquí en más.

Designación del modo "RUN" MODE 0
Designación del modo TRACE MODE 2
RUN EXE
Verificación del contenido del programa STOP
CONT
4 EXE
87 EXE
STOP
CONT
STOP
CONT
STOP
Valor de Y Y EXE
CONT
STOP
CONT
STOP
Repetir estos pasos.

READY P0	RUN	STOP	TRACE
P0-10	Y=0		
P0-20			
P0-20	INP N, X		
P0-30	FOR I=1 TO N		
P0-40	Y=Y+X↑2		
7569			
P0-45	STOP		
P0-50	NEXT I		

Este tipo de depuración usando el modo TRACE es el más adecuado método para la examinación del flujo total del programa y, en consecuencia, es muy conveniente para encontrar donde hay errores.

4-5 Comandos de Programa

4-5-1 Saltos y Lazos

■ Comandos de Salto

Los comandos de salto se pueden dividir en dos. Uno de ellos, llamado instrucción general GOTO, que es un salto incondicional. El otro es un salto condicional combinado con una instrucción general IF.

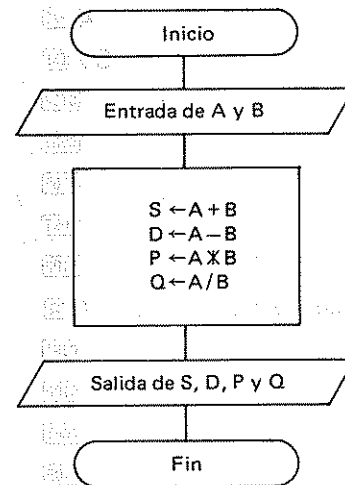
■ Instrucción general GOTO

Se considera a la instrucción general GOTO como un salto incondicional, porque es un comando que avanza el programa en forma incondicional hasta una posición designada (un número de línea).

Ejemplo 1: Agregar una instrucción general GOTO en un programa utilizado para determinar la suma, resta, producto y cociente entre los datos A y B.

Este programa es tal como se ilustra a continuación.

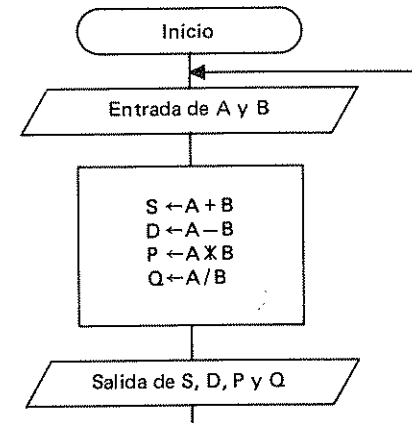
```
10 INP A,B
20 S=A+B
30 D=A-B
40 P=A*B
50 Q=A/B
60 PRT S,D,P,Q
70 END
```



Cada vez que se ejecuta, deberá ponerse en el modo "RUN" con el fin de permitir la entrada de datos. No obstante, si se entra una instrucción GOTO en la línea 70 en lugar de END, el control volverá a la posición apropiada donde se entran los datos (línea 10 en este programa).

Esta instrucción GOTO (GOTO 10) realiza un salto incondicional a la línea 10. Se obtendría el programa que se ilustra a continuación.

```
10 INP A,B
20 S=A+B
30 D=A-B
40 P=A*B
50 Q=A/B
60 PRT S,D,P,Q
70 GOTO 10
```



En este programa, una vez entrados los datos, se entra en un estado de espera de entrada (se visualiza el signo "?") y los próximos datos pueden entrarse inmediatamente.

Ejecución: Entrar los datos 15 y 3, y 903 y 43.

Operación:

RUN [EXE]

15 [EXE]

3 [EXE]

[CONT]

[CONT]

[CONT]

[CONT]

903 [EXE]

43 [EXE]

[CONT]

[CONT]

[CONT]

?
?
18
12
45
5
?
?
946
860
38829
21

En este programa, en el cual se usa una instrucción general GOTO incluida al final, el control vuelve a la línea número 10 "INP A, B". Sin embargo, si en este ejemplo se escribe un número de línea después de GOTO, este saltará a la línea del número designado. Por otro lado, si en lugar de un número de línea se escribe "#" y un número entre "0 a 9", este saltará al área de programa designada por dicho número.

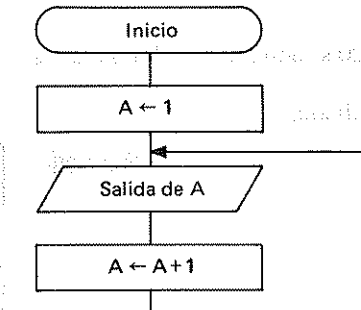
Ejemplo:

GOTO 10 (Saltar a la línea 10 y ejecutar desde la misma línea)

GOTO #5 (Saltar al área de programa P5 y ejecutar el programa que allí se encuentra)

Ejemplo 2: Hacer un programa para incrementar de a 1 el valor de A.

```
10 A=1
20 PRT A
30 A=A+1
40 GOTO 20
```



Explicación:

Puesto que el valor de A aumenta de a 1 en forma secuencial, es necesario asignar a A un valor inicial, en este caso 1. O sea, "A=1", que aparece en la línea número 10.

A continuación, PRT (imprimir) A.

En la línea 30, se asigna a A el resultado de la suma de A más 1, o sea, "A=A+1". Entonces, debido a que el proceso del programa provoca un salto por la instrucción general GOTO, es necesario saltar a la línea 20 en lugar de volver al comienzo. Sería, en este caso, GOTO 20.

De este modo, la instrucción general GOTO hace que el proceso del programa salte en forma incondicional a una posición designada.

Nota: Siempre que se utilice la instrucción general GOTO, deberá uno asegurarse de designar la posición correcta (el número de línea correcto). Aparecerá un error si se omite en la instrucción general GOTO el número de línea.

Aplicación de la Instrucción General PRT

La instrucción general PRT usada en este programa cambia el modo de visualización de acuerdo con las designaciones delimitadoras que se encuentran a continuación de los operandos.

En el ejemplo 1, S, D, P y Q están separados por comas (",").

La visualización se detiene después de S. Para visualizar D, presiónese la tecla [CONT]. Por lo tanto, cuando la separación es una coma, los resultados se visualizarán uno por vez. ¿Cómo será la visualización en caso que las separaciones se hagan por medio de ";" y no ",",?

A continuación se muestra el resultado.

Operación:

RUN EXE
15 EXE
3 EXE
CONT
903 EXE
43 EXE

?
?
18 12 45 5
?
?
946 860 38829 21

El signo ";" (punto y coma), en este caso, opera cuando hay varios resultados y hace que la totalidad se visualice secuencialmente.

Por otro lado, en la pantalla, aparece un espacio después de cada resultado. El signo "+" no se visualiza en la columna de signos, pero en realidad está presente.

18 12 45 5
columna de signos

Si el ejemplo 2 se ejecuta como está, se obtendrá un resultado como el que se muestra a continuación.

Operación:

RUN EXE
CONT
CONT
CONT

1
2
3
4

Sin embargo, si en la línea 20 se entra una ";" después de "PRT A",
20 PRT A;
se obtendrá el siguiente resultado.

RUN EXE

1	2	3	4	5	6	7	8	9	1
1	2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10	
2	3	4	5	6	7	8	9	10	1
3	4	5	6	7	8	9	10	11	

En consecuencia, resultados continuos se visualizan usando el signo ";". Existe, también, un comando "WAIT" (de espera) para detener la visualización por cierto período de tiempo con el fin de detener la visualización de la instrucción general PRT. Para más detalles, remitirse a la Página 59.

• Designación Indirecta de la Instrucción GOTO

En el ejemplo previo de la instrucción GOTO, la designación se hacía directamente y se provocaba el salto. En la designación indirecta, el destino del salto se determina según el valor de una variable. Se utiliza cuando no se puede escribir el destino del salto al principio del programa, como cuando el método de operación se interpreta según la información.

La designación indirecta está compuesta por:

GOTO variable o expresión numérica
GOTO # variable o expresión numérica

Este utiliza el valor de la variable (A, B, X, Y, etc.) o una expresión numérica (A+B, X+10, etc.) para determinar a que número de línea o área de programa se salta.

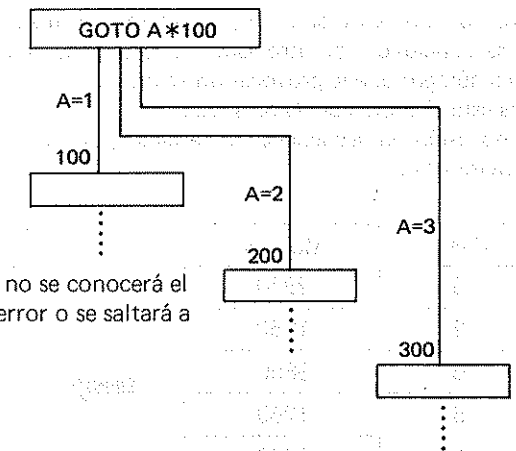
Ejemplo 1: GOTO A*100

En este programa cuando A es 1, 2 ó 3:

A=1, GOTO 100 indica un salto a la línea 100

A=2, GOTO 200 indica un salto a la línea 200

A=3, GOTO 300 indica un salto a la línea 300

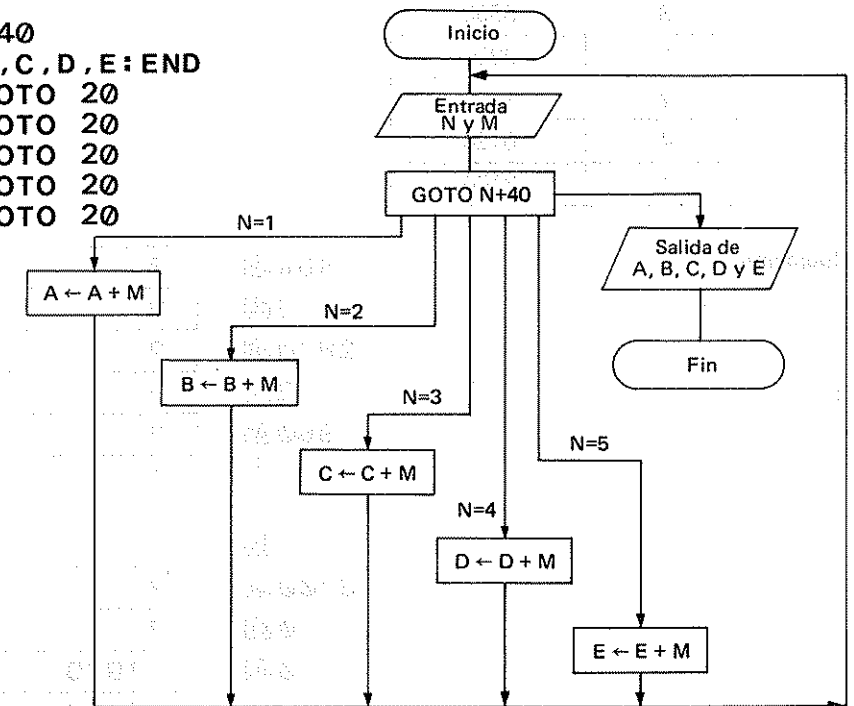


Cuando A tenga otro valor que no sea 1, 2 ó 3, no se conocerá el destino del salto. En este caso se presentará un error o se saltará a otro número de línea.

Ejemplo 2: Hacer un programa de totales clasificados usando la designación indirecta (clasificado en 5 divisiones).

El programa será tal como se muestra a continuación:

```
10 VAC
20 INP N,M
30 GOTO N+40
40 PRT A,B,C,D,E:END
41 A=A+M:GOTO 20
42 B=B+M:GOTO 20
43 C=C+M:GOTO 20
44 D=D+M:GOTO 20
45 E=E+M:GOTO 20
```



El comando "VAC" de la línea 10 es para borrar la memoria para uso de datos (la lleva a 0). En este programa se requiere un borrado previo con el fin de totalizar la entrada de datos en las líneas 41 a 45. En la instrucción general INP que se encuentra a continuación se entran el número clave (N) y el monto de las entradas.

Usando la instrucción GOTO de la línea 30, si el número clave (N) es 1, se salta a la línea 41, o sea, las entradas totales del número clave 1.

En el caso del número clave 2, se salta a la línea 42. De este modo se dividen las entradas (M) por números claves del 1 al 5.

N=1 : GOTO 41 → 41
N=2 : GOTO 42 → 42
N=3 : GOTO 43 → 43
N=4 : GOTO 44 → 44
N=5 : GOTO 45 → 45

A=A+M : GOTO 20 Ejecutar
B=B+M : GOTO 20 Ejecutar
C=C+M : GOTO 20 Ejecutar
D=D+M : GOTO 20 Ejecutar
E=E+M : GOTO 20 Ejecutar

Por lo tanto, cuando N es igual a 0, se saltará a la línea 40. Los resultados clasificados A, B, C, D y E se visualizan y se concluye. Por otro lado, en este programa, N deberá entrarse con un valor entre 0 y 5. Cualquier otro número puede provocar un error.

Demostremos este ejemplo con valores reales:

Los recibos no están en secuencia; éntreselos, por lo tanto, en secuencia y clasifíquense los totales (en menos de 5 divisiones).

Clave	Montos
3	2870
2	1960
5	3850
5	1250
1	2500
2	2310
3	1850
5	4370
3	5360
1	2220
2	1450
4	6120
1	3100



Clave	Montos
1	7820
2	5720
3	10080
4	6120
5	9470

Operación:

RUN [EXE]

3 [EXE]

2870 [EXE]

2 [EXE]

1960 [EXE]

...

1 [EXE]

3100 [EXE]

0 [EXE]

0 [EXE]

[CONT]

[CONT]

[CONT]

[CONT]

[CONT]

?
?
?
?
?

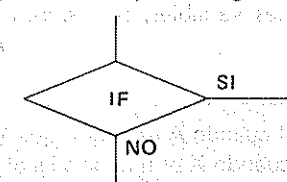
?
?
?
7820
5720
10080
6120
9470

* El signo ":" usado en las líneas 40 a 45 de este programa se denominan "multi-instrucción". Esta permite escribir en una línea diferentes comandos. Si se escriben comandos de esta forma, se pueden omitir números de línea y ahorrarse memoria. Gracias al uso de esta multi-instrucción, se pueden combinar muchos comandos. No deberá olvidarse, sin embargo, que el número de caracteres que se pueden escribir en una línea se limita a 62, incluyendo el número de línea.

● Instrucción general IF

La instrucción general IF se la considera un salto condicional por su naturaleza. Esta ejecuta su operación sólo cuando se satisfacen ciertas condiciones y es un comando para saltar a una posición designada.

Su representación gráfica en los diagramas de flujo es la siguiente:



Significa que si se satisface la instrucción IF, se avanza por la ruta "SI"; de lo contrario, se va por el camino "NO". En otras palabras, la instrucción IF es una etapa donde se debe tomar una decisión para determinar la siguiente operación según un resultado.

Esta instrucción general IF se utiliza para dar por terminado un lazo cuando se desconoce el número de datos o cuando se modifica la siguiente operación de acuerdo al resultado de una operación, etc.

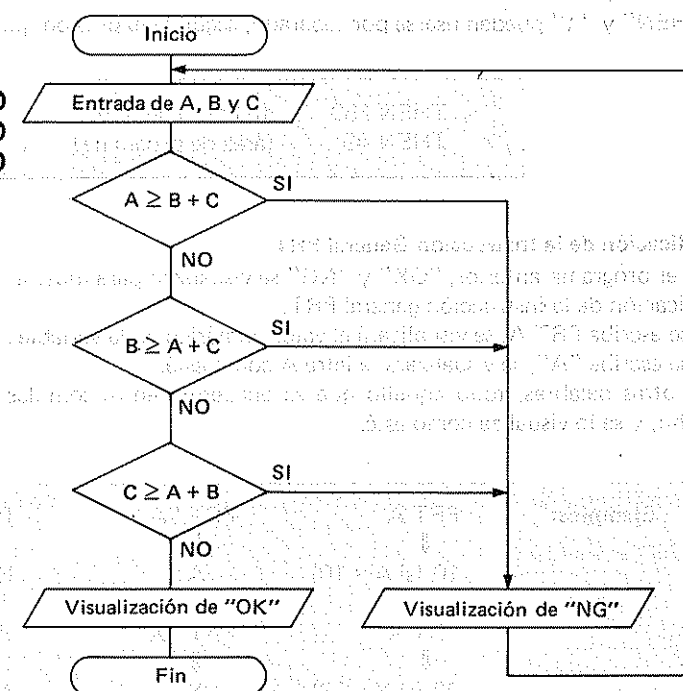
Ejemplo 1: Entrar el largo de tres lados de un triángulo y determinar si se puede formar dicho supuesto triángulo con los tres lados entrados.

El programa es el siguiente:

```

10 INP A,B,C
20 IF A≥B+C THEN 70
30 IF B≥A+C THEN 70
40 IF C≥A+B THEN 70
50 PRT "OK"
60 END
70 PRT "NG"
80 GOTO 10

```



En este programa se entran tres datos. Debido a la naturaleza de todo triángulo, la suma del largo de dos lados es mayor que el lado restante. Se compara, entonces, cada lado y, si no se puede formar un triángulo, "NG" se visualiza en la pantalla y el programa vuelve a la operación de entrada de datos. Si se puede formar un triángulo, "OK" se visualiza en la pantalla y se da por terminado el programa.

La instrucción general IF está compuesta por expresiones como las que se muestran a continuación:

IF expresión de comparación THEN número de línea (expresión numérica) o #n (n=0 a 9)

o

IF expresión de comparación; comando o instrucción de asignación

La expresión comparativa que sigue a "IF" compara el lado derecho de un signo de igualdad o desigualdad con el lado izquierdo del mismo. Cuando el resultado es SI, procederá al paso posterior a "THEN" o ";" ;". De lo contrario (en caso de ser NO), pasa a la línea siguiente. En otras palabras, en la línea 20, cuando A es mayor o igual que la suma de B y C, no se puede formar un triángulo, por lo cual se lleva a cabo el comando "THEN 70", o sea, salto a la línea 70. Este comando "THEN" incluye la función de la instrucción general GOTO.

Si después de "THEN" se escribe un número de línea, se saltará a la línea correspondiente a dicho número. Si se escriben " #" y un número de 0 a 9, se saltará a un área de programa (P0 a P9). Por otro lado, cuando el resultado de la expresión comparativa es SI, y se desea efectuar un comando o una instrucción de asignación en lugar de un salto, úsese ";" en lugar de "THEN".

En estas comparaciones se pueden usar constantes, variables, expresiones numéricas, constantes de caracteres y variables de caracteres.

$A > 10$	variable y constante (SI cuando A es mayor que 10)
$X \geq Y$	variable y variable (SI cuando X es mayor o igual que Y)
$N = L + 3$	variable y expresión numérica (YES cuando N es igual a la suma de L más 3)
$A\$ = "XYZ"$	variable de carácter y constante de carácter (SI cuando la secuencia ordenada de caracteres en A\$ es igual a "XYZ")
$P\$ \neq Q\$$	variable de carácter y variable de carácter (SI cuando la secuencia ordenada de caracteres en P\$ no es igual a la secuencia ordenada de caracteres en Q\$)
* No se pueden hacer comparaciones de variables con variables de caracteres.	
* La comparación de secuencias ordenadas de caracteres se aplica a los códigos ASCII.	

"THEN" y ";" pueden usarse por separado, según la expresión que le siga.

THEN 150	(número de línea)	; PRT A
THEN #9	(área de programa)	; Z = X + Y

Aplicación de la Instrucción General PRT

En el programa anterior, "OK" y "NG" se visualizan para mostrar el resultado de la comparación. Esta es la aplicación de la instrucción general PRT.

Si se escribe PRT A, se visualizará el valor numérico en la variable A.

Si se escribe "A", se visualizará la letra A como está.

En otras palabras, todo aquello que se encuentra entre comillas se lo considera un carácter propiamente dicho, y se lo visualiza como esté.

Ejemplos:	PRT A	PRT "A"	PRT "RESPONDER"
	↓	↓	↓
	10 (si A = 10)	A	RESPONDER
	PRT X	PRT "X"	PRT "N=" ; N
	↓	↓	↓
	23 (si X = 23)	X	N = 15 (si N = 15)

Ejemplo 2: Obtener los valores mínimo y máximo tal como se muestra a continuación.

```

10 INP A
20 B=A
30 C=A
40 I=1
50 INP A
60 IF A=0 THEN 110
70 IF A>B;B=A
80 IF A<C;C=A
90 I=I+1
100 GOTO 50
110 PRT I;B;C
120 END

```

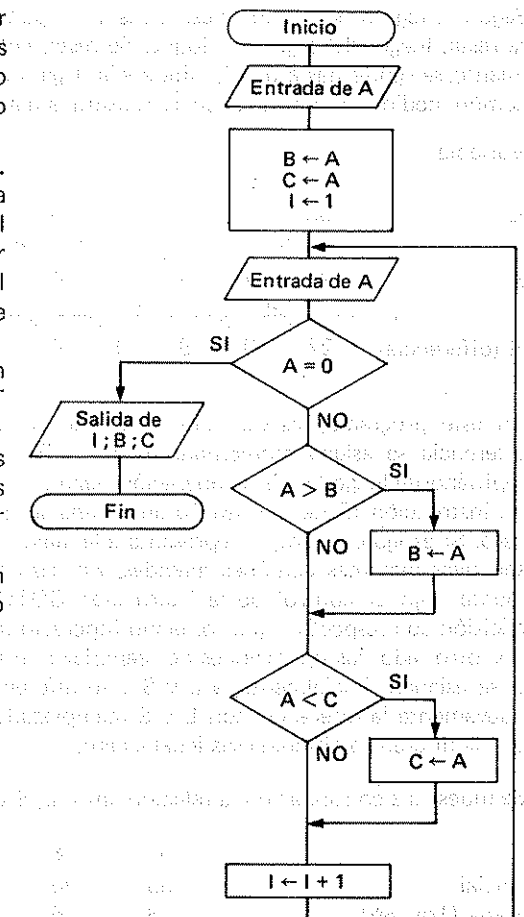
La instrucción general INP se usa en la línea 10 para entrar los datos iniciales. Estos datos iniciales corresponden a los valores mínimo y máximo. Se procesan, por lo tanto, junto con los datos de las líneas 20 y 30, asignándose el dato inicial con el valor mínimo B y el valor máximo C.

La línea 40 usa la variable I para contar el número de datos. En consecuencia, se entra un "1" como dato inicial. La instrucción de entrada de la línea 50 se usa para entrar el segundo y demás datos. Este procedimiento se repite por medio de la instrucción GOTO de la línea 100. Si se entra el dato "0" usando la instrucción general IF de la línea 60, se hará que el programa termine.

En otras palabras, si se entra "0" utilizando la instrucción INP de la línea 50, el control saltará a la instrucción PRT para dar salida al resultado usando la línea 110.

Las instrucciones IF de las líneas 70 y 80 juzgan si los datos de entrada son mayores o no que el valor máximo de los datos ya entrados o si son más pequeños que el valor mínimo, y llevan a cabo el reemplazo.

Una vez concluida la entrada de todos los datos, se visualizan el número de datos y los valores máximo y mínimo usando la línea 110.



• Instrucciones Generales GOTO y IF Misceláneas

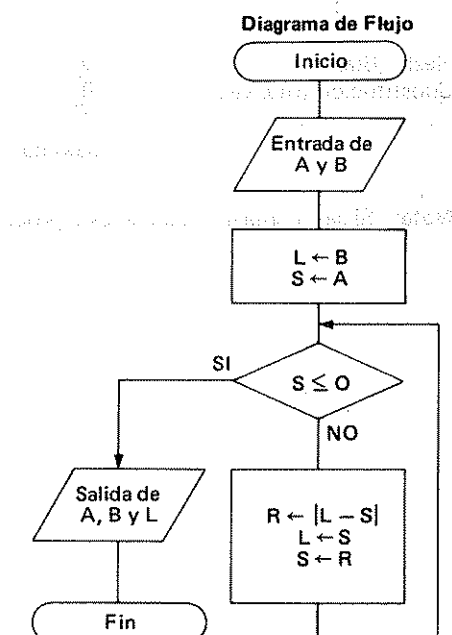
Hecharemos aquí un vistazo a algunos ejemplos prácticos de las instrucciones GOTO y IF.

Ejemplo 1: Programa para la determinación de la máxima común medida según el antilogaritmo de Euclides.

```

10 INP A,B
20 L=A
30 S=B
40 IF S≤0 THEN 90
50 R=ABS(L-S)
60 L=S
70 S=R
80 GOTO 40
90 PRT A;B;L
100 END

```



Según el logaritmo de Euclides, se resta el segundo bloque de datos del primero. El resultado (la diferencia) se resta, luego, del segundo bloque. Se resta, entonces esta diferencia del resultado anterior. Este procedimiento se repite hasta que la diferencia llega a cero. Cuando esto ocurre, la diferencia previa es la máxima común medida. A continuación se muestra la máxima común medida de 63 y 36.

Variable	L	S	R (diferencia)
	63	36	27
	36	27	9
	27	9	18
	9	18	9
	18	9	9
	9	9	0

← máxima común medida

← cuando llega a 0

En este programa, las variables para obtener las diferencias se designan a L y S, respectivamente. Esta diferencia se asigna momentáneamente a R; luego se reemplazan L y S por S y R, respectivamente, repitiéndose luego la misma operación (lazo).

La instrucción IF de la línea 40 determina si S, que es la diferencia después del reemplazo, es igual o no a cero. Si es igual a cero, L representa a la máxima común medida, visualizándose los datos A y B con sus respectivas máximas comunes medidas. En caso de no ser igual a cero, se determina la diferencia repetidamente bajo el control de la instrucción GOTO de la línea 80. La instrucción IF se usa para tomar la decisión con respecto a que comando (operación) debe realizarse, basándose en el resultado del cómputo. Por otro lado, las operaciones de reemplazo en las líneas 50 y 70 merecen especial atención. Los datos A y B se asignan, inicialmente, a L y S y su diferencia se asigna momentáneamente a R. Luego, se determina nuevamente la diferencia con L y S reemplazados por S y R, respectivamente. Este proceso se repite hasta que la diferencia obtenida sea igual a cero.

Se muestra a continuación la relación entre L, S y R.

	L	S	R
Inicial	63	36	0
Resta (1ra. vez)	63	36	27
Substitución (1ra. vez)	36	27	27
Resta (2da. vez)	36	27	9
Substitución (2da. vez)	27	9	9
Resta (3ra. vez)	27	9	18
Substitución (3ra. vez)	9	18	18
...	...	...	...
Resta (6ta. vez)	9	9	0
Substitución (6ta. vez)	9	0	0

↑ máxima común medida

R = ABS (L-S)
L = S, S=R
R = ABS (L-S)
L = S, S=R
R = ABS (L-S)
L = S, S=R
R = ABS (L-S)
L = S, S=R

Nota: Si se cambia el orden en este proceso, se perderá el sentido de las operaciones de substitución.

Ejemplo 2: Hacer un programa para determinar el mínimo común múltiplo.

```

10 INP A, B
20 I=1
30 M=A*I
40 IF M=INT(M/B)*B THEN 70
50 I=I+1
60 GOTO 30
70 PRT A, B, M
80 END

```

En este ejemplo, se entran primero los datos A y B; luego, se multiplica el valor inicial de A por uno, dos, tres y así sucesivamente, hasta que el resultado de esta multiplicación se iguale con el múltiplo más pequeño del valor de B, tal como se establece en la instrucción general IF. Esta instrucción IF de la línea 40 compara la variable "M" con la expresión numérica "INT (M/B)*B". En otras palabras, se compara la variable "M" con el resultado de dicha expresión numérica. Este método sería igual que el siguiente:

```

N=INT(M/B)*B
IF M=N THEN 70

```

Sin embargo, con este método se ahorran líneas y la cantidad de memoria utilizada.

El multiplicador I, que es aumentado de a uno por medio de la instrucción GOTO de la línea 60, se multiplica y juzga nuevamente.

Las variables aquí usadas son A, B, M e I. A y B son los dos datos. Puesto que M es un múltiplo de A, será más fácil saber el valor de M más adelante.

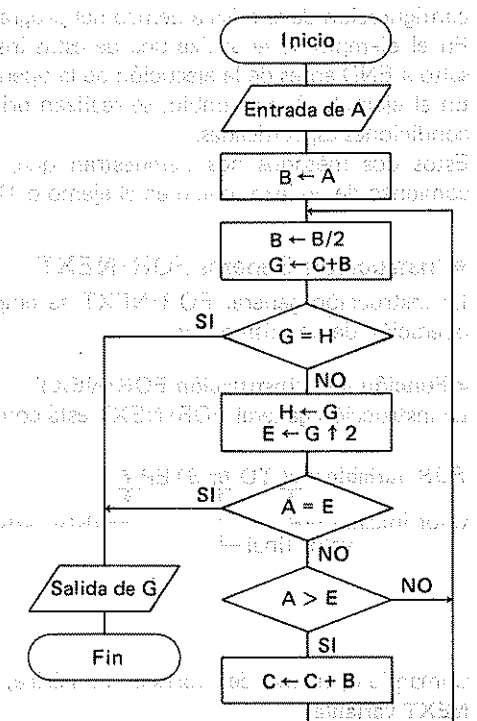
El multiplicador I se va incrementando de a uno con el fin de generar un nuevo múltiplo. Muy frecuentemente se usa como una variable que se incrementa con cada realización del lazo.

Ejemplo 3: Hacer un programa para determinar raíces cuadradas por dicotomía. El programa es el siguiente.

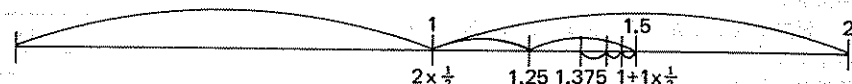
```

10 VAC
20 INP A
30 B=A
40 B=B/2
50 G=C+B
60 IF G=H THEN 120
70 H=G
80 E=G+2
90 IF A=E THEN 120
100 IF A>E; C=C+B
110 GOTO 40
120 PRT G
130 END

```



Según la dicotomía, primero se determina la mitad del valor del dato A y se comparan el cuadrado de esta mitad con el valor de A para ver si son iguales o no. En caso de ser negativo el resultado, se determina la mitad de la mitad previa y se la eleva al cuadrado, repitiéndose esta operación hasta que se obtiene un valor aproximado al de A. El proceso de cómputo se muestra a continuación.



En este programa, se incluye en la línea 10 un comando "VAC" con el fin de borrar la memoria. Luego se entra el dato A.

Este dato se usa posteriormente en las operaciones comparativas de las instrucciones generales IF. No se pueden hacer cambios directos. Por lo tanto, todo cambio se realiza por medio de la asignación de B (para cambiar B).

La instrucción general IF se utiliza para determinar el valor aproximado en la línea 60.

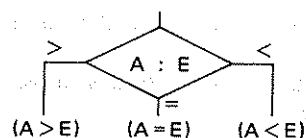
Se determina el límite superior de los dígitos de cómputo. Luego, se usan G y H para determinar el punto en el cual se alcanza el límite superior y, cuando el resultado actual es igual al previo, se da por concluida la operación.

La instrucción IF que se encuentra en la línea 90 es igual que la que se observa en el ejemplo 2. Si se puede dividir la raíz cuadrada, el cuadrado del resultado aquí determinado y el dato A serán iguales, visualizándose de este modo la solución.

La instrucción general IF de la línea 100 realiza pruebas con el fin de determinar en cuál de las dos partes divididas por igual (del resultado) se encuentra la solución (el valor aproximado).

El uso que se le da aquí a las instrucciones IF es diferente a lo previamente explicado; la función de cada una de estas instrucciones sigue, sin embargo, siendo la misma.

El programa en sí es algo complicado; es por que se requiere esta combinación de instrucciones generales IF. Las instrucciones IF de las líneas 90 y 100 comparan A y E entre sí. La representación gráfica de esta operación sería como se muestra a continuación:



El BASIC, sin embargo, no brinda la posibilidad de tres alternativas en las instrucciones generales IF. Se emplean, en su lugar, dos instrucciones IF combinadas.

Varias son las funciones que puede cumplir una instrucción IF, lo cual depende de la ubicación y configuración de la misma dentro del programa, hecho que se puede observar en los ejemplos 1 y 2.

En el ejemplo 1, se utiliza una de estas instrucciones al comienzo del lazo para que pueda provocarse un salto a END antes de la ejecución de la operación.

En el ejemplo 2, en cambio, se realizan primero las operaciones y luego se llevan a cabo las pruebas de las condiciones especificadas.

Estos dos métodos nos demuestran que, según los datos, una instrucción general IF puede entrarse al comienzo de un lazo (como en el ejemplo 1) o al final de una operación (como en el ejemplo 2).

■ Instrucción General FOR·NEXT

La instrucción general FOR·NEXT se emplea cuando se sabe el número de veces que se va a realizar la operación de un mismo lazo.

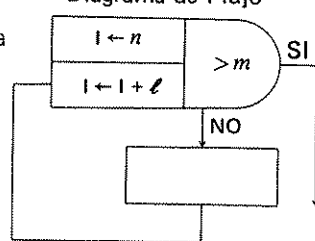
● Función de la Instrucción FOR·NEXT

La instrucción general FOR·NEXT está compuesta por:

FOR variable = n TO m STEP l

valor inicial → n valor final → m incremento ⇒ variable o expresión numérica → l

Diagrama de Flujo



Comando operativo de expresión numérica, etc.

NEXT variable

Esta instrucción comanda la acción especificada entre "FOR" y "NEXT" para que se ejecute repetidamente al mismo tiempo que una variable va cambiando de n a m a incrementos determinados por l. El control avanzará a la línea siguiente a "NEXT" cuando la variable llega al valor de m.

Por ejemplo, el programa que se muestra a continuación nos hace ver el modo de ejecución de cierta acción mientras la variable I aumenta desde 1 hasta 10 a incrementos de 2.

El STEP puede omitirse cuando el incremento es igual a uno.

FOR I = 1 TO 10 es igual que

FOR I = 1 TO 10

* La instrucción FOR·NEXT sólo puede aumentar una variable a incrementos de un valor específico, pero no puede hacer que decrezca. Si se desea hacer decrecer una variable de 10 a 1, deberá escribirse entonces un número negativo después de STEP, por ejemplo, "STEP-1".

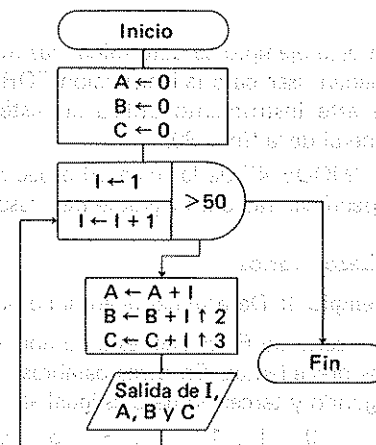
FOR I = 1 TO 10 STEP 2

NEXT I

Ejemplo 1: Hacer una tabla para $\sum_{i=1}^n i$, $\sum_{i=1}^n i^2$, $\sum_{i=1}^n i^3$, con n aumentando de 1 a 50.

Se muestra a continuación el programa apropiado.

```
10 VAC
20 FOR I=1 TO 50 STEP 1
30 A=A+I
40 B=B+I↑2
50 C=C+I↑3
60 PRT I, A, B, C
70 NEXT I
80 END
```



Este ejemplo se puede realizar solamente usando la instrucción general FOR·NEXT. Los totales de i , i^2 e i^3 ($\sum_{i=1}^n$) se visualizan mientras se va aumentando la variable n desde 1 hasta 50 a incrementos de uno.

Lo que significa que, la operación especificada entre "FOR..." de la línea 20 y "NEXT..." de la línea 70 se repite 50 veces, al mismo tiempo que I va aumentando de 1 a 50 a incrementos de uno. Este ejemplo puede llevarse a cabo usando una instrucción general IF.

```
10 VAC
20 I=1
30 A=A+I
40 B=B+I↑2
50 C=C+I↑3
60 PRT I, A, B, C
70 I=I+1
80 IF I=51;END
90 GOTO 30
```

De este modo, si se compara el ejemplo en el cual se usó la instrucción FOR·NEXT con el cual emplea la instrucción IF, se podrá interpretar y comprender claramente la función de la primera instrucción mencionada.

En otras palabras, la instrucción FOR·NEXT combina las funciones de prueba de las instrucciones IF y GOTO con una función de incremento.

```
20 FOR I=1 TO 50 STEP 1 ⇒ { 20 I=1
                             70 I=I+1
80 NEXT I ⇒ { 80 IF I=51;END
              90 GOTO 30
```

De este modo, se considera que esta instrucción (FOR·NEXT) es un comando muy conveniente y flexible cuando se sabe el número de veces que va a ocurrir un lazo, ya que posee ambas funciones de prueba e incremento.

4-5-2 Matrices

Las matrices se dividen en dos categorías: matrices unidimensionales (listas) y matrices bidimensionales (tablas). La elección para su uso depende de los elementos.

Las matrices unidimensionales se representan por $A(i)$ o $B(j)$, siendo (i) y (j) los subíndices. Las matrices bidimensionales se representan por medio de $x(i, j)$ o $y(n, m)$; en este caso, (i, j) y (n, m) son dos subíndices de dos variables.

A continuación se muestran dos ejemplos de los tipos de matrices previamente mencionados.

Matriz unidimensional

$i \backslash A$	
0	A0
1	A1
2	A2
3	A3
4	A4
5	A5
6	A6
7	A7
8	A8
9	A9
10	B0
...	

Matriz bidimensional

$i \backslash j$	0	1	2	3	4	5	6	7	8	9
0	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
1	B0	B1	B2	...	...	...	...	...	B8	B9
2	C0	C1	...	...	...	...	...	...	...	C9
3	D0	...	...	...	...	...	...	...	...	D9
4										
5										
6										
7										
8										
9										
10										
...										

Estas matrices utilizan las memorias A0 a A9 y hasta T9 cuando se amplía la memoria.

Las matrices unidimensionales se representan por medio de $A(0)$ hasta $A(199)$.

Las bidimensionales se representan por medio de $A(0,0)$ hasta $A(19,9)$.

En otras palabras, $A(0)$ y A0 corresponden a la misma memoria; lo mismo sucede con $A(10)$ y B0, $A(1,0)$ y B0 y $A(18,7)$ y S7.

Por otro lado, en las matrices solamente A se puede usar como variable.

* Cuando se usan matrices, es necesario ampliar el número de memorias según el tamaño de la primera. Esta operación se lleva a cabo usando el manual.

DEFM n (n se encuentra entre los límites 0 a 19 y deberá ser igual o mayor que 1 cuando se usa en una matriz).

EXE

(Para detalles, ver la Página 12).

Esta operación deberá llevarse a cabo. Si el número de memorias es menor que el tamaño de la matriz, aparecerá un error como resultado de la operación.

■ Matrices Unidimensionales

Una matriz unidimensional es una composición de 0 a 199 elementos con sus respectivos subíndices.

Ejemplo $A(0), A(1), \dots, A(199)$

Ejemplo 1: Hacer un programa para visualizar el valor de i en una matriz unidimensional $A(i)$ a medida que cambia de 1 hasta 10.

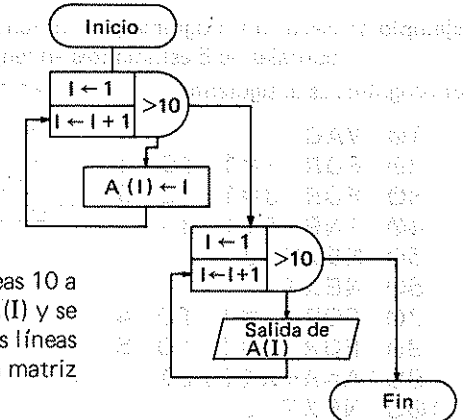
El programa se muestra a continuación.

```

10 FOR I=1 TO 10
20 A(I)=I
30 NEXT I
40 FOR I=1 TO 10
50 PRT A(I)
60 NEXT I
70 END

```

Diagrama de flujo



Las instrucciones generales FOR•NEXT que aparecen en las líneas 10 a 30 ya han sido explicadas. Sin embargo, en la línea 20 se usa $A(I)$ y se asigna a I un valor. Las instrucciones generales FOR•NEXT de las líneas 40 a 60 visualizan, de modo similar, el valor del contenido de la matriz usando la instrucción PRT de la línea 50.

Es necesario, por otro lado, ampliar el número de memorias a 20 antes de la ejecución de este programa.

DEFM 20 EXE

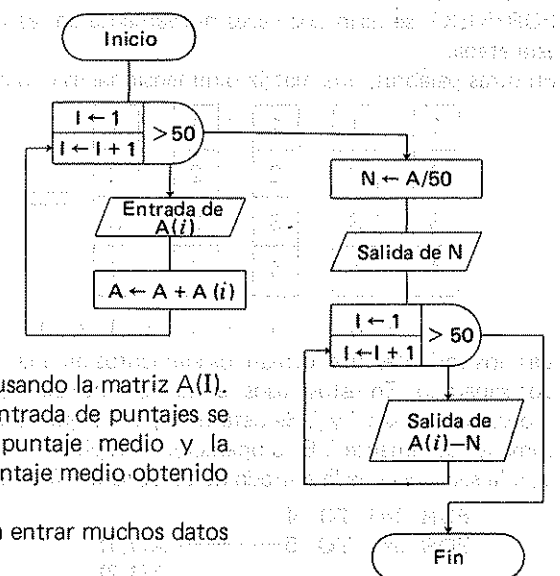
De este modo, la matriz almacena datos en forma de variables separadas simplemente cambiando los elementos, sin modificar los nombres de las variables. Se puede decir, entonces, que esta función es muy conveniente cuando se la usa junto con instrucciones FOR•NEXT.

Ejemplo 2: Hacer un programa para visualizar la diferencia entre el puntaje medio y los puntajes individuales entrando cada puntaje de los exámenes de 50 alumnos.

```

10 A=0
20 FOR I=1 TO 50
30 INP A(I)
40 A=A+A(I)
50 NEXT I
60 N=A/50
70 PRT N
80 FOR I=1 TO 50
90 PRT A(I)-N
100 NEXT I
110 END

```



En este programa, los puntajes se entran en la línea 30 usando la matriz $A(I)$. La suma total se obtiene usando la línea 40. Para la entrada de puntajes se usa una matriz (separada). Se calcula entonces el puntaje medio y la diferencia entre los datos almacenados (puntajes) y el puntaje medio obtenido y se visualizan.

De esta manera, gracias al uso de una matriz, se pueden entrar muchos datos con el simple uso de un programa corto.

En caso de no haberse usado una matriz, el programa sería de la siguiente manera:

```

A(1) ← A
A(2) ← B
A(3) ← C

```

y la instrucción general de entrada sería

INP A, B, C,

y se consumiría mucho tiempo en la separación de los datos.

■ Matrices Bidimensionales

Una matriz bidimensional es una combinación de matrices unidimensionales múltiples. El formato es $A(i, j)$, siendo A el nombre de la matriz e i y j los elementos. Las variables utilizadas como nombres de variables son similares a las variables usadas en las matrices unidimensionales. i y j son los elementos, siendo estos expresiones numéricas.

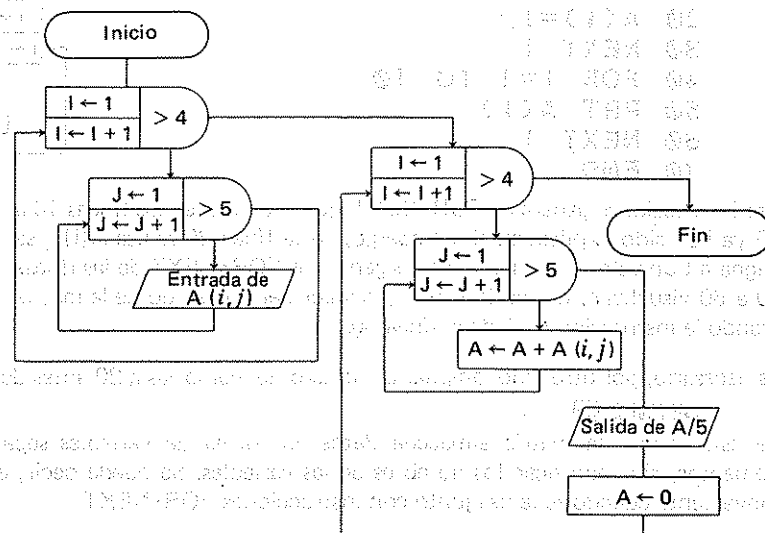
Ejemplo 1: Hacer un programa para obtener el puntaje medio de cada materia usando la suma total de los puntajes de 5 estudiantes en lenguaje, matemáticas, física y sociología.

El programa es el siguiente.

```

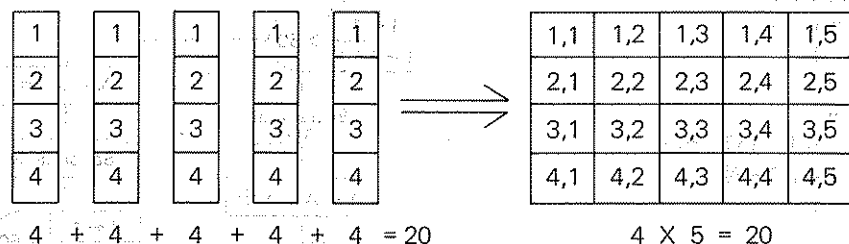
10 VAC
20 FOR I=1 TO 4
30 FOR J=1 TO 5
40 INP A(I, J)
50 NEXT J
60 NEXT I
70 FOR I=1 TO 4
80 FOR J=1 TO 5
90 A=A+A(I, J)
100 NEXT J
110 PRT J
120 A=0
130 NEXT I
140 END

```



Las porciones de entrada y salida se pueden dividir en 2 partes. Las instrucciones FOR•NEXT se usan dos veces en las líneas 20 a 60 y llevan a cabo la entrada en la matriz bidimensional. Las instrucciones FOR•NEXT se usan dos veces nuevamente en las líneas 70 a 130, dándose salida al puntaje promedio en esta etapa.

En otras palabras, una matriz bidimensional es una combinación de varias matrices unidimensionales.



Las dos variables que forman los elementos de una matriz bidimensional pueden cambiarse conjuntamente o por separado. En estos casos, es conveniente usar las instrucciones generales FOR•NEXT dos veces. En este programa se usan i y j . Se determina, antes que nada, i y luego se cambia j . A continuación, se cambia i y se cambia nuevamente j . Esta operación se repite en forma cíclica. No importa cual de los dos entra primero (i o j); la secuencia de la entrada de datos será diferente.

```

FOR I=1 TO 4
FOR J=1 TO 5 → A(1,1)
A(1,2)
A(1,3)
A(1,4)
A(1,5)
A(2,1)
...
A(4,4)
A(4,5)

```

```

FOR J=1 TO 5
FOR I=1 TO 4 → A(1,1)
A(2,1)
A(3,1)
A(4,1)
A(1,2)
...
A(3,5)
A(4,5)

```

Consideremos una matriz bidimensional con las variables $A(1,1)$ o $A(2,3)$ dispuestas en un formato matricial.

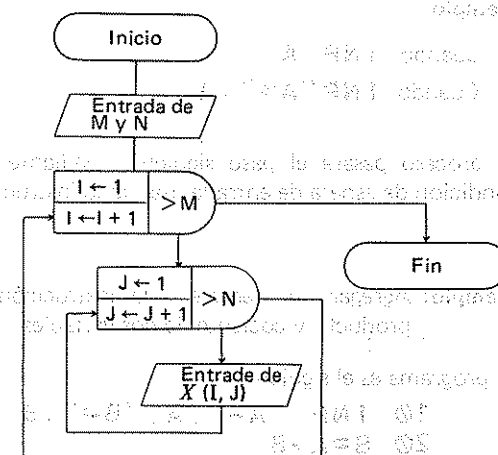
Ejemplo 2: Hacer un programa para reunir datos con el fin de obtener la suma total de la matriz. El número de datos se entra posteriormente.

Este programa se usa para almacenar todos los datos con el objeto de obtener la suma total del reticulado en una matriz bidimensional.

```

10 VAC
20 INP N,M
30 FOR I=1 TO M
40 FOR J=1 TO N
50 INP A(I, J)
60 NEXT J
70 NEXT I
80 END

```



Aquí, el programa es solamente para entradas. En realidad, este programa no termina después de la línea 80, sino que continúa.

Las instrucciones generales FOR•NEXT de las líneas 30 a 70 se usan para entrar datos del mismo modo que en el Ejemplo 1.

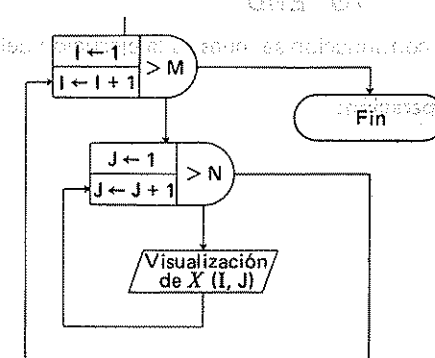
Ya que este programa representa sólo la porción de entrada, agreguemos la porción para el cálculo y salida según el uso que se le va a dar en realidad.

Para visualizar todos los datos, por ejemplo:

```

10 VAC
20 INP N,M
30 FOR I=1 TO M
40 FOR J=1 TO N
50 INP A(I, J)
60 NEXT J
70 NEXT I
80 FOR I=1 TO M
90 FOR J=1 TO N
100 PRT A(I, J)
110 NEXT J
120 NEXT I
130 END

```



Esta parte se agrega a partir de la línea 80. El método de uso es igual que el de las instrucciones FOR•NEXT de las líneas 30 a 70. La única diferencia es que se reemplaza o substituye la entrada por la salida.

De este modo, se puede decir que una matriz bidimensional es una especie de variable. Tiene, además, dos subíndices adjuntos al nombre de la variable.

Por ejemplo, la variable se determina por el valor numérico de i y j en de $A(i, j)$.

4-5-3 Comandos de Entrada/Salida

Este capítulo describe y explica los comandos de entrada/salida usados con esta calculadora.

■ Comandos de Entrada

Los comandos de entrada son aquellos que se utilizan para entrar datos durante la ejecución de programas y usan la instrucción INP y la funciones KEY.

● **Instrucciones INP**

La instrucción general INP se usa para entrar datos manualmente en las variables durante la ejecución de programas cuando se visualiza en la pantalla el signo "? ", durante la condición de espera de entrada.

Esta instrucción INP está compuesta por:

INP ["secuencia ordenada de caracteres"], variable, ["secuencia ordenada de caracteres"], variable . . .

Estas secuencias ordenadas de caracteres pueden omitirse. Sin embargo, una vez entradas, se visualizarán en la pantalla antes de entrar en la condición de espera de entrada. Estas pueden transformarse en mensajes. Las variables pueden ser numéricas, de caracteres o de caracteres exclusivas (\$), mientras que los datos de entrada se asignan.

Ejemplo:

Cuando **INP** A

Cuando **INP "A="**, A

?	N, M	981.02
A=? *	LY	805.58

El proceso pasará el paso siguiente mediante la entrada de datos y pulsando la tecla **EXE** durante la condición de espera de entrada usando la instrucción general INP.

Ejemplo: Agregar un mensaje a la instrucción INP en un programa para determinar la suma, resta, producto y cociente de dos variables.

El programa es el siguiente:

```

10 INP "A=",A,"B=",B
20 S=A+B
30 D=A-B
40 P=A*B
50 Q=A/B
60 PRT S,D,P,Q
70 END

```

A continuación se muestra la ejecución del programa.

Operación:

RUN	EXE	A=?
45	EXE	B=?
23	EXE	68
	CONT	22
	CONT	1035
	CONT	1.956521739

De este modo, las secuencias ordenadas de caracteres escritas en la instrucción general INP se visualizan en forma de mensajes facilitando la entrada por el teclado.

Si en esta instrucción general INP se entran caracteres para las variables numéricas, se presentará un error: Si se cancela dicho error por medio de la tecla **AC**, aparecerá en la pantalla el signo "?." nuevamente indicando el retorno a la condición de espera de entrada.

Por otro lado, la entrada para la variable numérica puede ser en forma de expresión numérica. En cuanto a la cancelación de la condición de espera de entrada, esta no podrá llevarse a cabo con sólo presionar la tecla **AC**; deberá usarse el siguiente procedimiento:

1. Si se presiona la tecla **EXE** directamente sin entrar ningún dato, "STOP" se visualizará en la pantalla y se detendrá la ejecución del programa. Si en este caso se presiona la tecla **AC**, se podrá cancelar la condición de espera de entrada. Para continuar con el programa, presiónese la tecla **CONT**.
2. Si se presionan las teclas **MODE** **⏏** para volver a designar el modo RUN, se cancelará la condición de espera de entrada. En este caso, no se podrá continuar con el programa.

● Funciones KEY

Esta función KEY lee un carácter por cada tecla presionada durante la ejecución de programas.

A diferencia de la instrucción general INP, esta función no puede utilizarse durante la condición de espera de entrada.

Si a medida que el programa avanza no se hace ninguna entrada desde el teclado, no pasará nada.

La función KEY está compuesta por:

variable de caracter = KEY

El o los caracteres leídos por la función KEY se asignan a la variable de carácter designada

Ejemplo:

```

10 A$=KEY: IF A$="" THEN 10
20 IF A$="1" THEN 100
30 IF A$="2" THEN 200
40 IF A$="3" THEN 300
50 GOTO 10

```

Este programa es solamente para la entrada y separación de datos; mediante el uso de la función KEY en la línea 10, se prueba si el o los datos de caracteres leídos se entraron o no por el teclado para su uso en la siguiente instrucción general IF.

Aún cuando no se presiona la tecla **EXE**, esta función KEY podrá leer la primera entrada por teclado. Sin embargo, esta función no se detiene del mismo modo que la instrucción general INP. Por medio de la combinación de esta función con la siguiente instrucción general IF se obtiene, entonces, una condición de espera de entrada.

Las líneas 20 a 40, por su lado, comparan el contenido de las variables de caracteres leídas y determinan los destinos de los saltos. En este caso, la función KEY puede leer sólo el carácter de una tecla.

■ Comandos de Salida

En los comandos de salida se dispone de una instrucción general PRT para visualizar resultados de cálculos o datos y una instrucción general DMS, la cual se utiliza para visualizar el valor numérico de la expresión numérica después de la conversión a grados, minutos y segundos (sexagesimales). Se dispone, también, de una instrucción general WAIT que se utiliza para determinar el tiempo de visualización en pantalla.

● Instrucción General PRT

Una instrucción general PRT se compone de la siguiente manera:

$$\text{PRT} \left[\begin{array}{c} \text{Función de control de salida} \\ \left(\begin{array}{c} ; \\ ; \end{array} \right) \end{array} \right] \left\{ \begin{array}{c} \text{Expresión numérica} \\ \text{Expresión de caracteres} \end{array} \right\} \left(\begin{array}{c} ; \\ ; \end{array} \right) \dots \dots \dots$$

En cuanto al contenido entre $\{ \}$, se pueden usar ambas alternativas.

Todo lo encerrado entre [] puede omitirse.

Esta instrucción general PRT visualiza el valor de la expresión numérica o de la variable numérica, además de las secuencias ordenadas de caracteres que se encuentran entre comillas (" ") o el contenido de la variable de caracteres.

Ejemplo:

```

10 INP "A="; A; "B="; B
20 C=INT (A/B)
30 D=A-B*C
40 PRT A; "/" ; B; "=" ; C; "..."; D
50 GOTO 10

```

Este programa calcula el resto, como en el cociente y resto cuando se divide A por B. Los valores de las variables se visualizan como en la línea 40. Las expresiones entre comillas (" ") se visualizan en forma de caracteres. Por otro lado, el modo de visualización depende del uso de ";", " " o ":" entre las variables y las secuencias ordenadas de caracteres.

Cuando se usa ";", los datos se visualizan secuencialmente, o sea que, cada dato se visualiza después de visualizado su precedente.

En el caso de emplearse ":", en cambio, cada dato se visualiza después de haber borrado la visualización del anterior.

Cuando no se designa el tiempo de visualización en pantalla por medio de la instrucción general WAIT y se utiliza ":", "STOP" se visualizará en la pantalla y se detendrá el proceso. Para continuar con la visualización, presiónese la tecla **CONT**.

• Instrucción General DMS

La instrucción general DMS es un comando para visualizar el valor numérico de la expresión numérica o la variable después de toda conversión a sexagesimal. Visualiza los ítems decimales en grados, minutos y segundos.

Ejemplo:

```

10 INP N
20 DMS N
30 GOTO 10

```

Este programa es para entrar algunos datos. Estos datos se convierten, como están, a sexagesimal. Aún si no se usa el comando "PRT", se puede llevar el dato a la pantalla con sólo la instrucción general DMS. A continuación se muestra la forma de visualización.

000°00'00.00"

Segundos (de 0,00 a 59,99)
Minutos (de 0 a 59)
Grados (de 0 a 99999)

Cuando el valor de la expresión numérica es mayor que 99999, se visualiza en forma de decimal. La parte decimal de los segundos se visualizará con redondeo a tres dígitos.

• Instrucción General WAIT

La instrucción general WAIT es un comando para determinar el tiempo de visualización cuando se usan las instrucciones PRT o DMS. El uso de esta instrucción evita la necesidad de emplear la tecla **CONT** para continuar con la visualización. Lleva a cabo la visualización durante determinado período de tiempo y luego procede al siguiente programa.

A continuación se muestra el formato de la instrucción general WAIT:

WAIT expresión numérica (0 ≤ expresión numérica ≤ 1000)

La porción decimal de la expresión numérica se elimina. La visualización se interrumpirá en el momento que se determina multiplicando el valor por aproximadamente 0,05 segundos.

Cuando no se lleva a cabo la designación al usar esta instrucción WAIT, o cuando el valor de la expresión numérica es mayor que 1000, se entrará en un estado de detención total. "STOP" se visualizará en la pantalla y se pasa luego a un estado de espera de entrada.

Ejemplo:

```

10 WAIT 5
20 FOR I=1 TO 100
30 PRT I
40 NEXT I

```

Este programa visualiza el valor desde 1 hasta 100. Cuando no se hace ninguna designación al usar la instrucción en cuestión, se detiene el proceso después de visualizar el valor de I. El siguiente valor se visualiza presionando la tecla **CONT**. Sin embargo, cuando al usarse la instrucción WAIT está designado el tiempo de visualización, la aparición en pantalla será cada aproximadamente 0,25 segundos.

La designación empleada en esta instrucción WAIT se cancela cuando se apaga la calculadora (ya sea manualmente o por la función de apagado automático) o en el momento de inicio de la ejecución de programas.

■ Funciones de Control de Salida

Las funciones de control de salida designan la posición y el tipo de salida.

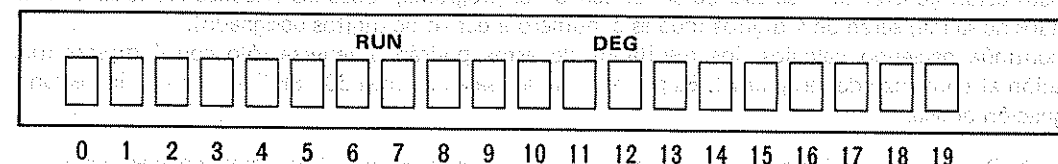
• Función CSR

Esta es una función de control usada en las instrucciones generales PRT. Su función es la de designar la posición de salida dentro de la pantalla (la cual tiene 20 posiciones) mediante el uso de la instrucción PRT. A continuación se muestra el formato de la función CSR:

PRT CSR { : } expresión numérica (La porción decimal de la expresión numérica se elimina, siendo el valor, entonces, de 0 a 19).

En esta función, se usa el valor de la expresión numérica para determinar a partir de qué unidad (desde la izquierda) de la pantalla se visualizarán los datos.

A continuación se muestra el método de conteo de las unidades en la pantalla.



Ejemplo:

```

10 INP X
20 PRT X;CSR 8;X↑2
30 GOTO 10

```

Este programa obtiene el cuadrado de un dato ya entrado y visualiza tanto el dato como el resultado. En la línea 20 de este programa se emplea una instrucción general PRT con la función CSR. El dato inicial "X" se visualiza como está (a la izquierda), mientras que "X↑2" se visualiza desde la octava posición, contando desde la izquierda, mediante la función CSR.

Al ejecutarse este programa, el mismo se visualizará del siguiente modo:

Operación:

RUN EXE	?
7 EXE	7
CONT	?
45 EXE	45 2025
CONT	?
852 CONT	852 725904

De este modo, resulta más fácil observar lo visualizado en la pantalla gracias a la visualización en dos posiciones.

* Cuando a continuación de "CSR" se encuentra ":", la visualización se hace desde la posición asignada; sin embargo, cuando se usa ":", no habrá visualización si no se borra una vez la pantalla. Lo que significa que, la visualización no se hace en forma continua.

● Instrucción General SET

Esta instrucción, como ya se explicó previamente, se usa para designar el número efectivo de dígitos y para la designación de decimales en los cálculos manuales; en esta sección se explicará como se lleva a cabo la designación escribiéndola en el programa.

En el ejemplo que a continuación se muestra del formato de la instrucción SET tenemos E, que designa el número efectivo de dígitos, F, que designa los decimales y N, que cancela las designaciones.

SET $\left\{ \begin{matrix} E n \\ F n \\ N \end{matrix} \right\}$ (n puede ser de 0 a 9)

Ejemplo 1:

```
10 SET E4
20 MODE 4:WAIT 10
30 FOR X=0 TO 180 STEP 5
40 PRT SIN X
50 NEXT X
60 SET N
70 END
```

Este programa obtiene el sen x a incrementos de 5 grados (del valor de x) desde 0 hasta 180 grados. En este ejemplo, para obtener un número efectivo de dígitos igual a 4, se lleva a cabo el sen x usando "SET E4".

Si la instrucción general SET se escribe al comienzo del programa, todas las visualizaciones subsiguientes (resultados de salida) serán de 4 dígitos (que es el número efectivo de dígitos designado).

Si se continúa haciendo cálculos, los resultados de estos podrán obtenerse sólo con 4 dígitos (por la designación al comienzo del programa). Es por eso que se usa en la línea 60 "SET N", con el fin de cancelar la designación previa.

Ejemplo 2: Designación de 4 dígitos para las posiciones de decimales para el programa del ejemplo 1.

```
10 SET F4
20 MODE 4:WAIT 10
30 FOR X=0 TO 180 STEP 5
40 PRT SIN X
50 NEXT X
60 SET N
70 END
```

De este modo, se puede controlar el número de dígitos de salida, gracias al uso de la instrucción general SET en forma similar a la aplicación que se le da en los cálculos manuales.

● Control del Formato de Salida

Los resultados de los cálculos o cómputos no siempre se pueden visualizar con un formato específico, ya que la posición del punto decimal y el número de dígitos implicados influyen según los valores de los resultados. En este caso, para facilitar la lectura, se puede hacer uso del control de formato con el objeto de especificar el tipo de formato de salida.

Los símbolos utilizados son: "#", "↑" y "↑". "#" se utiliza para especificar el número de dígitos en la mantisa (parte entera); mientras que "↑" se usa para designar la visualización de los exponentes.

##.###↑ (# puede ser igual o menor que 10)

Designación de la visualización exponencial
Número de dígitos en la parte decimal
Posición del punto decimal
Número de dígitos de la porción entera

El símbolo "#" a la izquierda del punto decimal designa el número de dígitos de la porción entera.

El símbolo "↑" a la derecha del punto decimal designa el número de dígitos de la porción decimal.

El símbolo "↑", al final de la expresión, designa la visualización para exponentes.

Formato	Explicación	Visualización cuando	X = 0.123456789 Y = 12.3456789
① #	Parte entera de 1 dígito, sin parte decimal	X = 0 Y = #	
② #.###	Parte entera de 1 dígito, parte decimal de 3 dígitos	X = 0.123 Y = #.###	
③ ##.#####	Parte entera de 2 dígitos, parte decimal de 5 dígitos	X = 0.12346 Y = 12.34568	
④ ##.##↑	Parte entera de 2 dígitos, parte decimal de 2 dígitos, con porción exponencial	X = 1.23E-01 Y = 1.23E 01	
⑤ #.###↑	Parte entera de 1 dígito, parte decimal de 3 dígitos, con porción exponencial	X = 1.235E-01 Y = 1.235E 01	

* Para aquellos casos como en los ejemplos ① y ②, si el número de enteros de la mantisa es desbordado, no se podrá llevar a cabo la visualización, apareciendo en la pantalla el formato de salida mismo.

Ejemplo:

```
10 MODE 4:WAIT 10
20 FOR X=30 TO 180 STEP 30
30 PRT SIN X: COS X
40 NEXT X
50 END
```

Este programa se utiliza para obtener sen x y cos x cada 30 grados. Ambos se visualizan en forma continua. Sin embargo, si se lo deja como está, la visualización de ambos causará una presentación irregular difícil de leer. En este caso, con el fin de normalizar la pantalla y cuando no se necesitan 10 dígitos para la mantisa, se podrá facilitar la lectura usando el control de formato.

Si se le agrega al programa el control de formato, este quedaría tal como se muestra a continuación:

```
10 MODE 4:WAIT 10
20 FOR X=30 TO 180 STEP 30
30 PRT ##.##↑; SIN X; CSR 10; COS X
40 NEXT X
50 END
```

Gracias al uso de este control de formato, ambos resultados pueden visualizarse simultáneamente, permitiéndose, además el ordenamiento de las posiciones de salida. De este modo se facilita la interpretación de lo visualizado.

La ejecución del programa es la siguiente:

Operación:

RUN [EXE]

5.00E-01	8.66E-01
8.66E-01	5.00E-01
1.00E 00	0.00E 00
8.66E-01	-5.00E-01
5.00E-01	-8.66E-01
0.00E 00	-1.00E 00

4-5-4 Funciones de Caracteres

Cuando hablamos de funciones de caracteres, nos referimos a variables de caracteres (A\$, B\$, \$, etc.). La forma en que se emplean estas es diferente, ya que se usan para entrar caracteres en lugar de valores numéricos. Estas funciones de caracteres determinan, por lo tanto, el tamaño de las variables de caracteres y operan para extraer algunos caracteres de secuencias ordenadas de caracteres en variables de caracteres.

LEN y MID

La función LEN es un comando para contar el número de caracteres en una variable de caracteres. Se puede saber, de este modo, el tamaño de la variable de caracteres.

La función MID es un comando para extraer algunos caracteres de la variable de caracteres exclusiva (\$). Se utiliza para reordenar las variables de caracteres.

Ya que la función LEN puede darnos a conocer el tamaño de una variable de caracteres, resulta ser un medio adecuado para la reservación de espacio de visualización (de la pantalla) para variables de caracteres o para su división cuando su tamaño no está establecido.

Ejemplo:

```
10 A$="123"
20 B$="456"
30 C$=A$+B$
40 D=LEN(C$)
50 PRT D
60 END
```

Este programa visualiza el tamaño (número de caracteres) de la variable de caracteres mediante la adición de las variables de caracteres A\$ y B\$.

Las variables de caracteres solamente pueden llevar a cabo sumas. No obstante, como no se conoce el tamaño de la variable de caracteres usando dicho resultado, se usa la función LEN con el fin de determinar el tamaño de tal suma.

Para la función LEN se pueden usar solamente variables de caracteres.

A continuación se muestra el formato de la función MID:

MID (m[, n]) (m, n: expresión numérica)

Extrae la porción del carácter n desde el m-ésimo carácter correspondiente a la secuencia ordenada de caracteres almacenada en la variable de carácter exclusiva (\$).

Si se omite n, se extraerán todos los caracteres después del m-ésimo carácter.

Por otro lado, tanto para n como para m se elimina la parte decimal, siendo sus valores de 1 a 30. Si la suma de n y m es mayor que el número de variables de caracteres almacenado más uno (1), se presentará un error.

Ejemplo: Cuando \$="ABCDEFGHIJ"

Extracción de cuatro caracteres a partir del tercero → MID (3,4) ... CDEF

Extracción de dos caracteres a partir del primero → MID (1,2) ... AB

Extracción de todos los caracteres a partir del quinto → MID (5) ... EFGHIJ

Ejemplo:

```
10 INP $
20 N=LEN($)/2
30 X$=MID(1,N):Y$=MID(N+1)
40 PRT X$,Y$
50 END
```

Este programa divide las secuencias ordenadas de caracteres de entrada en dos, usando las funciones LEN y MID, llevando luego a cabo la visualización.

Esto determina el tamaño de la variable de carácter exclusiva \$ que se encuentra en la línea 20. Las mitades se asignan a N y la primera y segunda mitades se dividen usando la función MID.

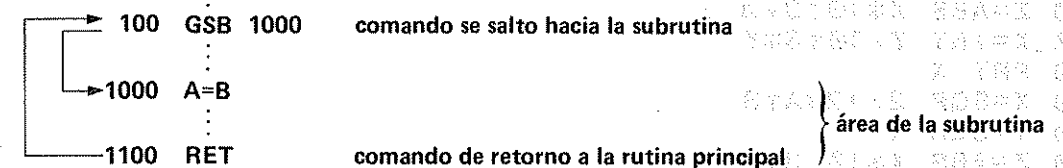
De este modo, cuando no está establecido el tamaño de la secuencia ordenada de caracteres de entrada, no se podrán extraer las mitades o terceras partes, razón por la cual se determina el tamaño usando la función LEN.

Por otro lado, en este caso, las secuencias ordenadas de caracteres que pueden entrarse en \$ están limitadas a 30 caracteres.

Sin embargo, como el número de caracteres en X\$ y Y\$ está limitado a 7 caracteres (cada uno), el número de caracteres de entrada en \$ se limita a 14.

4-5-5 Subrutinas

Una subrutina (también llamada subprograma) posee una rutina principal diferente a la de los programas (también llamados rutinas principales) previamente explicados. Toda subrutina es llamada desde la rutina principal y, al completarse su operación, se vuelve a la posición original de la rutina principal. En otras palabras, usándose el comando (GSB) para saltar desde la rutina principal a la subrutina, se salta de dicha posición a la subrutina designada. Una vez completada la operación de la subrutina, se vuelve a la posición (comando GSB) en la rutina principal desde donde se saltó hacia la subrutina, continuándose de este modo con el proceso de la rutina principal.



Los comandos requeridos para esta subrutina son las instrucciones generales "GSB" y "RET". El GSB es para saltar de la rutina principal hacia la subrutina, mientras que para retornar a la rutina principal se utiliza la instrucción general RET.

Esta instrucción general GSB se utiliza con los siguientes formatos:

- (1) **GSB expresión numérica** (se elimina la porción decimal de la expresión numérica y se usan los números de línea de 1 hasta 9999).
- (2) **GSB # expresión numérica** (se elimina la porción decimal de la expresión numérica y se usa $0 \leq n < 10$).

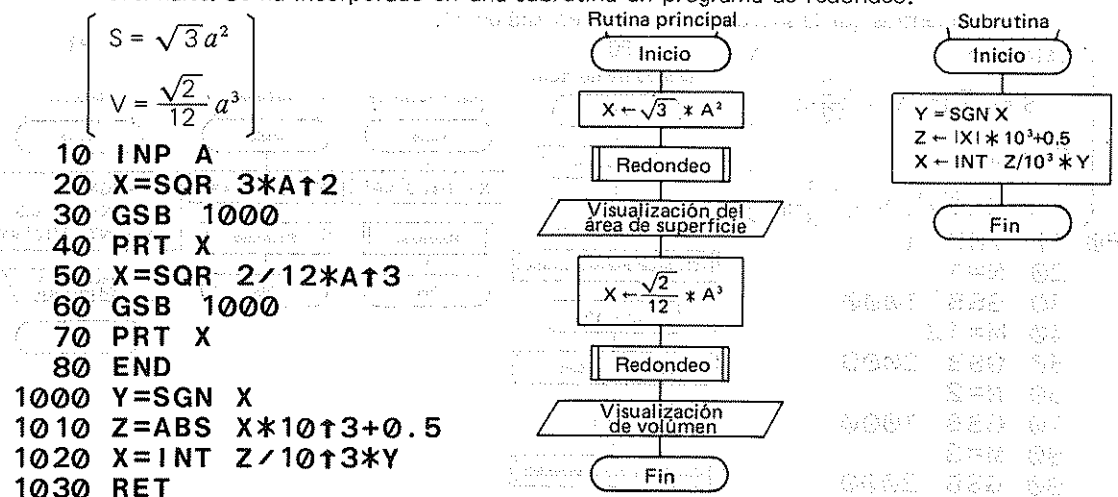
El primer método utiliza designación directa para escribir el número de línea directamente después de "GSB" y puede designar la ubicación de la subrutina indirectamente usando el valor contenido en la variable.

El segundo método utiliza una subrutina en otra área de programa (P0 a P9) para designar directa e indirectamente. Por otro lado, ninguna de las dos volverán si no se escribe al final de la subrutina la instrucción general RET.

■ Programas Fundamentales con Subrutinas

Se explicará en esta sección el método de disposición de las subrutinas usándose, para tal fin, ejemplos prácticos de programas.

Ejemplo 1: Hacer un programa para determinar el área de superficie y el volumen de un tetraedro (pirámide triangular), cuyos lados tienen un largo igual a A, haciéndose el cálculo a tres decimales. Se ha incorporado en una subrutina un programa de redondeo.



La rutina principal para el cálculo del área de superficie y el volumen del tetraedro es fácil de comprender ya que es similar a los programas considerados anteriormente. La diferencia consiste en la inclusión del programa de redondeo en la rutina principal, el cual sería la subrutina que comienza desde la línea 1000. Esta subrutina se encarga de computar el área de superficie y el volumen, tal como se determina en las líneas 20 y 50, a tres decimales por redondeo.

Las subrutinas pueden usarse tantas veces como se desee.

En este programa se utilizan dos veces; primero para el redondeo del valor del área de superficie y segundo para el redondeo del valor del volumen.

Si no se hubiera utilizado la subrutina y, en cambio, se hubiera empleado sólo una rutina principal (como en los ejemplos que se dieron hasta ahora), el programa sería mucho más largo, tal como se muestra a continuación:

```

10 INP A
20 X=SQR 3*A↑2
30 Y=SGN X
40 Z=ABS X*10↑3+0.5
50 X=INT Z/10↑3*Y
60 PRT X
70 X=SQR 2/12*A↑3
80 Y=SGN X
90 Z=ABS X*10↑3+0.5
100 X=INT Z/10↑3*Y
110 PRT X
120 END

```

Este programa y el que utiliza la subrutina, previamente descrito, llevan a cabo el mismo proceso (operaciones), pero, como se puede observar, el programa que usa la subrutina es más corto y más simple.

En otras palabras, las subrutinas se usan para simplificar el programa, eliminando operaciones reiterativas.

* En este ejemplo, como el programa de redondeo está incluido en una subrutina, no se incluye en esta última "PRT X". En este programa, en realidad, las líneas 30 a 60 y 80 a 110 son iguales, habiéndose incorporado esta porción en una subrutina.

Esta subrutina se considera, también, parte de la rutina principal, por lo cual no está incorporada dentro de esta, sino que está después del final ("END"), debiéndose terminarla con una instrucción general RET. Si se la incorpora dentro del programa mismo, se la interpretará como la rutina principal, procesándose la repetidamente y causando un error al ejecutarse la instrucción general RET.

Por lo tanto, como en este ejemplo, el área de la subrutina deberá estar separada comenzándola a partir de la línea 1000 ó 5000.

La subrutina es similar al comando GOTO (instrucción GOTO) y es considerada como un comando, con la única diferencia que esta siempre tiene adjunta una instrucción general RET que la hace volver a la posición de la instrucción GSB una vez terminada.

Ejemplo 2: Hacer un programa para determinar el área de superficie y el volumen de un tetraedro y un octaedro que tienen lados con un largo equivalente a A; hágase el cálculo a 4 decimales. La rutina principal y la subrutina para determinar el área y el volumen están incorporadas en P0, mientras que la subrutina de redondeo está en P1.

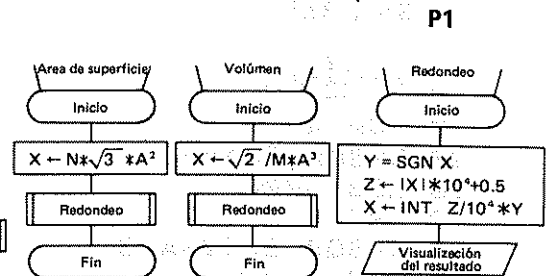
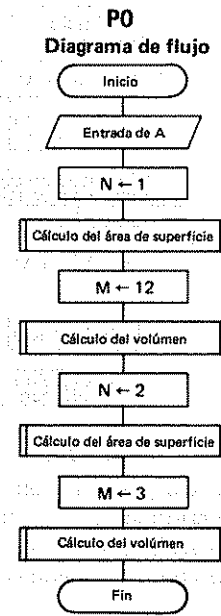
Tetraedro
 $S = \sqrt{3} a^2, V = \frac{\sqrt{2}}{12} a^3$

Octaedro
 $S = 2\sqrt{3} a^2, V = \frac{\sqrt{2}}{3} a^3$

```

P0 10 INP A
20 N=1
30 GSB 1000
40 M=12
50 GSB 2000
60 N=2
70 GSB 1000
80 M=3
90 GSB 2000
100 END
1000 X=N*SQR 3*A↑2
1010 GSB #1
1020 RET
2000 X=SQR 2/M*A↑3

```



```

2010 GSB #1
2020 RET
P1 10 Y=SGN X
20 Z=ABS X*10↑4+0.5
30 X=INT Z/10↑4*Y
40 PRT X
50 RET

```

En este programa se utilizan tres subrutinas: dos partes comunes para el cálculo de las áreas de superficie y los volúmenes y un programa de redondeo.

Usando este método, se pueden incorporar operaciones de cómputo de uso común en una subrutina para una ejecución repetitiva con sólo cambiar los valores de N y M. De este modo, si se incorporan partes comunes en subrutinas, se puede ahorrar memoria y facilitar la interpretación del método de cómputo.

Es por eso que en las líneas 1010 y 2010 de este programa se salta de una subrutina a otra. Esto es lo que se denomina "inclusión". Se llama una subrutina desde una similar con el fin de llamar otra subrutina desde una rutina principal.

Esta inclusión puede realizarse hasta 10 niveles (10 veces), apareciendo un error siempre que se supere este límite.

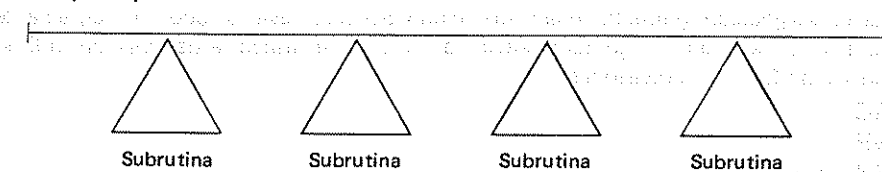
En otras palabras, se salta de la rutina principal a la subrutina del primer nivel. Esta subrutina actúa como la rutina principal y se salta a la siguiente subrutina (de segundo nivel). Aún en estos casos, la instrucción RET sobre la cual se basa la subrutina deberá estar escrita al final de cada subrutina sin falta.

Del mismo modo que la subrutina de redondeo, esta subrutina puede incorporarse a otra área de programa (en este caso, P1). Esta medida es muy práctica, ya que permite el uso de esta subrutina en común con otros programas (programas escritos en otras áreas de programa).

De este modo, se pueden usar muchas subrutinas en un programa.

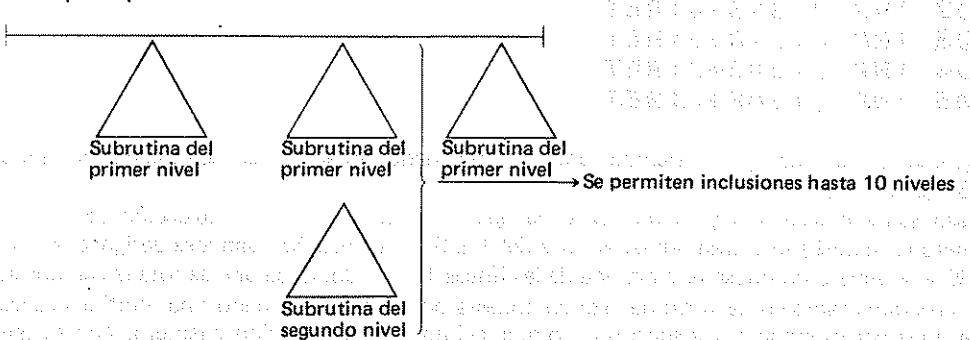
En forma similar a la inclusión, se permiten hasta 10 niveles.

Rutina principal



Gracias a este método, se pueden usar muchas subrutinas.

Rutina principal



Se permiten inclusiones hasta 10 niveles

De este modo, gracias al uso de subrutinas, se puede evitar todo tipo de redundancias. Por otro lado, en programas complicados, se puede usar una subrutina cuando hay partes separadas en grupos.

El comando GSB en una subrutina se lo considera en forma similar a la instrucción general GOTO. En otras palabras, determina el área de subrutina a la cual se salta desde la rutina principal y luego opera para volver a la posición original. Cuando la subrutina está escrita en el mismo programa, es necesario escribir END al final de la rutina principal, y si está repetida, es necesario escribir una instrucción general GOTO.

■ Designación Indirecta de una Subrutina

En el caso de las subrutinas, es posible llevar a cabo designación indirecta en forma similar a como se explicó para las instrucciones GOTO.

El método de uso es similar al de las instrucciones generales GOTO. Se salta a una subrutina designada y se vuelve a la posición original.

Ejemplo 1:

```

10 INP N
20 GSB N+100
30 PRT X*6
40 END
101 X=5:RET
102 X=10:RET
103 X=15:RET
104 X=20:RET
105 X=25:RET
106 X=30:RET
107 X=35:RET
108 X=40:RET
109 X=45:RET
110 X=50:RET

```

En este programa, la subrutina se designa indirectamente usando el valor de la entrada N. Cuando N está entre 1 y 10, se pasa a una subrutina entre 101 y 110, se determina el valor de la variable X y se visualiza el resultado de $X \times 6$.

De este modo, como la designación indirecta de la instrucción GSB determina el destino del salto (subrutina) usando el valor de la variable, cuando la variable es igual a 1, se salta a la primera subrutina (línea 101). Cuando la variable es igual a dos, se salta a la segunda subrutina (línea 102), y así sucesivamente. Se va designando indirectamente según el valor de la variable.

Ejemplo 2: Usar la designación indirecta de la instrucción GSB para llevar a cabo un programa de totales clasificados dado en un ejemplo previo, en el cual se utilizó la designación indirecta de la instrucción GOTO (5 divisiones).

```

10 VAC
20 INP I
30 GSB I+100
40 GOTO 20
100 PRT A,B,C,D,E:END
101 INP J:A=A+J:RET
102 INP J:B=B+J:RET
103 INP J:C=C+J:RET
104 INP J:D=D+J:RET
105 INP J:E=E+J:RET

```

Este programa es casi igual que el ejemplo previo mencionado, en el cual se usó designación indirecta de la instrucción general GOTO.

En este caso, se reemplaza la instrucción GOTO del ejemplo anterior por una instrucción GSB.

En este método también se entran números clave del 1 al 5 en la línea 20, usándose designación indirecta en la línea 30 y se separa usándose la subrutina de las líneas 101 a 105. Cada uno se totaliza en una subrutina. Para este programa también cada rutina está compuesta por una línea usando una multi-instrucción, o sea que, cada línea corresponde a una subrutina, lo que facilita su interpretación y lectura. Aún en este caso, la entrada del número clave se lleva a cabo en la línea 20, siendo este de 1 a 5. Si se utiliza cualquier otro número, se presentará un error y no habrá salto.

4-5-6 Funciones Generales

Las funciones generales son: funciones trigonométricas (seno, coseno y tangente, y las funciones incorporadas). Estas funciones se forman mediante la combinación de caracteres alfabéticos del teclado. Se pueden entrar, también por medio de los "comandos de una tecla" (por medio de la pulsación de una sola tecla), usándose éstas tan fácilmente como cualquier calculadora electrónica pequeña. Las funciones generales pueden usarse manualmente o incorporándolas en programas.

Nombre de la función	Formato	Ejemplo
Función trigonométrica	$\sin x$ $\cos x$ $\tan x$	SIN 30 SIN A SIN (N+5) COS 3.14 COS I COS (L-3) TAN 70 TAN F TAN (F X 2)
Función trigonométrica inversa	$\sin^{-1} x$ $\cos^{-1} x$ $\tan^{-1} x$	ASN 0.07 ASN P ASN (Z+Y) ACS $\pi/5$ ACS D ACS (C-X) ATN 6.5 ATN V ATN (Q+0.5)
Función hiperbólica	$\sinh x$ $\cosh x$ $\tanh x$	HSN 26.8 HSN T HSN (T+E) HCS 1.45 HCS 0 HCS (R/2) HTN 0.861 HTN H HTN (U+0.9)
Función hiperbólica inversa	$\sinh^{-1} x$ $\cosh^{-1} x$ $\tanh^{-1} x$	AHS 76.2 AHS G AHS (J+1) AHC 16 AHS W AHS (S X 2) AHT 0.91 AHT M AHT (H-E)
Raíz cuadrada	$\sqrt{x}$	SQR 69 SQR A SQR (R X S)
Función exponencial	e^x	EXP 8 EXP F EXP (D+L)
Logaritmo neperiano	$\log_e x$	LN 43 LN P LN (U+T)
Logaritmo común	$\log_{10} x$	LOG 24.6 LOG R LOG (G+15)
Integración (Entero máximo sin exceder x)	INT x	INT 347.457 INT V INT (Q+U) INT -45.43
Fraccionamiento (X con su parte entera extraída)	FRAC x	FRAC 73.54 FRAC N FRAC (H+B)
Valor absoluto	x	ABS -9.43 ABS L ABS (K/P)
Signos (si $x > 0$, 1) (si $x = 0$, 0) (si $x < 0$, -1)	SGN x	SGN 79 SGN E SGN (P-0)
Conversión de sexagesimal a decimal	DEG (Grados Minutos y Segundos)	DEG (39, 34, 15) DEG T DEG (R+5) Grados Minutos Segundos
Conversión de decimal a sexagesimal	DMS x	DMS 12.582 DMS Z DMS (P+0.05)
Conversión de coordenadas Cartesianas a Polares Polares a Cartesianas	RPC x, y PRC r, θ	RPC 14, 20.7 RPC A, B PRC 4.5, $2/\pi$ PRC F, F

Nombre de la función		Formato	Ejemplo
Especificación de la cifra significativa (X se determina al 10 ^Y -ésima cifra significativa por redondeo)		RND (X, Y)	RND (123.456, 2) RND (A, C) RND (F+E, G-5)
Generación de números aleatorios (Generación de números aleatorios uniformes entre 0 < x < 1)		RAN #	RAN #
Factoriales		x !	8 ! A ! (I+J) !
Unidad de medida angular	Grados	MODE 4	Un ángulo recto = 90 grados
	Radianes	MODE 5	Un ángulo recto = $\frac{\pi}{2}$ radianes
	Gradientes	MODE 6	Un ángulo recto = 100 gradientes

* x e y pueden ser constantes, variables o expresiones numéricas.

Estas funciones pueden usarse en un programa directamente ya que están incorporadas.

Ejemplo 1: Hacer un programa para obtener el largo de un lado de un triángulo usando el ángulo formado por los dos lados restantes.

$$[C = \sqrt{a^2 + b^2 - 2ab \cos \theta}]$$

```

10 MODE 4
20 INP A,B,C
30 D=SQR (A↑2+B↑2-2*A*B*COS DEG C)
40 PRT D
50 GOTO 20

```

El MODE 4 aparece en la línea 10 de este programa, ya que en la función trigonométrica se utilizan "grados" como la medida angular. Se entran luego el largo de los dos lados y el valor del ángulo por estos formado.

De acuerdo a la fórmula, se escriben en la expresión numérica la raíz cuadrada SQR y el coseno COS.

Ejemplo 2: Hacer un programa para averiguar la ganancia en dB (decibeles) de un amplificador que tiene una tensión de entrada igual a X y una tensión de salida igual a Y.

$$[dB = 20 \cdot \log_{10} \frac{Y}{X}]$$

```

10 INP X,Y
20 Z=20*LOG (Y/X)
30 PRT Z
40 GOTO 10

```

Si la tensión de entrada es X y la de salida es Y, la fórmula para averiguar la ganancia (Z) puede escribirse tal como se muestra en la línea 20.

Ejemplo 3: Hacer un programa para generar números aleatorios de tres dígitos usando la función de generación de números aleatorios y la de especificación de la cifra significativa.

```

10 N=RND(RAN#, -4)*1000
20 PRT N
30 END

```

Como los números aleatorios pueden generarse con mantisas de 10 dígitos dentro de lo delimitado en $0 < x < 1$, deberán tomarse tres dígitos como significativos y multiplicarlos por 1000 para obtener números de tres cifras.

4-5-7 Procesos Estadísticos

Esta calculadora tiene capacidad para llevar a cabo procesos estadísticos por facilidades incorporadas en ella (del mismo modo que las funciones incorporadas), posibilitando una fácil realización de estos procesos. Esta capacidad para procesos estadísticos incluye cálculos de desviación estándar, cálculos de análisis de regresión para la determinación de coeficientes de correlación, etc.

■ Cálculos de Desviación Estándar

El cálculo de la desviación estándar puede realizarse manualmente o por medio de un programa escrito. En este capítulo se explica el cálculo de la desviación estándar por medio de programas.

En cuanto al cálculo manual de la desviación estándar, remitirse a la sección correspondiente a "Cálculos Manuales".

Ejemplo 1: Hacer un programa para averiguar la diferencia entre cada dato y el promedio, almacenando los datos mediante el uso de una matriz unidimensional.

```

10 SAC
20 INP "N=", N
30 FOR I=1 TO N
40 INP D
50 STAT D
60 A(I)=D
70 NEXT I
80 PRT "MX="; MX
90 FOR I=1 TO N
100 PRT "NO. "; I; "="; A(I)-MX
110 NEXT I
120 END

```

El comando SAC, que aparece en la línea 10 de este programa, es un comando que borra la memoria de sumas estadísticas. El comando VAC no borra la memoria para cálculos estadísticos.

Es necesario usar este comando antes de llevar a cabo cualquier cálculo estadístico. El número de datos se entra en la línea 20 y, por medio de la instrucción general FOR-NEXT, se determina el número de veces que se van a entrar los datos.

En la instrucción general FOR-NEXT siguiente, se totalizan los datos y se almacenan todos ellos en la matriz A(I), por medio del uso del comando de entrada de datos "STAT" para cálculos de desviación estándar.

Después de la entrada de datos, como MX es el comando de obtención del valor medio, se usa nuevamente una instrucción general FOR-NEXT para restar de cada dato almacenado en la matriz el valor medio y para visualizar el resultado.

Llevemos a cabo el siguiente ejemplo usando este programa:

Ejemplo: número de datos = 8

Datos: 55, 54, 51, 55, 53, 53, 54, 52

Si se lleva a cabo la ejecución con estos datos, el resultado será el siguiente:

Operación:

DEFM1 EXE

VAR: 36 PRG: 1600

RUN EXE

N=?

8 EXE

?

55 EXE

?

54 EXE

?

:

52 EXE

MX= 53.375

CONT

NO. 1= 1.625

CONT

NO. 2= 0.625

CONT

NO. 3=-2.375

CONT

NO. 4= 1.625

CONT

NO. 5=-0.375

CONT

NO. 6=-0.375

CONT

NO. 7= 0.625

CONT

NO. 8=-1.375

* Cuando se usa la matriz de este modo, no olvidarse de ampliar la capacidad de la memoria.

Por otro lado, los comandos usados para obtener el resultado de la desviación estándar son, en este ejemplo, los siguientes:

CNT número de datos (n)
 SX sumatoria ($\sum x$)
 SX2 sumatoria de cuadrados ($\sum x^2$)
 MX media ($\bar{x}$)
 SDX desviación estándar (σ_{n-1})
 SDXN desviación estándar (σ_n)

Ejemplo 3: Hacer un programa para averiguar el puntaje medio y la desviación estándar σ_n , como así también el valor de desviación del puntaje de cada estudiante.

Remitirse al ejemplo 1
 Valor de desviación $SS = \frac{(x - \bar{x}) \cdot 10}{\sigma_n} + 50$

```
10 SAC
20 INP N
30 FOR I=1 TO N
40 INP D
50 STAT D
60 A(I)=D
70 NEXT I
80 PRT "MX=";MX,"SDXN=";SDXN
90 FOR I=1 TO N
100 S=(A(I)-MX)*10/SDXN+50
110 PRT "NO.";I;"=";S
120 NEXT I
130 END
```

En este programa, de la línea 10 a la 70, el contenido es el mismo que el del ejemplo 1; sin embargo, a partir de la línea 80, que es el programa para cálculo y visualización del problema, es diferente. El valor de desviación SS se obtiene en la línea 100. Este valor de desviación se obtiene para cada puntaje (de los estudiantes) usando la instrucción general FOR-NEXT.

Del modo descripto, se puede incorporar directamente al programa el promedio o desviación estándar mediante el uso de la función de desviación estándar ya incorporada.

Por otro lado, para entrar con frecuencia, se lleva a cabo la operación en forma similar al cálculo manual. Además, cuando se incorpora la entrada de datos a un programa, la anulación y corrección de datos equivocados deberá hacerse manualmente.

■ Análisis de Regresión y Coeficientes de Correlación

Además del cálculo de la desviación estándar, se incorpora en forma de función de control estadístico una función para el cálculo del análisis de regresión (estadística de 2 variables).

Para el cálculo de análisis de regresión, sólo se incorpora la regresión lineal. Por medio del uso de esta regresión lineal y de un programa, se pueden, no obstante, obtener regresiones logarítmicas, exponenciales y de potencia.

Cálculos de Regresión Lineal

- La fórmula para regresión lineal es: $y = A + Bx$. El término constante A y el coeficiente de regresión B se calculan tal como se muestra a continuación:

$$A = \frac{\sum y - B \cdot \sum x}{n} \quad B = \frac{n \cdot \sum xy - \sum x \cdot \sum y}{n \cdot \sum x^2 - (\sum x)^2}$$

- El coeficiente de correlación r del par de datos de entrada se calcula usándose la siguiente fórmula:

$$r = \frac{n \cdot \sum xy - \sum x \cdot \sum y}{\sqrt{\{n \cdot \sum x^2 - (\sum x)^2\} \{n \cdot \sum y^2 - (\sum y)^2\}}}$$

- Estas fórmulas se manejan en el programa igual que en el cálculo de desviación estándar.

Ejemplo: La entrada de datos se lleva a cabo manualmente. Usando el resultado del cómputo obtenido, ensamblar un programa para calcular el término constante A, el coeficiente de regresión B, el coeficiente de correlación r , el coeficiente de determinación (r^2) y la covarianza:

$$\left(\frac{\sum xy - n \cdot \bar{x} \cdot \bar{y}}{n-1} \right)$$

```
10 PRT "A=";LRA,"B=";LRB,"R=";COR,"R12=";COR12
20 X=(SXY-CNT*MX*MY)/(CNT-1)
30 PRT X
40 END
```

Como este programa es sólo un programa de visualización única, la entrada de datos se lleva a cabo manualmente.

A continuación se muestran dos métodos de entrada de datos.

dato x, dato y [STAT]

dato x, dato y: frecuencia [STAT]

El "LRA" aquí utilizado corresponde al término constante A, "LRB" al coeficiente de regresión B y "COR" al coeficiente de correlación r . Además, el coeficiente de determinación y la covarianza se calculan usando las fórmulas y luego se visualizan.

Por otro lado, las funciones para la visualización de los resultados son las siguientes:

CNT número de datos (n)
 SX sumatoria de x ($\sum x$)
 SY sumatoria de y ($\sum y$)
 SX2 sumatoria de x al cuadrado ($\sum x^2$)
 SY2 sumatoria de y al cuadrado ($\sum y^2$)
 SXY sumatoria del producto de datos ($\sum xy$)
 MX promedio de x ($\bar{x}$)
 MY promedio de y ($\bar{y}$)
 SDX desviación estándar de x (σ_{n-1})
 SDY desviación estándar de y (σ_{n-1})
 SDXN desviación estándar de x (σ_n)
 SDYN desviación estándar de y (σ_n)
 LRA término constante (A)
 LRB coeficiente de regresión (B)
 COR coeficiente de correlación (r)
 EOX valor estimado de x ($\hat{x}$)
 EOY valor estimado de y ($\hat{y}$)

De este modo, como el análisis de regresión se calcula automáticamente con sólo usar los datos de entrada, es muy útil para programas comunes o programas de visualización.

Cuando se entran los datos manualmente, asegurarse de presionar $\boxed{F1}$ antes de llevar a cabo la entrada de los mismos.

Cálculos de Regresión Logarítmica

- La fórmula de regresión logarítmica es: $y = A + B \cdot \ln x$.
- Esta regresión logarítmica se lleva a cabo usando la función para el cálculo de regresión lineal. El método para la entrada de datos es diferente al usado en la regresión lineal, lo que lo hace muy práctico para entrar datos al programa.

Ejemplo: Ensamblar un programa usando la función de análisis de regresión lineal para llevar a cabo el análisis de regresión logarítmica solamente entrando los datos x e y .

La salida del resultado del análisis se realiza manualmente y no se incluye en el programa.

```
10 SAC
20 INP X,Y
30 STAT LN X,Y
40 GOTO 20
```

Este es un muy simple programa, compuesto solamente por una porción de entrada. Está compuesto por el comando SAC, el cual borra la memoria de suma de datos, la instrucción general INP y la instrucción STAT para la entrada de datos. Esta instrucción general STAT utiliza el logaritmo $\ln$ de x .

Como la fórmula de regresión es $y = A + B \cdot \ln x$, y puede entrarse como está. Sin embargo, x deberá entrarse como el logaritmo de x .

De este modo, se ensambla un programa para su parte de entrada solamente. La salida se puede también llevar a cabo manualmente. En este programa, sin embargo, es necesario después de la entrada de datos detener el programa (presionando **MODE** **(0)**) y presionar la tecla de función para dar salida.

Por otro lado, el número de datos deberá entrarse inicialmente con el fin de obtener terminación automática y poder visualizar el resultado del análisis del mismo modo que en el ejemplo de desviación estándar. Este programa tiene una frecuencia igual a 1. Para posibilitar la entrada de frecuencias, ensámblase el programa de la siguiente manera:

```
10 SAC
20 INP X,Y,Z
30 STAT LN X,Y;Z
40 GOTO 20
```

Sin embargo, como en este programa no se puede omitir la entrada de la frecuencia, aún cuando esta es igual a 1, no olvidarse de entrar "1".

Cálculos de Regresión Exponencial

- La fórmula de regresión exponencial es: $y = A \cdot e^{B \cdot x}$ ($\ln y = \ln A + B \cdot x$).
- Esta regresión exponencial se lleva a cabo usando la función de análisis de regresión lineal, del mismo modo que en la regresión logarítmica.

Ejemplo: Ensamblar un programa de entrada/salida para el análisis de regresión exponencial. La salida visualiza los términos constantes, los coeficientes y los coeficientes de correlación.

```
10 SAC
20 INP "DATA=",N
30 FOR I=1 TO N
40 INP "X=",X,"Y=",Y
50 STAT X, LN Y
60 NEXT I
70 PRT "A=";EXP LRA,"B=";LRB,"R=";COR
80 END
```

Las líneas 10 a 60 de este programa usan un método similar al de los programas anteriores. Sin embargo, el dato Y no se usa como está, sino que se entra como logaritmo $\ln y$. y deberá entrarse como el logaritmo de y ($\ln y$), ya que la fórmula de regresión es $\ln y = \ln A + B \cdot x$. Esta parte de entrada lleva a cabo la entrada inicial del número de datos usando una instrucción general FOR-NEXT.

El término constante A de la porción de salida se computa como $\ln A$, con el fin de usarlo como está. Para obtener A , es necesario obtener el antilogaritmo de $\ln A$ y calcular $\text{EXP } A$ (e^A) usando la función exponencial. El resultado corresponde al término constante A . Los coeficientes de regresión y de correlación, B y r respectivamente, pueden usarse como están.

Cálculos de Regresión de Potencia

- La fórmula de regresión de potencia es: $y = A \cdot x^B$ ($\ln y = \ln A + B \cdot \ln x$).
- Esta regresión de potencia también se lleva a cabo usando la función de análisis de regresión lineal.

Ejemplo: Ensamblar un programa para el análisis de regresión de potencia con visualización de los términos constantes, los coeficientes de regresión y de correlación como resultado del análisis, obteniéndose, de este modo, los valores estimados.

```
10 SAC
20 INP "DATA=",N
30 FOR I=1 TO N
40 INP "X=",X,"Y=",Y
50 STAT LN X, LN Y
60 NEXT I
70 PRT "A=";EXP LRA,"B=";LRB,"R=";COR
80 INP "X OR Y",Z$
```

```
90 IF Z="Y" THEN 130
100 INP "Y=",Y
110 PRT EXP EOY LN Y
120 GOTO 80
130 INP "X=",X
140 PRT EXP EOY LN X
150 GOTO 80
```

Las líneas 10 a 70 de este programa tienen el mismo tipo de configuración de entrada/salida que el ejemplo anterior; la única diferencia se observa en la instrucción general STAT para la entrada de datos que se encuentra en la línea 50. Se entran ambos datos x e y como logaritmos $\ln$. Es necesario entrar el logaritmo de cada dato (x e y , en este caso) ya que la fórmula de regresión es $\ln y = \ln A + B \cdot \ln x$.

Como A en la parte de salida se obtiene en la forma de $\ln A$, el resultado es el antilogaritmo A . El programa para el cálculo del valor estimado, por su lado, presenta dos casos. Uno de ellos estima $\hat{y}$ por el dato x . El otro, estima $\hat{x}$ por el dato y . Se juzga, primero, en las líneas 80 y 90. En otras palabras, si $Z$$ es igual a X , se entra el dato y para obtener $\hat{x}$. Si $Z$$ es igual a Y , se entra el dato x para obtener $\hat{y}$.

De este modo, aún cálculos de regresión de potencia pueden llevarse a cabo fácilmente usando un programa. Si se comete un error al componer la parte de entrada de datos del programa, se puede corregir manualmente o incluir en el programa una operación de recuperación de error.

4-5-8 Contraseñas

En esta calculadora, se pueden asignar contraseñas con el fin de proteger los programas ya compuestos. Una contraseña es un "nombre clave". Cuando el programador o el operador del instrumento no desea que otras personas conozcan el contenido de cierto programa, o no quiere que otras personas modifiquen el mismo, se pueden incluir estas contraseñas, las cuales inhabilitan los comandos "LIST" o "CLR".

Se muestra a continuación el formato de una contraseña.

PASS "ABC"

Las letras o números entre comillas, en este caso "ABC", componen la contraseña. Se puede usar cualquier número o letra, con un máximo de 8 caracteres.

Ejemplos: PASS "CASIO"
PASS "#\$"
PASS "AZ-1"

Como las contraseñas son efectivas para cada área de programa (P0 a P9), se las pueden incluir sólo a P0 hasta P9.

Las contraseñas pueden borrarse borrando el programa en su totalidad mediante el uso del comando "CLR ALL" o volviéndose a asignar una nueva contraseña.

Se puede borrar transitoriamente el comando "LIST" colocando (entrando) una contraseña a continuación de dicho comando.

Ejemplo:

PASS "CASIO" **(EXE)** contraseña establecida

LIST "CASIO" **(EXE)** se cancela momentáneamente la contraseña (o LIST 20 "CASIO" **(EXE)**)

Esta cancelación momentánea de la contraseña es solamente efectiva para el comando LIST y en el momento de entrada. Si a continuación no se vuelve a entrar la contraseña, se presentará un error de contraseña.

4-5-9 Especificaciones Opcionales

■ CMT (cinta magnética de cassette)

A este equipo se puede conectar un grabador para usarlo como un dispositivo de memoria externa. Podrán protegerse muchos datos o programas importantes, ya que estas cintas (MT) pueden almacenar información. Varios son los tipos de grabadores que pueden usarse para grabar. Por otro lado, si el grabador tiene un terminal para control remoto, se podrá aprovechar dicha facilidad controlando desde la unidad principal a través del adaptador opcional FA-2, lo cual sería muy conveniente. En cuanto a la conexión de la calculadora al grabador, remitirse al manual de instrucciones del FA-2, donde se da información detallada sobre el tema.

En caso de utilizarse un CMT (grabador para cinta de cassette) sin este terminal de control remoto, deberá operársele manualmente siguiendo las instrucciones de operación de dicho grabador.

En cuanto al control remoto, se podrán controlar las operaciones de arranque y parada para la grabación y reproducción, no ocurriendo lo mismo con las operaciones de avance y rebobinado rápidos.

● Comandos SAVE y LOAD

A continuación se muestra el comando SAVE usado para almacenar programas.

SAVE [#n "nombre de archivo"] (n puede ser 0 a 9, los ítems entre paréntesis pueden omitirse).

#n indica el área de programa. Si es #0, se almacenará el programa que se encuentre en P0. Si es #3, se almacenará el programa que se encuentre en P3.

Para el nombre de archivo podrán usarse hasta 8 caracteres (letras, números o símbolos), los cuales pueden ser iguales o no.

Ejemplos:

"A"
"BBB"
"*****"

Tanto #n como el nombre de archivo podrán omitirse. Si se omite #n se almacenará el programa que se encuentre en el área de programa que se está usando en el momento que se ejecuta el comando SAVE.

Este comando SAVE almacenará aquel programa que se encuentre en el área de programa designada. Si se desean almacenar todos los programas escritos en las todas áreas de programa (P0 a P9), úsese el comando SAVE ALL (ver la página 77).

Procedimiento:

- (1) Montar una cinta y tomar nota de la indicación del contador.
- (2) Poner en marcha el CMT en la condición de grabación.
- (3) Operar SAVE [#n "nombre del archivo"] [EXE].
- (4) Una vez completada la grabación del programa, la cinta se detendrá automáticamente (si el CMT utilizado no tiene el terminal para control remoto, deténgase la cinta manualmente).

* El comando SAVE puede usarse sólo manualmente, por lo cual no se lo puede escribir en ningún programa.

A continuación se muestra el comando LOAD el cual se utiliza para la recuperación (o reproducción) de programas grabados.

LOAD [#n "nombre de archivo"] (n puede ser de 0 a 9; los ítems entre paréntesis pueden omitirse).

#n indica el área de programa del programa a reproducirse. Si es igual a P1, por ejemplo, el programa se reproducirá para cargarse en P1.

El nombre de archivo aquí utilizado es similar al usado con el comando SAVE. Este nombre de archivo puede omitirse. Si se lo omite, se cargará (se reproducirá) el programa escrito inicialmente después de la puesta en marcha.

Por otro lado, si se omite #n, el programa se cargará en el área de programa que se esté usando en el momento de la ejecución del comando LOAD.

Este comando LOAD puede utilizarse manualmente o incorporándolo en un programa.

Operación:

Cuando se lo utiliza manualmente:

- (1) Montar la cinta y prepararla como para iniciarla un poco antes de la posición designada, haciendo uso de la indicación del contador (dos o tres números antes).
- (2) Poner en marcha el CMT por medio del terminal remoto con el aparato en la condición de reproducción.
- (3) Operar LOAD [#n "nombre de archivo"] [EXE].
- (4) En el caso de usarse un CMT sin terminal de control remoto, póngaselo en marcha inmediatamente después de la ejecución del comando LOAD (en la condición de reproducción).
- (5) Una vez terminada la obtención del programa, la cinta se detendrá automáticamente (si el CMT no tiene el terminal de control remoto, detener la cinta manualmente).

Visualización durante la carga del programa **PF: AAA**
archivo del programa nombre del archivo

Visualización cuando se termina de recuperar el programa ... **READY Pn**

* Se podrá cargar el programa en cualquier área de programa, aún cuando en ésta haya todavía algún programa almacenado con anterioridad. En este caso, se cancelará el contenido del programa viejo a partir del primer número de línea del nuevo programa.

Ejemplo:

Programa actualmente en P0	Carga de un nuevo programa	Contenido de P0 después de la ejecución de LOAD
<pre>10 INP A } 90 PRT B 100 FOR I=1 TO 100 } 300 PRT C</pre>	<pre>100 INP X } 200 PRT Y</pre>	<pre>10 INP A } 90 PRT B 100 INP X } 200 PRT Y</pre>

Inscripción del comando LOAD en un programa:

- (1) Escribir el comando LOAD en el programa.
- (2) Poner en marcha el CMT por medio del terminal remoto en la condición de reproducción.
- (3) El programa con el comando LOAD escrito se inicia en el modo RUN (MODE 0).
- (4) En el caso de un CMT sin terminal remoto, póngaselo en marcha inmediatamente después de iniciado el programa usando la posición de reproducción.
- (5) Una vez completada la recuperación del programa, la cinta se detendrá automáticamente y se iniciará el programa cargado.

Cuando el comando LOAD se ejecuta por medio de su incorporación en un programa, una vez terminada la carga, el programa escrito se ejecutará secuencialmente.

Ejemplo:

Ejecución del programa Nombre de archivo "A" Nombre de archivo "B" Nombre de archivo "C"
P0: 10 LOAD#0 "A" 100 INP X 100 INP Y 100 INP Z
20 LOAD#0 "B" 500 GOTO 20 500 GOTO 30 500 END
30 LOAD#0 "C" 500 GOTO 20 500 GOTO 30 500 END

Para este tipo de programa, si se ejecuta la línea número 10, se leerá antes que nada el programa con el nombre de archivo "A". Luego se ejecuta el programa "A". Una vez terminado el programa "A", el control vuelve a la línea 20 y se lee y ejecuta el programa con el nombre de archivo "B". De este modo, cuando se lo incorpora en el programa, la lectura y ejecución se lleva a cabo en forma secuencial.

● Comandos PUT y GET

Todo almacenamiento de datos se lleva a cabo usando el comando PUT.

PUT ["nombre de archivo"] variable (los ítems entre paréntesis pueden omitirse)

Las variables son \$, A y B hasta T9.

Para almacenar las variables de A a Z, escríbase "A, Z".

Ejemplos: Para SAVE \$,A,B,C,D PUT ["nombre de archivo"] \$, A, D

Para SAVE A para Z PUT ["nombre de archivo"] A, Z

Para SAVE A para Z y A0 para T9 PUT ["nombre de archivo"] A, T9

PUT ["nombre de archivo"] A, A(199)

* Cuando se escriben variables, "A, Z" significa desde A hasta Z. El almacenamiento se lleva a cabo en la secuencia de A a Z, por lo cual no se podrá escribir en una secuencia de mayor a menor como ser, por ejemplo, de Z a A.

Ejemplo: PUT H, A → error

La recuperación de datos se lleva a cabo usando el comando GET.

GET ["nombre de archivo"] variable (los ítems entre paréntesis pueden omitirse)

La variable se escribe en forma de "\$, A" o "\$, A, Z", del mismo modo que en el ejemplo del comando PUT.

Tanto el comando PUT como el GET pueden usarse o ejecutarse manualmente o incorporándolos en un programa.

El procedimiento para la operación manual de estos es similar al de los comandos para programas SAVE y LOAD.

Para la ejecución por incorporación en un programa, y en el caso de usarse un CMT con terminal remoto, tanto la puesta en marcha como la detención pueden llevarse a cabo automáticamente. Para el caso de CMT sin terminal remoto, será necesaria una buena sincronización.

Ejemplo: Para almacenar Para la recuperación
10 GET "D" A, Z

490 PUT "D" A, Z
500 END

Para un programa como el de arriba, primero se lleva a cabo el almacenamiento de datos tal como se describe a continuación:

- (1) Montar la cinta y tomar nota de la indicación del contador.
- (2) Iniciar el CMT en la condición de grabación.
- (3) Iniciar el programa con el comando PUT escrito.
- (4) Al leerse el comando PUT, la cinta comenzará a moverse para detenerse automáticamente una vez terminado.

Luego, se lleva a cabo la recuperación de datos tal como se describe a continuación:

- (1) Colocar la cinta un poco antes de la posición designada haciendo uso de la indicación del contador.
- (2) Poner en marcha el CMT en la condición de reproducción.
- (3) Iniciar el programa con el comando GET escrito.
- (4) Al leerse el comando GET, la cinta comenzará a moverse y se detendrá automáticamente al terminarse.

Visualización durante el comando LOAD

VF: D

archivo de datos nombre del archivo

Por otro lado, el nombre de archivo puede omitirse en forma similar a los comandos de programa SAVE/LOAD.

● Comandos SAVE/LOAD para Programas/Datos Completos

Para los comandos SAVE/LOAD previamente mencionados, el programa se escribe en un área de programa y se obtiene de dicha área. Es posible también almacenar y cargar (con los comandos SAVE y LOAD) programas que abarquen varias áreas de programa (por ej.: de P0 a P3 ó de P2 a P6); lo mismo ocurre con la totalidad de las áreas de programas y datos.

A continuación se muestran los comandos de archivo para toda la memoria:

SAVE ALL ["nombre de archivo"]

LOAD ALL ["nombre de archivo"] (los ítems entre paréntesis pueden omitirse)

Esta operación se lleva a cabo de modo similar que con los comandos SAVE/LOAD para programa.

Visualización durante la recuperación total

del archivo de memoria

PVF: ZZ

totalidad del archivo de memoria nombre del archivo

* Si se ejecuta este comando LOAD ALL, se usará el comando DEFM para expandir la memoria de datos, ya que en este caso se leen todos los datos y programas almacenados en la cinta.

● Verificación del archivo escrito en la cinta

Para verificar la exactitud del programa o datos escritos en la cinta, se utiliza el comando VER.

VER ["nombre de archivo"] (los ítems entre paréntesis pueden omitirse)

El procedimiento de operación es similar al del comando LOAD para programas.

■ Impresora

Esta calculadora puede conectarse a una mini-impresora (FP-10). Se podrán obtener los listados de programas y datos si se conecta esta impresora. Se pueden imprimir en ella los resultados de los cálculos, etc. en el momento de la ejecución.

En cuanto a la operación y conexión de esta mini-impresora, remitirse a su manual de instrucciones correspondiente (FP-10).

Para imprimir los listados de programa o datos, presiónense **MODE** **7** para seleccionar el modo PRT ("PRT" en pantalla).

Por otro lado, para la impresión de los resultados de los cálculos, el modo PRT se puede designar manualmente o puede escribirse **MODE** **7** en el programa, lo que permitiría una impresión parcial.

La cancelación del modo PRT puede llevarse a cabo manualmente presionando **MODE** **8** o escribiendo en el programa lo mismo.

● Listado de Programa

Para imprimir un listado de programa, ejecútase el comando LIST en el modo RUN.

LIST { { número de línea } } { "contraseña" } (los ítems entre paréntesis pueden omitirse)

Cuando se escribe el número de línea, el listado se imprimirá a partir de dicho número de línea. Si se escribe #0 a 9, se imprimirá el listado del programa correspondiente área (P0 a P9).

Por otro lado, se requerirá la contraseña si esta está incluida. De lo contrario, no será necesaria.

● Listado de Datos

Para imprimir el listado de datos, ejecútase el comando LIST en el modo RUN.

LIST V

Usando este comando, se imprimirá todo el contenido de los datos (variables).

● Totalidad de los Listados de Programas/Datos

Para escribir simultáneamente el listado de programas y de datos de P0 a P9, ejecútase el comando LIST ALL en el modo RUN.

LIST ALL

El comando LIST ALL hace que se impriman todos los listados de programas contenidos en P0 a P9.

No obstante, las áreas de impresión sin ninguna entrada no se imprimirán.

En los programas con contraseña, se imprime la expresión PASS.

● Resultados de los Cálculos

Si se ejecuta el programa después de seleccionar el modo PRT, los resultados de los cálculos se imprimirán simultáneamente con su visualización, usando las instrucciones generales PRT y DMS.

Por otro lado, se podrá obtener la impresión de solamente el resultado deseado si se escribe en el programa la designación del modo PRT.

En cuanto a la salida del resultado del cálculo, si se usa la designación por la función CSR con anterioridad a la presente posición de impresión, esta se interrumpirá ya que la cabeza de impresión no puede desplazarse hacia atrás.

Lista de Mensajes de Error

Código de error	Significado	Causa	Medida Correctiva
1	Superación de la capacidad de la memoria o de la memoria de retención temporal del sistema.	- El número de pasos no es suficiente. No puede escribirse el programa. - Superación de la capacidad de la memoria de retención temporal.	- Borrar los programas innecesarios o reducir el número de memorias. - Dividir las fórmulas y hacerlas más simples.
2	Error de sintaxis	- Error de formato en el programa, etc. - Los formatos de la derecha y la izquierda de una instrucción general de asignación, etc. son diferentes.	Corregir todo error en el programa de entrada, etc.
3	Error matemático	- El resultado del cálculo de una expresión matemática excede el valor 10^{100}. - El argumento de una función numérica está fuera del límite de entrada. - El resultado es "indefinido" o "imposible".	- Corregir la fórmula o los datos del cálculo. - Verificar los datos.
4	Error de número de línea no definida	No se designó ningún número de línea a la instrucción general GOTO o GSB.	Corregir el número de línea designado.
5	Error de argumento	- El argumento de una función que lo requiere está fuera del límite de entrada. - El argumento de matriz no se encuentra dentro de los límites 0 a 199 para una matriz unidimensional y (0, 0) a (19, 9) para una matriz bidimensional.	Corregir el error de argumento.
6	Error de variable	Se está usando una variable no definida por el comando DEFN.	Incrementar la variable si fuere necesario.
7	Error de inclusión	- Aparece una instrucción RET cuando no se está ejecutando ninguna subrutina. - Aparece una instrucción NEXT cuando no se está en el lazo FOR. - El número de niveles de subrutinas es mayor que 10. - El número de lazos FOR-NEXT es mayor que 8.	- Anular las instrucciones RET o NEXT innecesarias. - Mantener las subrutinas o lazos de instrucciones FOR-NEXT dentro de los límites de nivel establecidos.
8	Error de contraseña	- Se intentó usar un comando inutilizado por una contraseña (CLR o LIST). - Se entró una contraseña diferente a la establecida.	Entrar la contraseña correcta.
9	Error de función opcional	Se intentó la ejecución de un comando como el SAVE en el modo PRT cuando no había ninguna impresora o CMT conectados.	- Conectar la impresora o el CMT. - Cancelar el modo PRT.

Lista de Comandos de Programa

Clasificación	Nombre del comando	Formato	Función
Instrucciones generales de entrada	INP	INP secuencia ordenada de variables	Permite la entrada de datos desde el teclado durante la ejecución de un programa. La ejecución del programa se interrumpe hasta después de terminada la entrada de dichos datos. P. 57
	KEY	Variable de caracteres = KEY	Lee el carácter entrado durante la ejecución de un programa y lo asigna a una variable de caracteres. Ya que este comando no interrumpe el programa, en caso de no hacerse ninguna entrada desde el teclado, no se asignará nada a la variable de caracteres. P. 58
Instrucciones generales de salida	PRT	PRT función de control de salida { ; } elemento de salida { ; } ...	Da salida al elemento de salida especificado bajo un formato específico. P. 58
	DMS	DMS variable	Convierte el valor de una variable a sexagesimal y visualiza el resultado de la conversión. P. 59
	WAIT	WAIT n * $0 \leq n < 1000$	Determina el tiempo de visualización para las instrucciones generales PRT y DMS. P. 59
Ramificación	GOTO	GOTO { número de línea } { variable }	Hace que el control salte al número de línea especificado. P. 39
	IF...{THEN}...	IF expresión comparativa { THEN número de línea } ; comando	Hace que el control salte al número de línea especificado a continuación de THEN o que ejecute el comando que se encuentre después del signo ";" si el resultado de la expresión comparativa es positivo. En aquellos casos en que el resultado de la expresión comparativa es negativa, hace que el control proceda a la línea siguiente del programa. P. 44
	GSB	GSB { número de línea } { variable }	Hace que el control salte a la subrutina ubicada en la posición especificada por el número de línea que se encuentra a continuación de GSB, ejecutándose la misma. Una vez ejecutada la subrutina, el control retorna a la instrucción general GSB gracias a la instrucción general RET, para proceder con el comando que se encuentre a continuación de GSB. P. 64

Clasificación	Nombre del comando	Formato	Función
	RET	RET	Indica el fin de una subrutina. De este modo, el control vuelve al número de línea correspondiente a la posición a continuación de la instrucción general GSB. P. 64
Lazos	FOR	FOR $v = e_1$ TO e_2 [STEP e_3] * v denota una variable numérica, mientras que e_1 , e_2 y e_3 representan expresiones numéricas, respectivamente. El lazo se repite " $\left\lceil \frac{e_2 - e_1}{e_3} \right\rceil + 1$ " veces entre las instrucciones generales FOR y NEXT. Si se omite e_3 , se toma un incremento de "1". P. 49	Declara el comienzo de un lazo, en el cual el valor numérico v varía desde el valor inicial e_1 hasta el valor final e_2 , a incrementos de e_3 .
	NEXT	NEXT	Indica el fin de un lazo FOR. Si el resultado de v más e_3 es igual o menor que e_2 , el lazo se repite. Si el resultado es mayor que e_2 , el control pasa a la línea a continuación de la instrucción general NEXT. P. 49
Detención de ejecución	STOP	STOP	Interrumpe en forma transitoria la ejecución de un programa para poner al sistema en un estado de espera de entrada desde el teclado. La ejecución podrá reanudarse presionándose la tecla CONT . P. 37
Fin de ejecución	END	END	Indica la finalización de un programa, haciendo que el sistema vuelva al estado previo a la ejecución del mismo. La ejecución de un programa finalizada por medio de este comando no podrá reanudarse presionándose la tecla CONT . P. 30
Borrado de datos	VAC	VAC	Borra todos los datos de variables de un programa. P. 42
Visualización del listado de programa	LIST	LIST [número de línea]	Visualiza un listado de todas las instrucciones generales de un programa desde el número de línea especificado en adelante. P. 32
Visualización del listado de datos	LIST V	LIST V	Visualiza un listado de los datos en las memorias. P. 78
Visualización del listado de programa/datos	LIST ALL	LIST ALL	Visualiza un listado de todas las instrucciones generales de un programa y de los datos en las memorias. P. 78
Ejecución de programas	RUN	RUN [número de línea]	Inicia la ejecución de un programa a partir del número de línea especificado. P. 31

Clasificación	Nombre del comando	Formato	Función
Borrado de programas	CLR	CLR	Borra el programa contenido en el área de programa en ése momento especificada. P. 30
	CLR ALL	CLR ALL	Borra todos los programas. P. 30
Protección de programas	PASS	PASS "contraseña"	Al entrarse una contraseña, se inhiben ciertas operaciones como las especificadas por LIST y LOAD. No obstante, la contraseña se cancelará si se vuelve a entrar la misma por segunda vez. P. 74

Especificaciones

los valores/
obtenidos

no se aplican

■ Tipo

FX-702P

■ Funciones para Cálculos Básicos

Números negativos, exponentes, operaciones aritméticas incluyendo paréntesis (suma, resta, multiplicación y división con función de discriminación de la secuencia de prioridades — lógica algebraica de verdad).

■ Funciones incorporadas

Funciones trigonométrica y trigonométrica inversa (unidad de medida angular en grados, radianes o gradientes), funciones hiperbólica e hiperbólica inversa, funciones logarítmica y exponencial, factoriales, raíces cuadradas, potencias, conversión de coordenadas, conversión a número entero, extracción de la parte entera, valor absoluto, codificación, conversión de decimal $\leftrightarrow$ sexagesimal, designación del número efectivo de dígitos, designación de decimales, números aleatorios y π .

■ Funciones para Cálculos Estadísticos

Desviación estándar: número de datos, sumatoria, sumatoria de cuadrados, promedios y desviación estándar (de dos tipos).

Desviación estándar: número de datos, sumatoria de x , sumatoria de y , sumatoria de x al cuadrado, sumatoria de y al cuadrado, sumatoria del producto de datos, promedio de x , promedio de y , desviación estándar de x (de dos tipos), desviación estándar de y (de dos tipos), término constante, coeficiente de regresión, coeficiente de correlación, valor estimado de x y valor estimado de y .

■ Comandos

INP, KEY, PRT, DMS, IF-THEN, GOTO, FOR, NEXT, GSB, RET, WAIT, STAT, DEL, SAC, MODE, SET, STOP, END, SAVE, SAVE ALL, LOAD, LOAD ALL, GET, PUT, VER, PASS, RUN, LIST, LIST V, LIST ALL, CLR, CLR ALL

■ Funciones de caracteres

LEN, MID

■ Función de control de salida

CSR

■ Gama de cálculos

$\pm 1 \times 10^{-99}$ hasta $\pm 9.999999999 \times 10^{99}$ y 0

En el cálculo interno se usa una mantisa de 12 dígitos

Capacidad de dígitos funcionales

Gama de entrada

Precisión

$\sin x$, $\cos x$, $\tan x$

$|x| < 1440^\circ$ (8π rad, 1600 gra)

10mo. dígito ± 1

$\sin^{-1} x$, $\cos^{-1} x$

$|x| \leq 1$

$\tan^{-1} x$

$\sinh x$, $\cosh x$

$|x| \leq 230$

$\tanh x$

$\sinh^{-1} x$

$|x| \leq 10^{99}$

$\cosh^{-1} x$

$1 \leq x \leq 10^{99}$

$\tanh^{-1} x$

$|x| < 1$

$\log x$, $\ln x$

$x > 0$

e^x

$|x| \leq 230$

$\sqrt{x}$

$x \geq 0$

$x!$

$x \leq 69$

x^y ($x \uparrow y$)

Cuando $x < 0$, y es un número natural

$R \rightarrow P$

$|x| \leq 10^{99}$, $|y| \leq 10^{99}$

$P \rightarrow R$

$|\theta| < 1440^\circ$ (8π rad, 1600 gra)

decimal $\rightarrow$ sexagesimal

Menor que ± 99999.999999

■ Sistema de Programa

Sistema de programa almacenado

■ Lenguaje de programa

BASIC

■ Número de pasos

Desde 80 hasta un máximo de 1680 pasos (con protección por alimentación)

■ Número de programas incorporados

Hasta 10 grupos (de P0 a P9)

■ Número de memorias

Desde 26 hasta un máximo de 226 memorias más la variable de carácter exclusiva (\$) (con protección por alimentación)

■ Número de memorias de retención temporal

subrutinas — 10 niveles

lazos FOR-NEXT — 8 niveles

valores numéricos — 10 niveles

elementos de cálculo — 20 niveles

■ Método y sistema de visualización

Pantalla de cristal líquido con mantisa de 10 dígitos (incluyendo el signo menos), parte exponencial de 2 dígitos, con visualización sexagesimal y de las condiciones F1, F2, ARC, HYP, RUN, WRT, STOP, DEG, RAD, GRA, TRACE y PRT.

■ Elementos de la pantalla

Pantalla de veinte dígitos con patrón de puntos (cristal líquido)

■ Componentes principales

LSI C-MOS y otros

■ Consumo de energía

0,01W (Max.)

■ Fuente de alimentación

2 baterías de litio (CR2032)

Esta unidad proporciona con la batería tipo CR2032 aproximadamente 240 horas (200 cuando se usan las unidades opcionales) de operación continua.

■ Apagado automático (Auto power-off)

La calculadora se apaga después de aproximadamente 8 minutos después de terminada la última operación.

■ Gama de temperatura ambiente

de 0°C a 40°C (32°F a 104°F)

■ Dimensiones

17Al x 165An x 82Pr (mm)

■ Peso

176 g incluyendo las baterías