

Capítulo

20

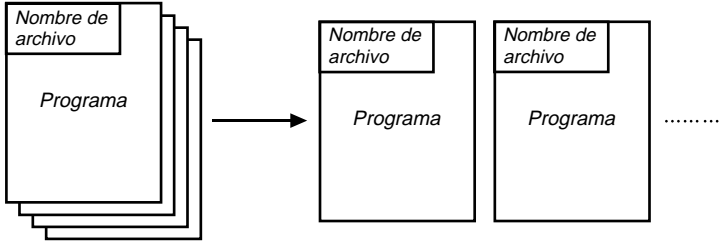


Programación

- 20-1 Antes de comenzar con la programación
- 20-2 Ejemplos de programación
- 20-3 Depurando un programa
- 20-4 Calculando el número de bytes usados por un programa
- 20-5 Función de secreto
- 20-6 Buscando un archivo
- 20-7 Buscando datos dentro de un programa
- 20-8 Editando nombres de archivo y contenidos de programa
- 20-9 Borrando programas
- 20-10 Mandos de programa prácticos
- 20-11 Referencia de mandos
- 20-12 Presentación de texto
- 20-13 Usando las funciones de la calculadora dentro de los programas

20-1 Antes de comenzar con la programación

La función de programación le permite realizar cálculos repetitivos y complejos de manera rápida y fácil. Los mandos y cálculos son ejecutados secuencialmente, exactamente como en las instrucciones múltiples del cálculo manual. Se pueden almacenar múltiples programas bajo los nombres de archivo para una fácil recuperación y edición.



Seleccione el icono **PRGM** en el menú principal e ingrese el modo PRGM. Al hacerlo, la lista de mandos y programas aparece sobre la presentación.

Seleccione el área de memoria —
(utilice las teclas ▲ y ▼
para cambiar de posición).

Program List	
OCIA	: 37
TRIANGLE	: 17
AREA *	: 33
GRAPHICS	: 17
MEASURE	: 17
OCTONARY	: 17
EXE EDIT NEW DEL DELA	▷



P.368

P.362



- {EXE}/{EDIT} ... {ejecutar}/{editar} programa
- {NEW} ... {programa nuevo}
- {DEL}/{DEL-A} ... borrar {programa específico}/{todos los programas}
- {SRC}/{REN} ... {búsqueda}/{cambio} de nombre de archivo
- {LOAD} ... {carga un programa de la biblioteca de programas incorporada}

* Para los detalles, vea el manual de biblioteca de programas **separado**.

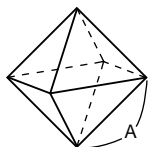
- Si no hay programas almacenados en la memoria cuando ingresa el modo PRGM, el mensaje **"No Programs"** aparece sobre la presentación, y en el menú de funciones solamente se muestra el ítem NEW (F3).

Los valores a la derecha de la lista de programa indican el número de bytes usados hasta ahora por cada programa.

20-2 Ejemplos de programación

Ejemplo 1 Calcular el área de una superficie y volumen de tres octaedros regulares con las dimensiones mostradas en la tabla siguiente.

Almacene la fórmula de cálculo bajo el nombre de archivo OCTA.



Longitud de un lado (A)	Area de superficie (S)	Volumen (V)
7 cm	cm ²	cm ³
10 cm	cm ²	cm ³
15 cm	cm ²	cm ³

Las siguientes fórmulas son las usadas para el cálculo del área de superficie S y volumen V de un octaedro regular, del cual se conocen la longitud y un lado.

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

Cuando se ingresa una fórmula nueva, primero se debe registrar el nombre de archivo y luego ingresar el programa real.

●Para registrar un nombre de archivo

Ejemplo Registrar el nombre de archivo OCTA.

- Tenga en cuenta que un nombre de archivo puede tener hasta ocho caracteres de longitud.

1. Visualice el menú de lista de programa y presione **F3** (NEW) para visualizar un menú, que contiene los ítemes siguientes.

- {RUN}/{BASE} ... ingreso de programa de {cálculo general}/{base numérica}
- {n0} ... {registro de contraseña}
- {SYBL} ... {menú de símbolo}

2. Ingrese el nombre del archivo.

O C T A

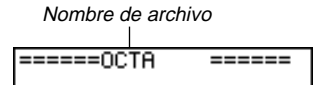
Program Name
[OCTA]

- El cursor cambia de forma para indicar el ingreso de caracteres alfabéticos.
- Los siguientes son los caracteres que pueden usarse para el nombre de un archivo: A hasta la Z, r, θ, espacios, [,], {, }, ', ", ~, 0 al 9, ., +, -, ×, ÷
- Tenga en cuenta, no obstante, que **[.01]** y **[]** no pueden ingresarse para el nombre de un programa que contiene cálculos con valores binarios, octales, decimales o hexadecimales.



P.360

- Utilice **F1** (RUN) para ingresar un programa para los cálculos generales (un programa a ser ejecutado en el modo COMP). Para los programas que relacionan especificaciones de sistema numérico, utilice **F2** (BASE). Tenga en cuenta que el ingreso de los programas luego de presionar **F2** (BASE) se indican por **B** a la derecha del nombre de archivo.
 - Presionando **F6** (SYBL) visualiza un menú de los símbolos (' , " , ~) que pueden ingresarse.
 - Mientras realiza un ingreso puede borrar un carácter, moviendo el cursor al carácter que desea borrar y presionando **DEL**.
3. Presione **EXE** para registrar el nombre de archivo y cambiar la pantalla de ingreso de programa.



- El registro de un nombre de archivo utiliza 17 bytes de memoria.
- La pantalla de ingreso de nombre de archivo permanece sobre la presentación si presiona **EXE** sin ingresar un nombre de archivo.
- Para salir de la pantalla de ingreso de nombre de archivo y retornar a la lista de programas sin registrar un nombre de archivo, presione **EXIT**.
- Cuando registra el nombre de un programa que contiene cálculos con valores binarios, octales, decimales o hexadecimales, el indicador **B** se fija a la derecha del nombre de archivo.

•Para ingresar un programa

Los ítemes siguientes están incluidos en el menú de función de la pantalla de ingreso de programa, que se usa para el ingreso de programa.

- **{TOP}/{BTM}** ... {parte superior}/{parte inferior} de programa
- **{SRC}** ... {búsqueda}
- **{MENU}** ... {menú de modos}
- **{SYBL}** ... {menú de símbolos}

•Para cambiar los modos dentro de un programa

- Presionando **F4** (MENU) mientras la pantalla de ingreso de programa se encuentra sobre la presentación ocasiona que aparezca el menú de cambio de modo. Puede usar este menú para ingresar cambios de modo dentro de sus programas.
- **{STAT}/{MAT}/{LIST}/{GRPH}/{DYNA}/{TABL}/{RECR}**

Para los detalles acerca de estos modos, vea la sección “Para seleccionar un icono”, así también como las secciones de este manual que describen lo que puede hacer en cada modo.

- El menú siguiente aparece siempre que presiona **F4** (MENU) mientras ingresa un programa que relaciona las especificaciones de bases numéricas.
- **{d ~ o}/{LOG}**



P.365

P.364



P.3



- Presionando **[F6]** (SYBL) visualiza un menú de los símbolos (' , ~ , * , / , #) que pueden ingresarse dentro de un programa.
- Presionando **[SHIFT]** **[SETUP]** visualiza un menú de mandos que pueden usarse para cambiar los ajustes de la pantalla de ajustes básicos dentro de un programa.
- **{ANGL}/{COOR}/{GRID}/{AXES}/{LABL}/{DISP}/{P/L} {DRAW}/{DERV}/ {BACK}/{FUNC}/{SIML}/{S-WIN}/{LIST}/{LOCS} {T-VAR}/{ΣDSP}/{RESID}**

Para los detalles en cada uno de estos mandos, vea la sección “Menús de teclas de funciones en la pantalla de ajustes básicos”.

El siguiente menú de teclas de funciones aparece si presiona **[SHIFT]** **[SETUP]** mientras ingresa un programa que contiene cálculos con valores binarios, octales, decimales o hexadecimales.

- **{Dec}/{Hex}/{Bin}/{Oct}**

Los contenidos de un programa real son idénticos a los cálculos manuales. A continuación se muestra cómo el cálculo del área de superficie y volumen de un octaedro regular serán calculados usando un cálculo manual.

Área de superficie S **[2]** **[X]** **[SHIFT]** **[✓]** **[3]** **[X]** <valor de A> **[x²]** **[EXE]**
 Volumen V **[SHIFT]** **[✓]** **[2]** **[÷]** **[3]** **[X]** <valor de A> **[∧]** **[3]** **[EXE]**

Este cálculo también puede realizarse asignando el valor para la longitud de un lado de la variable A.

Longitud de lado A
 <valor de A> **[⇐]** **[ALPHA]** **[A]** **[EXE]**
 Área de superficie S **[2]** **[X]** **[SHIFT]** **[✓]** **[3]** **[X]** **[ALPHA]** **[A]** **[x²]** **[EXE]**
 Volumen V **[SHIFT]** **[✓]** **[2]** **[÷]** **[3]** **[X]** **[ALPHA]** **[A]** **[∧]** **[3]** **[EXE]**

Si simplemente ingresa los cálculos manuales mostrados previamente, la calculadora los ejecuta desde el comienzo hasta el final, sin parar. Los siguientes mandos hacen posible interrumpir un cálculo para el ingreso de valores y para visualizar los resultados intermedios.

- ?: Este mando realiza una pausa en la ejecución del programa, y visualiza un signo de pregunta como un indicador de solicitud para el ingreso de un valor de asignación a una variable. La sintaxis para este mando es: ? → <nombre de variable>.
- ▲: Este mando realiza una pausa en la ejecución del programa, y visualiza el último resultado de cálculo obtenido o texto. Es similar a presionar **[EXE]** en un cálculo manual.



- Para los detalles completos en el uso de éstos y otros mandos, vea la sección “Mandos de programa prácticos”.

A continuación se muestran ejemplos de cómo se usan realmente los mandos ? y



SHIFT PRGM F4(?) → ALPHA A F6(▷) F5(:)

2 X SHIFT ✓ 3 X ALPHA A x²

F6(▷) F5(▲)

SHIFT ✓ 2 ÷ 3 X ALPHA A ^ 3

```
=====OCTA =====
?→A:2×√3×A²,
√2+3×A³_
```

SHIFT QUIT o EXIT EXIT

```
Program List
001H : ST
```

•Para ejecutar un programa

1. Mientras la lista de programas se encuentra sobre la presentación, utilice las teclas ▲ y ▼ para destacar en brillante el nombre del programa que desea ejecutar.
2. Presione F1 (EXE) o EXE para ejecutar el programa.

Veamos cómo funciona el programa que ingresamos anteriormente.

Longitud de un lado (A)	Area de superficie (S)	Volumen (V)
7 cm	169,7409791 cm ²	161,6917506 cm ³
10 cm	346,4101615 cm ²	471,4045208 cm ³
15 cm	779,4228634 cm ²	1590,990258 cm ³

```
Program List
001H : ST
```

F1 (EXE) o EXE

```
?
?
```

7 EXE
(Valor de A)

```
?
7
169.7409791
- DISP -
```

Resultado intermedio producido por ▲

EXE EXE

```
?
7
169.7409791
161.6917506
?
```

1 0 EXE

```
?
7
169.7409791
161.6917506
?
10
346.4101615
- DISP -
```

[EXE]

7	169.7409791
	161.6917506
?	
i0	346.4101615
	471.4045208

⋮

⋮



P.378

- Presionando [EXE] mientras el resultado final del programa se encuentra sobre la presentación vuelve a ejecutar el programa.
- También puede ejecutar un programa mientras se encuentra en el modo **RUN** ingresando: Prog "<nombre de archivo>" [EXE].
- Se producirá un error si el programa especificado por Prog "<nombre de archivo>" no se encuentra.

20-3 Depurando un programa

Un problema en un programa que hace que el programa no funcione correctamente es lo que se llama defecto o “bug”, y el proceso de eliminar tales programas se llama depurado o “debugging”. Cualquiera de los síntomas siguientes indica que el programa contiene “bugs” y que se requiere de un depurado.

- Mensaje de error que aparece cuando se ejecuta el programa.
- Resultados que no están dentro de lo esperado.

●Para eliminar los defectos o “bugs” que ocasionan mensajes de error

Un mensaje de error, tal como el que se muestra a continuación, aparece siempre que ocurre algo ilegal durante la ejecución del programa.

```
Ma ERROR
```



P.436

Cuando aparece tal mensaje, presione ◀ o ▶ para visualizar la ubicación en donde se genera el error, junto con el cursor. Compruebe la “Tabla de mensajes de error”, para los pasos a tomar para corregir la situación.

P.360

- Tenga en cuenta que presionando ◀ o ▶ no visualizará la ubicación del error si el programa está protegido mediante una contraseña.

●Para eliminar los defectos o “bugs” que ocasionan malos resultados

Si el programa produce resultados que no se encuentran dentro de lo normalmente esperado, verifique los contenidos del programa y realice los cambios necesarios. Para los detalles en cómo cambiar los contenidos del programa, vea la sección “Editando nombres de archivo y contenidos de programa”.



P.365

20-4 Calculando el número de bytes usados por un programa

Existen dos tipos de mandos: mandos de 1 byte* y mandos de 2 bytes*.

* Un byte es una unidad de capacidad de memoria que puede usarse para el almacenamiento de datos.

- Ejemplos de mandos de 1 byte: sen, cos, tan, log, (,), A, B, C, 1, 2, etc.
- Ejemplos de mandos de 2 bytes: Lbl 1, Goto 2, etc.

Mientras el cursor se ubica dentro de un programa, a cada presión de las teclas ◀ o ▶ ocasiona que el cursor se mueva un byte.

- La cantidad de memoria que se está usando y la cantidad de memoria restante puede verificarse en cualquier momento, seleccionando el icono **MEM** en el menú principal e ingresando el modo MEM. Para los detalles vea la sección “Condición de la memoria (MEM)”.



20-5 Función de secreto

Cuando se ingresa un programa, se lo puede proteger con una contraseña que limita el acceso a los contenidos del programa, a solamente aquéllos que conocen la contraseña. Los programas que no están protegidos mediante una contraseña pueden ser ejecutados por cualquiera sin el ingreso de la contraseña.

●Para registrar una contraseña

Ejemplo Crear un archivo de programa bajo el nombre AREA y protegerlo con la contraseña CASIO.

1. Mientras la lista de programas se encuentra sobre la presentación, presione **F3** (NEW) e ingrese el nombre de archivo para el archivo de programa nuevo.

F3 (NEW)
A **R** **E** **A**

```
Program Name  
[AREA ]
```

2. Presione **F5** (**π0**) y luego ingrese la contraseña.

F5 (**π0**)
C **A** **S** **I** **O**

```
Program Name  
[AREA ]  
Password?  
[CASIO ]
```

- El procedimiento de ingreso de una contraseña es idéntico al usado para el ingreso de un nombre de archivo.
3. Presione **EXE** para registrar el nombre de archivo y contraseña. Ahora puede ingresar los contenidos del archivo de programa.
 - El registro de una contraseña utiliza 16 bytes de memoria.
 - Presionando **EXE** sin ingresar una contraseña registra solamente el nombre de archivo, sin una contraseña.
 4. Luego de ingresar el nombre, presione **SHIFT** **QUIT** para salir del archivo de programa y retornar a la lista de programas. Los archivos que están protegidos mediante contraseñas están indicados por un asterisco a la derecha del nombre de archivo.

```
Program List  
QUIT : 37  
AREA * : 33
```

●Para recuperar un programa

Ejemplo Recuperar el archivo llamado AREA que está protegido por la contraseña CASIO.

1. En la lista de programas, utilice las teclas **▲** y **▼** para mover la parte realzada en brillante al nombre del programa que desea recuperar.



P.353

2. Presione **F2** (EDIT).

```
Program Name  
[AREA ]  
Password?  
[ ]
```

3. Ingrese la contraseña y presione **EXE** para recuperar el programa.

- Si llega a ingresar una contraseña errónea aparecerá el mensaje **"Mismatch"**.

20-6 Buscando un archivo

Existen tres métodos diferentes para buscar un nombre de archivo especificado.

•Para encontrar un archivo usando la búsqueda por visualización

Ejemplo Utilizar la búsqueda por visualización para recuperar el programa llamado OCTA.

1. Mientras la lista de programas se encuentra sobre la presentación, utilice las teclas \blacktriangle y \blacktriangledown para ir visualizando a través de la lista de nombres de programa hasta encontrar el que desea.

Program List.	
OCTA	: 17
TRIANGLE	: 17
AREA	* : 33
GRAPHICS	: 17
MEASURE	: 17
OCTONARY	: 17
[EXE] [EDIT] [NEW] [DEL] [DEL] []	

[F2]

2. Cuando la parte destacada en brillante se ubica sobre el nombre del archivo que desea, presione [F2] (EDIT) para recuperarlo.

```

=====OCTA=====
2+A:2*J3*A^2.
J2+3*A^3
    
```

•Para encontrar un archivo usando la búsqueda por nombre de archivo

Ejemplo Utilizar la búsqueda por nombre de archivo para recuperar el programa llamado OCTA.

1. Mientras la lista de programas se encuentra sobre la presentación, presione [F3] (NEW) e ingrese el nombre del archivo que desea encontrar.
 - Si el archivo que está buscando está protegido mediante una contraseña, también debe ingresar la contraseña.

[F3] (NEW)

[O] [C] [T] [A]

Program Name
[OCTA] []

2. Presione [EXE] para recuperar el programa.
 - Si no hay ningún programa cuyo nombre de archivo coincida con lo que ha ingresado, se creará un archivo nuevo usando el nombre que ha ingresado.

•Para encontrar un archivo usando la búsqueda por carácter inicial

Ejemplo Utilizar la búsqueda por carácter inicial para recuperar el programa llamado OCTA.

1. Mientras la lista de programas se encuentra sobre la presentación, presione [F6] (\triangleright) [F1] (SRC) e ingrese los caracteres iniciales del archivo que desea encontrar.

[F6] (\triangleright) [F1] (SRC)

[O] [C] [T]

Search For Program
[OCTA] []



P.360

2. Presione **EXE** para iniciar la búsqueda.

```
Program List
OCTA      : 37
OCTONARY  : 17
```

- Se recuperarán todos los archivos cuyos nombres comiencen con los caracteres que ha ingresado.
 - Si no hay ningún programa cuyo nombre de archivo comience con los caracteres que ha ingresado, sobre la presentación aparecerá el mensaje **"Not Found"**. Si esto llega a suceder, presione **EXIT** para borrar el mensaje de error.
3. Utilice las teclas **▲** y **▼** para realzar en brillante el nombre de archivo del programa que desea recuperar, y luego presione **F2** (EDIT) para recuperarlo.

20-7 Buscando datos dentro de un programa

Ejemplo Buscar la letra "A" dentro del programa llamado OCTA.

1. Recupere el programa.
2. Presione **F3** (SRC) e ingrese los datos que desea buscar.

F3 (SRC)

ALPHA **A**

```
=====OCTA=====
2+A:2*√3*A²,
√2+3*A³
```

```
Search For Text
-----
A_
-----
SVB
```

- No se puede especificar el símbolo de línea nueva (↵) o mando de presentación (▲) para el dato buscado.
3. Presione **EXE** para comenzar la búsqueda. Los contenidos del programa aparecen sobre la pantalla con el cursor ubicado en la primera instancia del dato que ha especificado.

```
=====OCTA=====
?+A:2*√3*A²,
√2+3*A³
<Search> SVB
```

Indica que la operación de búsqueda se encuentra en progreso.

4. Presione **EXE** para buscar la siguiente instancia del dato.

```
=====OCTA=====
?+A:2*√3*A²,
√2+3*A³
```

- Si no hay partes coincidentes dentro del programa para el dato que ha especificado, los contenidos del programa aparecen con el cursor ubicado en el punto desde el cual ha comenzado la búsqueda.
- Una vez que los contenidos del programa se encuentran sobre la pantalla, puede usar las teclas de cursor para mover el cursor a otra posición antes de buscar la siguiente instancia del dato. Solamente la parte del programa que comienza desde la ubicación de cursor actual es buscada cuando se presiona **EXE**.
- Una vez que la búsqueda encuentra una instancia de su dato, ingresando caracteres o moviendo el cursor ocasiona que la operación de búsqueda se cancele (borrando el indicador de búsqueda desde la presentación).
- Si comete un error mientras ingresa caracteres que está buscando, presione **AC** para borrar su ingreso y reingresar desde el inicio.

20-8 Editando nombres de archivo y contenidos de programa

● Para editar un nombre de archivo

Ejemplo Cambiar el nombre de un archivo desde TRIANGLE a ANGLE.

1. Mientras la lista de programas se encuentra sobre la presentación, utilice las teclas \blacktriangle y \blacktriangledown para mover la parte destacada en brillante al archivo cuyo nombre desea editar, y luego presione $\boxed{F6}$ (\blacktriangleright) $\boxed{F2}$ (REN).

```

Rename
[ TRIANGLE ]
    
```

2. Realice los cambios que desea.

\boxed{DEL} \boxed{DEL} \boxed{DEL}

```

Rename
[ ANGLE ]
    
```

3. Presione \boxed{EXE} para registrar el nombre nuevo y retornar a la lista de programas.
 - Si las modificaciones que realiza resultan en un nombre de archivo que es idéntico al nombre de un programa que ya existe almacenado en la memoria, aparece el mensaje **"Already Exists"**. Cuando esto suceda, puede realizar cualquiera de las dos operaciones siguientes.
 - Presione \blacktriangleright o \blacktriangleleft para borrar el error y retornar a la pantalla de ingreso de nombre de archivo.
 - Presione \boxed{AC} para borrar el nombre de archivo nuevo e ingresar uno nuevo.

● Para editar los contenidos de un programa

1. Busque el nombre del programa que desea editar en la lista de programas.
2. Recupere el programa.
 - Los procedimientos que usa para la edición de los contenidos del programa son idénticos al usado para la edición de los cálculos manuales. Para los detalles, vea la sección "Edición de cálculos".
 - Las teclas de funciones siguientes son también prácticas cuando se editan los contenidos de un programa.

$\boxed{F1}$ (TOP) Mueve el cursor a la parte superior del programa.

```

=====OCTA=====
?+A: 2*sqrt(3)*A^2,
sqrt(2+3)*A^3
    
```

$\boxed{F2}$ (BTM) Mueve el cursor a la parte inferior del programa.

```

=====OCTA=====
?+A: 2*sqrt(3)*A^2,
sqrt(2+3)*A^3_
    
```



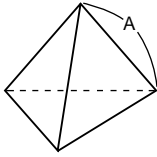
P.20



P.353

Ejemplo 2 Usar el programa OCTA para crear un programa que calcule el área de superficie y el volumen de tetraedros regulares, conociendo la longitud de un lado.

Utilice TETRA como el nombre del archivo.



Longitud de un lado (A)	Area de superficie (S)	Volumen (V)
7 cm	cm ²	cm ³
10 cm	cm ²	cm ³
15 cm	cm ²	cm ³

Las siguientes son las fórmulas usadas para el cálculo del área de superficie S y volumen V de tetraedro regular, conociendo la longitud de un lado.

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

Cuando ingrese el programa, utilice las siguientes operaciones de tecla.

Longitud de lado A **SHIFT** **PRGM** **F4** (?) **⇐** **ALPHA** **A** **F6** (▷) **F5** (:)
 Area de superficie S ... **SHIFT** **✓** **3** **X** **ALPHA** **A** **x²** **F6** (▷) **F5** (▲)
 Volumen V **SHIFT** **✓** **2** **÷** **1** **2** **X** **ALPHA** **A** **^** **3**

Compare esto con el programa para el cálculo del área de superficie y volumen de un octaedro regular.

Longitud de lado A **SHIFT** **PRGM** **F4** (?) **⇐** **ALPHA** **A** **F6** (▷) **F5** (:)
 Area de superficie S ... **2** **X** **SHIFT** **✓** **3** **X** **ALPHA** **A** **x²** **F6** (▷) **F5** (▲)
 Volumen V **SHIFT** **✓** **2** **÷** **3** **X** **ALPHA** **A** **^** **3**

Como puede ver, puede producir el programa TETRA realizando los cambios siguientes en el programa OCTA.

- Borrando **2** **X** (subrayado usando una línea ondulada)
- Cambiando **3** a **1** **2** (subrayado usando una línea sólida)

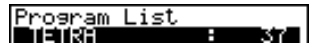
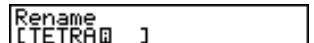
Editemos OCTA para producir el programa TETRA.

1. Edite el nombre del programa.

F6 (▷) **F2** (REN) **T** **E** **T** **R** **A**



EXE

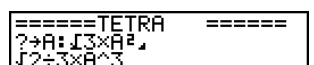


2. Edite los contenidos del programa.

F2 (EDIT)



▶▶▶▶ **DEL** **DEL**



◀ ◁ SHIFT INS 1 2

```
=====TETRA =====
?→A:√3×A²,
√2+12√3×A³
```

DEL

```
=====TETRA =====
?→A:√3×A²,
√2+12×A³
```

SHIFT QUIT

Tratemos de ejecutar el programa.

Longitud de un lado (A)	Area de superficie (S)	Volumen (V)
7 cm	84,87048957 cm ²	40,42293766 cm ³
10 cm	173,2050808 cm ²	117,8511302 cm ³
15 cm	389,7114317 cm ²	397,7475644 cm ³

F1 (EXE) o EXE

```
?
?
```

7 EXE

(Valor de A)

```
?
7
84.87048957
- DISP -
```

EXE EXE

```
?
7
84.87048957
40.42293766
?
```

1 0 EXE

```
?
84.87048957
40.42293766
?
10
173.2050808
- DISP -
```

EXE

```
?
84.87048957
40.42293766
?
10
173.2050808
117.8511302
```



⋮

⋮

20-9 Borrando programas

Existen dos métodos para borrar un nombre de archivo y su programa.

●Para borrar un programa específico

1. Mientras la lista de programas se encuentra sobre la presentación, utilice las teclas  y  para mover la parte destacada en brillante al nombre del programa que desea borrar.
2. Presione **F4** (DEL).
3. Presione **F1** (YES) para borrar el programa seleccionado o **F6** (NO) para cancelar la operación sin borrar nada.

●Para borrar todos los programas

1. Mientras la lista de programas se encuentra sobre la presentación, presione **F5** (DEL.A).
2. Presione **F1** (YES) para borrar todos los programas de la lista o **F6** (NO) para cancelar la operación sin borrar nada.
 - También puede borrar todos los programas usando el modo **MEM**. Para los detalles vea la sección “Borrando los contenidos de la memoria”.



P.26

20-10 Mandos de programa prácticos

Además de los mandos de cálculo, esta calculadora también incluye una variedad de mandos de relación y de salto, que pueden usarse para crear en forma rápida y fácil programas que realizan cálculos repetidos.

Menú de programa

Para visualizar el menú de programa presione **SHIFT** **PRGM**.

- **{COM}/{CTL}/{JUMP}/{CLR}/{DISP}/{REL}/{I/O}**
- **{?}** ... {mando de entrada}
- **{▲}** ... {mando de salida}
- **{:}** ... {mando de instrucción múltiple}

■ COM (menú de mando de programa)

Seleccionando {COM} desde el menú de programa visualiza los ítemes de menú de función siguiente.

- **{If}/{Then}/{Else}/{I-End}/{For}/{To}/{Step}/{Next}/{While}/{WEnd}/{Do}/{Lp-W}**
... mando {If}/{Then}/{Else}/{IfEnd}/{For}/{To}/{Step}/{Next}/{While}/
{WhileEnd}/{Do}/{LpWhile}

■ CTL (menú de mando de control de programa)

Seleccionando {CTL} desde el menú de programa visualiza los ítemes de menú de función siguiente.

- **{Prog}/{Rtrn}/{Brk}/{Stop}** ... mando {Prog}/{Return}/{Break}/{Stop}

■ JUMP (menú de mando de salto)

Seleccionando {JUMP} desde el menú de programa visualiza los ítemes de menú de función siguientes.

- **{Lbl}/{Goto}** ... mando {Lbl}/{Goto}
- **{⇒}** ... {mando jump}
- **{Isz}/{Dsz}** ... {salto e incremento}/{salto y decremento}

■ CLR (menú de mando de borrado)

Seleccionando {CLR} desde el menú de programa visualiza los ítemes de menú de función siguientes.

- **{Text}/{Grph}/{List}** ... borra el {texto}/{gráfico}/{lista}

■ DISP (menú de mando de presentación)

Seleccionando {DISP} desde el menú de programa de presentación visualiza los ítemes de menú de función siguientes.

- {Stat}/{Grph}/{Dyna} ... delineado de {gráfico estadístico}/{gráfico}/
{gráfico dinámico}
- {F-Tbl} ... {Menú de mando de gráfico y tabla}
Los siguientes son los ítemes que aparecen en el menú anterior.
 - {Tbl}/{G-Con}/{G-Plt} ... mando {DispF-Tbl}/{DrawFTG-Con}/
{DrawFTG-Plt}
- {R-Tbl} ... {fórmula de recurrencia y cálculo de recurrencia}
Los siguientes son los ítemes que aparecen en el menú anterior.
 - {Tbl}/{Web}/{an-Cn}/{Σa-Cn}/{an-Pl}/{Σa-Pl} ... mando {DispR-Tbl}/
{DrawWeb}/{DrawR-Con}/{DrawRΣ-Con}/{DrawR-Plt}/{DrawRΣ-Plt}

■ REL (mandos de operador de relación de salto condicional)

Seleccionando {REL} desde el menú de programa visualiza los siguientes ítemes de menú de función.

- {=}/{≠}/{>}/{<}/{≥}/{≤} ... operadores de relación {=}/{≠}/{>}/{<}/{≥}/{≤}

■ I/O (mandos de entrada y salida)

Seleccionando {I/O} desde el menú de programa visualiza los ítemes de menú de función siguientes.

- {Lcte}/{Gtky}/{Send}/{Recv} ... mando {Locate}/{Getkey}/{Send()}/{Receive()}
- La apariencia del menú de funciones difiere ligeramente para un programa que contiene cálculos con valores binarios, octales, decimales o hexadecimales, pero las funciones en el menú son las mismas.

20-11 Referencia de mandos

■ Indice de mandos

Break	378
ClrGraph	382
ClrList	382
ClrText	382
DispF-Tbl, DispR-Tbl	383
Do~LpWhile	377
DrawDyna	383
DrawFTG-Con, DrawFTG-Plt	383
DrawGraph	383
DrawR-Con, DrawR-Plt	384
DrawRΣ-Con, DrawRΣ-Plt	384
DrawStat	384
DrawWeb	384
Dsz	380
For~To~Next	375
For~To~Step~Next	376
Getkey	385
Goto~Lbl	380
If~Then	373
If~Then~Else	374
If~Then~Else~IfEnd	375
If~Then~IfEnd	374
Isz	381
Locate	385
Prog	378
Receive (.....	386
Return	379
Send (.....	387
Stop	379
While~WhileEnd	377
? (Mando de ingreso)	372
▲ (Mando de salida)	372
: (Mando de instrucción múltiple)	373
↵ (Retorno de carro)	373
⇒ (Código de salto)	381
=, ≠, >, <, ≥, ≤ (Operador de relación)	387

Las siguientes son las convenciones usadas en esta sección cuando se describen los diferentes y variados mandos.

- Texto en negrita Los mandos reales y otros ítems que siempre deben ser ingresados se muestran en negrita.
- {Llaves} Las llaves se usan para encerrar un número de ítems, uno de los cuales debe ser seleccionado cuando se usa un mando. No ingrese las llaves cuando ingresa un mando.
- [Corchetes] Los corchetes se usan para encerrar ítems que son opcionales. No ingrese los corchetes cuando ingrese un mando.
- Expresiones numéricas ... Las expresiones numéricas (tales como 10, 10 + 20, A) indican constantes, cálculos, constantes numéricas, etc.
- Caracteres alfabéticos Los caracteres alfabéticos indican series literales (tales como AB).

■ Mandos de operación básicos

? (Mando de ingreso)

Función: Indica solicitando el ingreso de valores para la asignación a las variables durante la ejecución de un programa.

Sintaxis: ? → <nombre de variable>

Ejemplo: ? → A ↵

Descripción:

1. Este mando interrumpe momentáneamente la ejecución del programa e indica solicitando el ingreso de un valor o expresión para la asignación a una variable. Cuando se ejecuta el mando de ingreso, aparece "?" sobre la presentación y la calculadora permanece en espera para el ingreso.
2. El ingreso en respuesta al mando de ingreso debe ser un valor o una expresión, y la expresión no puede ser una instrucción múltiple.

▲ (Mando de salida)

Función: Visualiza resultados intermedios durante la ejecución de un programa.

Descripción:

1. Este mando interrumpe momentáneamente la ejecución de un programa y visualiza el texto de caracteres alfabéticos o el resultado de un cálculo inmediatamente anterior.
2. El mando de salida debe usarse en posiciones en donde debería normalmente presionar la tecla **EXE** durante un cálculo manual.

: (Mando de instrucción múltiple)

Función: Conecta dos instrucciones para una ejecución secuencial sin parar.

Descripción:

1. Diferente al mando de salida (▲), las instrucciones conectadas con el mando de instrucción múltiple se ejecutan sin parar.
2. El mando de instrucción múltiple puede usarse para enlazar dos expresiones de cálculo o dos mandos.
3. También puede usar un retorno de carro indicado por ↵ en lugar del mando de instrucción múltiple.

↵ (Retorno de carro)

Función: Conecta dos instrucciones para una ejecución secuencial sin parar.

Descripción:

1. La operación del retorno de carro es idéntica al del mando de instrucción múltiple.
2. Usando un retorno de carro en lugar de un mando de instrucción múltiple hace que la presentación sea más fácil de leer.

■ Mandos de programa (COM)**If~Then**

Función: La instrucción "Then" solamente se ejecuta cuando la condición "If" es verdadera (diferente de cero).

Sintaxis:

$$\text{If } \begin{array}{l} \text{<condición>} \\ \text{expresión numérica} \end{array} \left\{ \begin{array}{c} \text{↵} \\ \vdots \\ \text{▲} \end{array} \right\} \text{ Then } \text{<instrucción>} \left[\left\{ \begin{array}{c} \text{↵} \\ \vdots \\ \text{▲} \end{array} \right\} \text{ <instrucción>} \right]$$

Parámetros: condición, expresión numérica

Descripción:

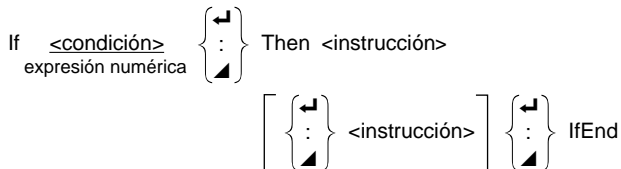
1. La instrucción "Then" solamente se ejecuta cuando la condición "If" es verdadera (diferente de cero).
2. Si la condición es falsa (cero), la instrucción "Then" no se ejecuta.
3. Una condición "If" siempre debe ser acompañada por una instrucción "Then". Omitiendo la instrucción "Then" resulta en un error.

Ejemplo: If A = 0 ↵
Then "A = 0"

If~Then~IfEnd

Función: La instrucción "Then" se ejecuta solamente cuando la condición "If" es verdadera (diferente de cero). La instrucción "IfEnd" se ejecuta siempre: luego de ejecutarse la instrucción "Then" o directamente después de la condición "If" cuando la condición "If" es falsa (cero).

Sintaxis:



Parámetros: condición, expresión numérica

Descripción:

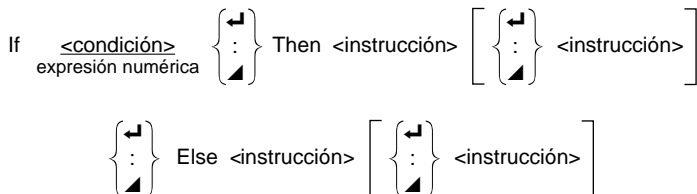
Este mando es casi idéntico a "If~Then". La única diferencia es que la instrucción IfEnd se ejecuta siempre, sin tener en cuenta si la condición "If" es verdadera (diferente de cero) o falsa (cero).

Ejemplo: If A = 0 ↙
 Then "A = 0" ↙
 IfEnd ↙
 "END"

If-Then-Else

Función: La instrucción "Then" se ejecuta solamente cuando la condición "If" es verdadera (diferente de cero). La instrucción "Else" se ejecuta cuando la condición "If" es falsa (cero).

Sintaxis:



Parámetros: condición, expresión numérica

Descripción:

1. La instrucción "Then" se ejecuta cuando las condiciones "If" son verdaderas (diferente de cero).
2. La instrucción "Else" se ejecuta cuando las condiciones "If" son falsas (cero).

Ejemplo: If A = 0 ↙
 Then "TRUE" ↙
 Else "FALSE"

If~Then~Else~IfEnd

Funci3n: La instrucci3n "Then" se ejecuta solamente cuando la condici3n "If" es verdadera (diferente de cero). La instrucci3n "Else" se ejecuta cuando la condici3n "If" es falsa (cero). La instrucci3n "IfEnd" se ejecuta siempre siguiendo a la instrucci3n "Then" o instrucci3n "Else".

Sintaxis:

$$\text{If } \begin{array}{l} \langle \text{condici3n} \rangle \\ \text{expresi3n num3rica} \end{array} \left\{ \begin{array}{l} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\} \text{Then } \langle \text{instrucci3n} \rangle \left[\left\{ \begin{array}{l} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\} \langle \text{instrucci3n} \rangle \right]$$

$$\left\{ \begin{array}{l} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\} \text{Else } \langle \text{instrucci3n} \rangle \left[\left\{ \begin{array}{l} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\} \langle \text{instrucci3n} \rangle \right] \left\{ \begin{array}{l} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\} \text{IfEnd}$$

Par3metros: condici3n, expresi3n num3rica

Descripci3n:

Este mando es casi id3ntico a "If~Then~Else". La 3nica diferencia es que la instrucci3n IfEnd se ejecuta siempre, sin tener en cuenta si la condici3n "If" es verdadera (diferente de cero) o falsa (cero).

Ejemplo: ? → A ↵
 If A = 0 ↵
 Then "TRUE" ↵
 Else "FALSE" ↵
 IfEnd ↵
 "END"

For~To~Next

Funci3n: Este mando repite todo lo que hay entre la instrucci3n "For" y la instrucci3n "Next". El valor inicial se asigna a la variable de control con la primera ejecuci3n, y el valor de la variable de control es incrementado de uno con cada ejecuci3n. La ejecuci3n contin3a hasta que el valor de la variable de control excede el valor final.

Sintaxis:

$$\text{For } \langle \text{valor inicial} \rangle \rightarrow \langle \text{nombre de variable de control} \rangle \text{ To } \langle \text{valor final} \rangle \left\{ \begin{array}{l} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\}$$

$$\left[\langle \text{instrucci3n} \rangle \left\{ \begin{array}{l} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\} \right] \text{Next}$$

Parámetros:

- Nombre de variable de control: A hasta la Z
- Valor inicial: valor o expresión que produzca un valor (es decir $\text{sen } x$, A, etc.)
- Valor final: valor o expresión que produzca un valor (es decir $\text{sen } x$, A, etc.)

Descripción:

1. Cuando el valor de la variable de control es mayor que el valor final, sin ejecutar las instrucciones entre “For” y “Next”, y luego la ejecución continúa desde la instrucción siguiente Next.
2. Una instrucción “For” debe tener siempre una instrucción “Next” correspondiente, y la instrucción “Next” debe venir siempre luego de su instrucción correspondiente “For”.
3. La instrucción “Next” define el final del bucle o ciclo creado por “For~Next”, y de esta manera debe estar incluida siempre. De lo contrario resultará en un error.

Ejemplo: For 1 → A To 10 ↵
 A × 3 → B ↵
 B ▲
 Next

For~To~Step~Next

Función: Este mando repite todo lo que hay entre la instrucción “For” y la instrucción “Next”. El valor inicial se asigna a la variable de control con la primera ejecución, y el valor de la variable de control es cambiado de acuerdo al valor de intervalo con cada ejecución. La ejecución continúa hasta que el valor de la variable de control excede el valor final.

Sintaxis:

$$\text{For} \langle \text{valor inicial} \rangle \rightarrow \langle \text{nombre de variable de control} \rangle \text{ To } \langle \text{valor final} \rangle \text{ Step } \langle \text{valor de intervalo} \rangle \left\{ \begin{array}{l} \text{↵} \\ \text{:} \\ \text{▲} \end{array} \right\}$$

Next

Parámetros:

- Nombre de variable de control: A hasta la Z
- Valor inicial: valor o expresión que produzca un valor (es decir $\text{sen } x$, A, etc.)
- Valor final: valor o expresión que produzca un valor (es decir $\text{sen } x$, A, etc.)
- Valor de intervalo: valor numérico (omitiendo este valor el intervalo se ajusta a 1).

Descripción:

1. Este mando es básicamente idéntico a “For~To~Next”. La única diferencia es que puede especificarse el intervalo.
2. Omitiendo el valor del intervalo ajusta automáticamente el intervalo a 1.

3. Haciendo que el valor inicial sea menor que el valor final, y especificando un valor de intervalo positivo ocasiona que la variable de control sea incrementada con cada ejecución. Haciendo que el valor inicial sea mayor que el valor final, y especificando un valor de intervalo negativo ocasiona que la variable de control sea disminuida con cada ejecución.

Ejemplo: For 1 → A To 10 Step 0.1 ↵
 A × 3 → B ↵
 B ▲
 Next

Do~LpWhile

Función: Este mando repite mandos específicos en tanto su condición es verdadera (diferente de cero).

Sintaxis:

$$\text{Do } \left\{ \begin{array}{c} \text{↵} \\ \text{:} \\ \text{▲} \end{array} \right\} \sim \text{LpWhile } \langle \text{expresión} \rangle$$

Parámetros: expresión

Descripción:

1. Este mando repite los mandos contenidos en el bucle en tanto su condición es verdadera (diferente de cero). Cuando la condición se convierte en falsa (cero), la ejecución procede desde la instrucción siguiendo la instrucción LpWhile.
2. Como la condición viene después de la instrucción LpWhile, la condición es comprobada (verificada) luego de que todos los mandos dentro del bucle son ejecutados.

Ejemplo: Do ↵
 ? → A ↵
 A × 2 → B ↵
 B ▲
 LpWhile B > 10

While~WhileEnd

Función: Este mando repite los mandos específicos en tanto su condición sea verdadera (diferente de cero).

Sintaxis:

$$\text{While } \langle \text{expresión} \rangle \left\{ \begin{array}{c} \text{↵} \\ \text{:} \\ \text{▲} \end{array} \right\} \sim \text{WhileEnd}$$

Parámetros: expresión

Descripción:

1. Este mando repite los mandos contenidos en el bucle en tanto su condición es verdadera (diferente de cero). Cuando la condición se convierte en falsa (cero), la ejecución procede desde la instrucción siguiendo la instrucción WhileEnd.

- Como la condición viene después de la instrucción While, la condición es comprobada (verificada) luego de que los mandos dentro del bucle son ejecutados.

Ejemplo: 10 → A ↵
 While A > 0 ↵
 A - 1 → A ↵
 "GOOD" ↵
 WhileEnd

■ Mandos de control de programa (CTL)

Break

Función: Este mando rompe la ejecución de un bucle y continúa desde el mando siguiente siguiendo el bucle o ciclo.

Sintaxis: Break ↵

Descripción:

- Este mando rompe la ejecución de un bucle y continúa desde el mando siguiendo el bucle.
- Este mando puede usarse para romper la ejecución de una instrucción "For", instrucción "Do" e instrucción "While".

Ejemplo: While A>0 ↵
 If A > 2 ↵
 Then Break ↵
 IfEnd ↵
 WhileEnd ↵
 A ▲ ←———— Se ejecuta después de la ruptura

Prog

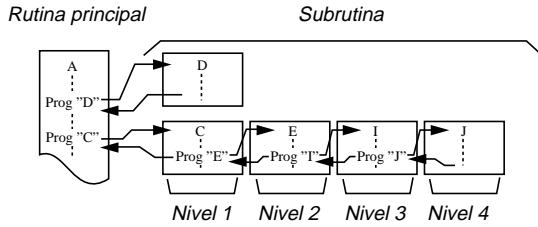
Función: Este mando especifica la ejecución de otro programa como una subrutina. En el modo RUN, este mando ejecuta un programa nuevo.

Sintaxis: Prog "nombre de archivo" ↵

Ejemplo: Prog "ABC" ↵

Descripción:

- Aun cuando este programa se ubica dentro de un bucle, su ejecución rompe inmediatamente el bucle y ejecuta una subrutina.
- Este mando puede usarse tantas veces como sea necesario dentro de una rutina principal, para llamar a subrutinas independientes para que realicen tareas específicas.
- Una subrutina puede usarse en múltiples ubicaciones en la misma rutina principal, o puede ser llamada por cualquier número de rutinas principales.



4. Llamando una subrutina ocasiona que sea ejecutada desde el comienzo. Luego de que la ejecución de una subrutina se completa, la ejecución retorna a la rutina principal, continuando desde la instrucción siguiente al mando Prog.
5. Un mando "Goto-Lbl" dentro de una subrutina es válida solamente dentro de esa subrutina. Si no puede usarse para saltar a un rótulo fuera de la subrutina.
6. Si una subrutina con el nombre de archivo especificado por el mando Prog no existe, se genera un error.
7. En el modo **RUN**, ingresando el mando Prog y presionando **EXE** ejecuta el programa especificado por el mando.

Return

Función: Este mando realiza un retorno desde una subrutina.

Sintaxis: Return ↵

Descripción:

La ejecución de un mando "Return" dentro de una rutina principal ocasiona que la ejecución de un programa se pare.

Ejemplo:

Prog "A"	Prog "B"
1 → A ↵	For A → B To 10 ↵
Prog "B" ↵	B + 1 → C ↵
C ▲	Next ↵
	Return

La ejecución del programa en el archivo A visualiza el resultado de la operación (11).

Stop

Función: Este mando finaliza la ejecución de un programa.

Sintaxis: Stop ↵

Descripción:

1. Este mando finaliza la ejecución de un programa.
2. La ejecución de este mando dentro de un bucle finaliza la ejecución sin que se genere un mensaje de error.

Ejemplo: For 2 → 1 To 10 ↵
 If 1 = 5 ↵
 Then "STOP" : Stop ↵
 IfEnd ↵
 Next

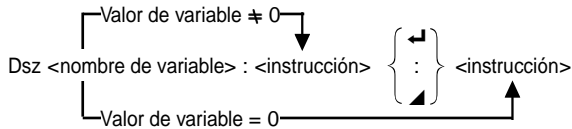
Este programa cuenta de 2 a 10. Cuando la cuenta alcanza a 5, no obstante, finaliza la ejecución y visualiza el mensaje "STOP".

■ Mandos de saltos (JUMP)

Dsz

Función: Este mando es un salto de cuenta que disminuye el valor de una variable de control en 1, y luego salta si el valor actual de la variable es cero.

Sintaxis:



Parámetros:

Nombre de variable: A hasta la Z, r, θ

[Ejemplo] Dsz B : Disminuye el valor asignado a la variable B en 1.

Descripción:

Este mando disminuye el valor de una variable de control en 1, y luego lo comprueba (verifica). Si el valor actual no es cero, la ejecución continúa con la instrucción siguiente. Si el valor actual es cero, la ejecución salta a la instrucción siguiendo el mando instrucción múltiple (:), mando de presentación (▲), o retorno de carro (↵).

Ejemplo: 10 → A : 0 → C :
 Lbl 1 : ? → B : B+C → C :
 Dsz A : Goto 1 : C ÷ 10

Este programa indica solicitando el ingreso de 10 valores, y luego calcula el promedio de los valores ingresados.

Goto~Lbl

Función: Este mando realiza un salto incondicional a una ubicación especificada.

Sintaxis: Goto <valor o variable> ~ Lbl <valor o variable>

Parámetros: Valor (de 0 a 9), variable (A hasta la Z, r, θ)

Descripción:

1. Este mando consiste de dos partes: Goto *n* (en donde *n* es un valor de 0 al 9) y Lbl *n* (en donde *n* es el valor especificado para Goto). Este mando ocasiona que la ejecución del programa salte a la instrucción "Lbl" cuyo valor coincide con el valor especificado por la instrucción "Goto".

2. Este mando puede usarse para hacer un bucle de nuevo al comienzo de un programa o para saltar a cualquier ubicación dentro del programa.
3. Este mando puede usarse en combinación con saltos condicionales y saltos de cuenta.
4. Si no hay ninguna instrucción "Lbl" cuyo valor coincida con el valor especificado por la instrucción "Goto", se generará un error.

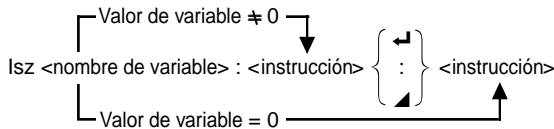
Ejemplo: ? → A : ? → B : Lbl 1 :
 ? → X : A × X + B ▲
 Goto 1

Este programa calcula $y = AX + B$ para tantos valores como para cada variable que desee ingresar. Para salir de la ejecución de este programa, presione **[AC]**.

Isz

Función: Este mando es un salto de cuenta que aumenta el valor de una variable de control en 1, y luego salta si el valor actual de la variable es cero.

Sintaxis:



Parámetros:

Nombre de variable: A hasta la Z, r, θ

[Ejemplo] Isz A : Aumenta el valor asignado a la variable A en 1.

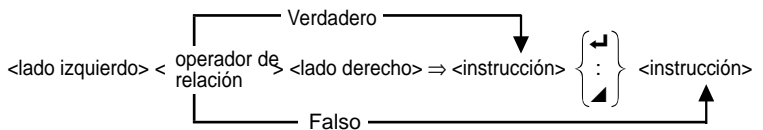
Descripción:

Este mando aumenta el valor de una variable de control en 1, y luego lo comprueba (verifica). Si el valor actual no es cero, la ejecución continúa con la instrucción siguiente. Si el valor actual es cero, la ejecución salta a la instrucción siguiendo al mando de instrucción múltiple (:), mando de presentación (▲), o retorno de carro (↵).

⇒ (Código de salto)

Función: Este código se usa para ajustar las condiciones para un salto condicional. El salto se ejecuta siempre que las condiciones son falsas.

Sintaxis:





P.387

Parámetros:

lado izquierdo/lado derecho: variable (A hasta la Z, r, θ), constante numérica, expresión de variable (tal como: $A \times 2$)

operador de relación: =, \neq , >, <, \geq , \leq

Descripción:

1. El salto condicional compara los contenidos de dos variables o los resultados de dos expresiones, y se realiza una decisión entre si se ejecuta o no el salto basado en los resultados de la comparación.
2. Si la comparación retorna un resultado verdadero, la ejecución continúa con la instrucción siguiendo el mando \Rightarrow . Si la comparación retorna un resultado falso, la ejecución salta a la instrucción siguiendo el mando de instrucción múltiple (:), mando de presentación (\blacktriangleleft), o retorno de carro (\blacktriangleleft).

Ejemplo: Lbl 1 :? \rightarrow A :

$A \geq 0 \Rightarrow \sqrt{A} \blacktriangleleft$

Goto 1

Con este programa, ingresando un valor de cero o mayor calcula y visualiza la raíz cuadrada del valor ingresado. Ingresando un valor menor de cero retorna al indicador de solicitud de ingreso sin calcular nada.

■ Mandos de borrado (CLR)

ClrGraph

Función: Este mando borra la pantalla gráfica.

Sintaxis: ClrGraph \blacktriangleleft

Descripción: Este mando borra la pantalla de gráfico durante la ejecución de un programa.

ClrList

Función: Este mando borra los datos de lista.

Sintaxis: ClrList \blacktriangleleft

Descripción: Este mando borra los contenidos de la lista actualmente seleccionada (Lista 1 a Lista 6) durante la ejecución de un programa.

ClrText

Función: Este mando borra la pantalla de texto.

Sintaxis: ClrText \blacktriangleleft

Descripción: Este mando borra el texto que hay en la pantalla durante la ejecución de un programa.

■ Mandos de presentación (DISP)**DispF-Tbl, DispR-Tbl**

Función: Estos mandos visualizan tablas numéricas.

Sintaxis:

DispF-Tbl ↵

DispR-Tbl ↵

Descripción:

1. Estos mandos generan tablas numéricas durante la ejecución de un programa de acuerdo con las condiciones definidas dentro del programa.
2. DispF-Tbl genera una tabla de funciones, mientras DispR-Tbl genera una tabla de recurrencias.

DrawDyna

Función: Este mando ejecuta una operación de delineado de gráfico dinámico.

Sintaxis: DrawDyna ↵

Descripción: Este mando ejecuta una operación de delineado de gráfico dinámico durante la ejecución de un programa, de acuerdo con las condiciones de delineado definidas dentro del programa.

DrawFTG-Con, DrawFTG-Plt

Función: Estos mandos grafican funciones.

Sintaxis:

DrawFTG-Con ↵

DrawFTG-Plt ↵

Description:

1. Estos mandos grafican funciones de acuerdo con las condiciones definidas dentro del programa.
2. DrawFTG-Con produce un gráfico de tipo conectado, mientras DrawFTG-Plt produce un gráfico de tipo de puntos marcados.

DrawGraph

Función: Este mando dibuja un gráfico.

Sintaxis: DrawGraph ↵

Descripción: Este mando traza un gráfico de acuerdo con las condiciones de delineado definidas dentro del programa.

DrawR-Con, DrawR-Plt

Función: Estos mandos grafican expresiones de recurrencia, con $a_n(b_n)$ como el eje vertical y n como el eje horizontal.

Sintaxis:

DrawR-Con ↵

DrawR-Plt ↵

Descripción:

1. Estos mandos grafican expresiones de recurrencia, con $a_n(b_n)$ como el eje vertical y n como el eje horizontal, de acuerdo con las condiciones definidas dentro del programa.
2. DrawR-Con produce un gráfico de tipo conectado, mientras DrawR-Plt produce un gráfico de tipo de puntos marcados.

DrawRΣ-Con, DrawRΣ-Plt

Función: Estos mandos grafican expresiones de recurrencia, con $\Sigma a_n(\Sigma b_n)$ como el eje vertical y n como el eje horizontal.

Sintaxis:

DrawRΣ-Con ↵

DrawRΣ-Plt ↵

Descripción:

1. Estos mandos grafican expresiones de recurrencia, con $\Sigma a_n(\Sigma b_n)$ como el eje vertical y n como el eje horizontal, de acuerdo con las condiciones definidas dentro del programa.
2. DrawRΣ-Con produce un gráfico de tipo conectado, mientras DrawRΣ-Plt produce un gráfico de tipo de puntos marcados.

DrawStat

Función: Este mando traza un gráfico estadístico.

Sintaxis:

DrawStat ↵

Descripción:

Este mando traza un gráfico estadístico de acuerdo con las condiciones definidas dentro del programa.

DrawWeb

Función: Este mando grafica convergencias/divergencias de una expresión de recurrencia (WEB Graph).

Sintaxis: DrawWeb [nombre de la expresión de recurrencia], [número de líneas] ↵

Ejemplo: DrawWeb $a_{n+1}(b_{n+1})$, 5 ↵

Descripción:

1. Este mando grafica convergencias/divergencias de una expresión de recurrencia (WEB Graph).
2. Omitiendo la especificación del número de líneas automáticamente se especifica el valor por omisión 30.

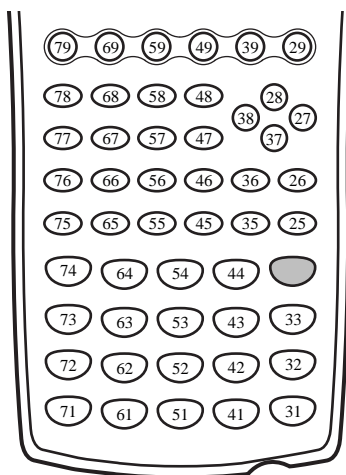
■ Mandos de entrada/salida (I/O)**Getkey**

Función: Este mando retorna el código que corresponde a la última tecla presionada.

Sintaxis: Getkey ↵

Descripción:

1. Este mando retorna el código que corresponde a la última tecla presionada.



2. Si no se presiona ninguna tecla previo a la ejecución de este mando, se retorna un cero.
3. Este mando puede usarse dentro de un bucle.

Locate

Función: Este mando visualiza los caracteres alfanuméricos en una posición especificada sobre la pantalla de texto.

Sintaxis:

Locate <número de columna>, <número de línea>, <valor>

Locate <número de columna>, <número de línea>, <nombre de variable>

Locate <número de columna>, <número de línea>, "<serie>"

[Ejemplo] Locate 1, 1, "AB" ↵

Parámetros:

- Número de línea: números desde 1 al 7.
- Número de columna: números desde 1 al 21.
- Valor: valores numéricos.
- Nombre de variable: A hasta la Z.
- Serie: serie de caracteres.

Descripción:

1. Este mando visualiza valores (incluyendo contenidos de variables) o texto en una posición especificada sobre la pantalla de texto.
2. La línea está designada por un valor de 1 al 7, mientras la columna se designa por un valor de 1 al 21.

**Ejemplo:** Cls ↵

Locate 7, 1, "CASIO CFX"

Este programa visualiza el texto "CASIO CFX" en el centro de la pantalla.

- En algunos casos, el mando ClrText debe ejecutarse antes de ejecutar el programa anterior

Receive (

Función: Este mando recibe datos desde un dispositivo externo.

Sintaxis: Receive (<dato>)

Descripción:

1. Este mando recibe datos desde un dispositivo externo.
2. Los siguientes tipos de mandos pueden ser recibidos por este mando.
 - Valores individuales asignados a variables.
 - Datos de matrices (todos los valores – los valores individuales no pueden especificarse).
 - Lista de datos (todos los valores – los valores individuales no pueden especificarse).
 - Datos de imagen.

Send (

Función: Este mando transmite los datos a un dispositivo externo.

Sintaxis: Send (<dato>)

Descripción:

1. Este mando transmite los datos a un dispositivo externo.
2. Los siguientes tipos de mandos pueden ser enviados por este mando.
 - Valores individuales asignados a variables.
 - Datos de matrices (todos los valores – los valores individuales no pueden especificarse).
 - Lista de datos (todos los valores – los valores individuales no pueden especificarse).

■ Operadores de relación de salto condicional (REL)

=, ≠, >, <, ≥, ≤

Función: Estos operadores de relación se usan en combinación con el mando de salto condicional.

Sintaxis:

$$\langle \text{lado izquierdo} \rangle \langle \text{operador de relación} \rangle \langle \text{lado derecho} \rangle \Rightarrow \langle \text{instrucción} \rangle \left. \begin{array}{c} \lrcorner \\ : \\ \llcorner \end{array} \right\} \langle \text{instrucción} \rangle$$

Parámetros:

lado izquierdo/lado derecho: variable (A hasta la Z, r, θ), constante numérica, expresión de variable (tales como: $A \times 2$)

Operador de relación: =, ≠, >, <, ≥, ≤

Descripción:

1. En el mando de salto condicional se pueden usar los siguientes seis operadores de relación:

$\langle \text{lado izquierdo} \rangle = \langle \text{lado derecho} \rangle$: verdadero cuando $\langle \text{lado izquierdo} \rangle$ igual $\langle \text{lado derecho} \rangle$

$\langle \text{lado izquierdo} \rangle \neq \langle \text{lado derecho} \rangle$: verdadero cuando $\langle \text{lado izquierdo} \rangle$ no es igual $\langle \text{lado derecho} \rangle$

$\langle \text{lado izquierdo} \rangle > \langle \text{lado derecho} \rangle$: verdadero cuando $\langle \text{lado izquierdo} \rangle$ es mayor $\langle \text{lado derecho} \rangle$

$\langle \text{lado izquierdo} \rangle < \langle \text{lado derecho} \rangle$: verdadero cuando $\langle \text{lado izquierdo} \rangle$ es menor $\langle \text{lado derecho} \rangle$

$\langle \text{lado izquierdo} \rangle \geq \langle \text{lado derecho} \rangle$: verdadero cuando $\langle \text{lado izquierdo} \rangle$ es mayor o igual $\langle \text{lado derecho} \rangle$

$\langle \text{lado izquierdo} \rangle \leq \langle \text{lado derecho} \rangle$: verdadero cuando $\langle \text{lado izquierdo} \rangle$ es menor o igual $\langle \text{lado derecho} \rangle$

2. Para los detalles acerca del uso del salto condicional, vea \Rightarrow ("Código de salto").

20-12 Presentación de texto

Se puede incluir un texto dentro de un programa encerrándolo simplemente entre comillas. Tal texto aparecerá sobre la presentación durante la ejecución de un programa, lo cual significa que puede agregar rótulos para ingresar indicadores y resultados.

Programa	Presentación
? → X	?
"X =" ? → X	X = ?

- Si el texto es seguido por una fórmula de cálculo, asegúrese de insertar un mando de presentación (▲), un retorno de carro (↵) o mando de instrucción múltiple (:) entre el texto y el cálculo.
- Ingresando más de 21 caracteres ocasiona que el texto se mueva hacia abajo a la línea siguiente. La pantalla comienza a pasar el texto visualizado automáticamente, si el texto ocasiona que la pantalla se llene por completo.

20-13 Usando las funciones de la calculadora dentro de los programas



P.80

■ Usando las operaciones de fila de matriz dentro de un programa

Estos mandos le permiten manipular filas de una matriz en un programa.

- Para este tipo de programa, asegúrese de usar el modo **MAT** para ingresar la matriz, y luego cambiar al modo **PRGM** para ingresar el programa.

● Para transponer los contenidos de dos filas (Swap)

Ejemplo 1 Transponer los valores de la fila 2 y fila 3 de la matriz siguiente:

$$\text{Matriz A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

La siguiente es la sintaxis para este programa.

Swap A, 2, 3

Nombre de matriz

Ejecutando este programa produce el resultado siguiente.

(Modo MAT)

	1	2
1	1	2
2	5	6
3	3	4

● Para calcular una multiplicación escalar (*ROW)

Ejemplo 2 Calcular la multiplicación escalar de la fila 2 de la matriz del Ejemplo 1, multiplicando por 4.

La siguiente es la sintaxis para usar con este programa.

*Row 4, A, 2

Nombre de matriz

Multiplicador

Ejecutando este programa produce el resultado siguiente.

(Modo MAT)

	1	2
1	1	2
2	12	16
3	5	6



③ Y = Type ↵

"X ^ 4 - X ^ 3 - 24X^2 + 4X + 80" → Y1 ↵

⑤ G SelOn 1 ↵

⑥ Orange G1 ↵

⑦ DrawGraph

③ F4 F4 F3 F1

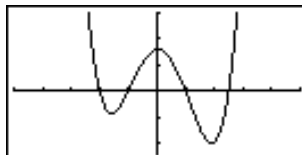
④ VARS F4 F1 EXIT EXIT

⑤ F4 F4 F1 F1 EXIT

⑥ F4 F2

⑦ SHIFT PRGM F6 F2 F2

Ejecutando este programa produce el resultado que se muestra aquí.



■ Usando las funciones de gráfico dinámico dentro de un programa

El uso de las funciones de gráfico dinámico permite realizar operaciones repetidas del gráfico dinámico. A continuación se muestra cómo especificar la gama de un gráfico dinámico dentro de un programa.

• Gama de gráfico dinámico

1 → D Inicio ↵

5 → D Finalización ↵

1 → D Intervalo ↵

Programa de ejemplo

ClrGraph ↵

View Window -5, 5, 1, -5, 5, 1 ↵

Y = Type ↵

"AX + 1" → Y1 ↵

② D SelOn 1 ↵

③ D Var A ↵

1 → ④ D Start ↵

5 → ⑤ D End ↵

1 → ⑥ D pitch ↵

⑦ DrawDyna

① VARS F4 F1 EXIT EXIT

② F4 F5 F1

③ F3

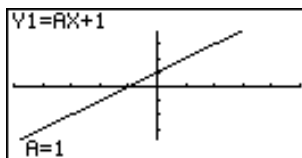
④ VARS F5 F1

⑤ F2

⑥ F3

⑦ SHIFT PRGM F6 F2 F3

Ejecutando este programa produce el resultado que se muestra aquí.



⋮ ↑
↓ ⋮



■ Usando las funciones de tabla y gráfico dentro de un programa

Las funciones de tabla y gráfico (Table & Graph) dentro de un programa pueden generar tablas numéricas y realizar operaciones gráficas. A continuación se muestran los variados tipos de sintaxis que se usan cuando se programa con las funciones de tabla y gráfico.

- Ajuste de gama de tabla
 - 1 → F Inicio ↵
 - 5 → F Finalización ↵
 - 1 → F Intervalo ↵
- Generación de tabla numérica
 - DispF-Tbl ↵
- Operación de delineado de gráfico
 - Tipo conectado: DrawFTG-Con ↵
 - Tipo de puntos marcados: DrawFTG-Plt ↵

Programa de ejemplo

ClrGraph ↵

ClrText ↵

View Window 0, 6, 1, -2, 106, 2 ↵

Y = Type ↵

"3X² - 2" → Y1 ↵

① T SelOn 1 ↵

0 → ② F Start ↵

6 → ③ F End ↵

1 → ④ F pitch ↵

⑤ DispF-Tbl ↵

⑥ DrawFTG-Con

① F4 F6 F1 F1

② VARS F6 F1 F1

③ F2

④ F3

⑤ SHIFT PRGM F6 F2 F4 F1

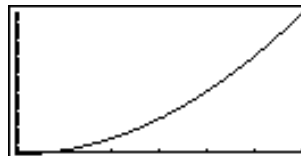
⑥ SHIFT PRGM F6 F2 F4 F2

Ejecutando este programa produce el resultado que se muestra aquí.

Tabla numérica

X	Y1
1	-1
2	10
3	25

Gráfico





P.218

■ Usando las funciones de tabla y gráfico de recurrencia dentro de un programa

La incorporación de las funciones de tabla y gráfico de recurrencia dentro de un programa, le permite generar tablas numéricas y realizar operaciones gráficas. A continuación se muestran los variados tipos de sintaxis que se usan cuando se programa con las funciones de tabla y gráfico de recurrencia.

- Ingreso de fórmula de recurrencia

a_{n+1} Type \downarrow Especifica el tipo de recurrencia.

" $3a_n + 2$ " $\rightarrow a_{n+1}$ \downarrow

" $4b_n + 6$ " $\rightarrow b_{n+1}$ \downarrow

- Ajuste de gama de tabla

1 \rightarrow R Start \downarrow

5 \rightarrow R End \downarrow

1 $\rightarrow a_0$ \downarrow

2 $\rightarrow b_0$ \downarrow

1 $\rightarrow a_n$ Start \downarrow

3 $\rightarrow b_n$ Start \downarrow

- Generación de tabla numérica

DispR-Tbl \downarrow

- Operación de delineado de gráfico

Tipo conectado: DrawR-Con \downarrow , DrawR Σ -Con \downarrow

Tipo de puntos marcados: DrawR-Plt \downarrow , DrawR Σ -Plt \downarrow

- Gráfico de convergencia/divergencia estadística (WEB graph)

DrawWeb a_{n+1} , 10 \downarrow

Programa de ejemplo

ClrGraph \downarrow

View Window 0, 1, 1, 0, 1, 1 \downarrow

① a_{n+1} Type \downarrow

① [F4] [F6] [F2] [F3] [F2] [EXIT]

" $-3a_n^2 + 3a_n$ " $\rightarrow a_{n+1}$ \downarrow

② [F4] [F2]

" $3b_n - 0.2$ " $\rightarrow b_{n+1}$ \downarrow

0 \rightarrow ③ R Start \downarrow

③ [VARS] [F6] [F2] [F2] [F1]

6 \rightarrow R End \downarrow

0.01 $\rightarrow a_0$ \downarrow

0.11 $\rightarrow b_0$ \downarrow

0.01 $\rightarrow a_n$ Start \downarrow

0.11 $\rightarrow b_n$ Start \downarrow

④ DispR-Tbl \downarrow

④ [SHIFT] [PRGM] [F6] [F2] [F5] [F1]

⑤ DrawWeb a_{n+1} , 30

⑤ [SHIFT] [PRGM] [F6] [F2] [F5] [F2] [EXIT] [EXIT] [EXIT]

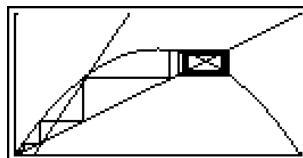
⑥ [F4] [F6] [F2] [F4] [F3]

Ejecutando este programa produce el resultado que se muestra aquí.

Tabla numérica

$n+1$	$\Delta n+1$	$\Delta^2 n+1$
0	0.01	0.11
1	0.0297	0.13
2	0.0864	0.19
3	0.2369	0.37

Gráfico de recurrencencia



P.234

■ Usando las funciones de clasificación de lista dentro de un programa

Estas funciones le permiten clasificar los datos de las listas en orden ascendente o descendente.

- Orden ascendente

① SortA (List 1, List 2, List 3)

— Listas a ser clasificadas (se pueden especificar hasta seis listas)

① [F4] [F3] [F1] [EXIT] ② [OPTN] [F1] [F1]

- Orden descendente

SortD (List 1, List 2, List 3)

— Listas a ser clasificadas (se pueden especificar hasta seis listas)

■ Usando la función de cálculo de resolución en un programa

Puede incorporar una función de cálculo de resolución en un programa.

La siguiente es la sintaxis para usar la función de resolución en un programa.

Solve($f(x)$, n , a , b)

— Límite superior
— Límite inferior
— Valor estimado inicial

Programa de ejemplo

① Solve($2X^2 + 7X - 9$, 1, 0, 1) ① [OPTN] [F4] [F1]

- En la función $f(x)$, solamente puede usarse X como una variable en las expresiones. Las otras variables (A a Z, r, θ) son tratadas como constantes, y el valor actualmente asignado a estas variables se aplican durante el cálculo.
- El ingreso de los cierres de paréntesis, límite inferior a y límite superior b pueden omitirse.
- Las soluciones obtenidas usando la resolución (Solve) puede incluir errores.
- Tenga en cuenta que no puede usar una resolución, diferencial, diferencial cuadrática, integración, valor máximo/mínimo o expresión de cálculo de Σ dentro de un término de cálculo de resolución.



P.250

■ Usando gráficos y cálculos estadísticos dentro de un programa

La inclusión de operaciones gráficas y cálculos estadísticos dentro de los programas le permiten calcular y graficar datos estadísticos.

● Para ajustar las condiciones y delinear un gráfico estadístico

Siguiendo a "StatGraph", debe especificar las siguientes condiciones de gráfico:

- Condición de delineado/no delineado de gráfico (DrawOn/DrawOff).
- Tipo de gráfico.
- Ubicación de dato del eje x (nombre de lista).
- Ubicación de dato del eje y (nombre de lista).
- Ubicación de datos de frecuencia (nombre de lista).
- Tipo de marca.
- Color de gráfico



P.252

Las condiciones gráficas que se requieren dependen en el tipo de gráfico. Para los detalles, vea la sección "Cambiando los parámetros de un gráfico".

- La siguiente es una especificación de condición gráfica típica para un gráfico de dispersión o gráfico lineal xy .

S-Gph1 DrawOn, Scatter, List1, List2, 1, Square, Blue ↵ ↵

En el caso de un gráfico lineal xy reemplace "Scatter" en la especificación anterior con " xy Line".

- La siguiente es una especificación de condición gráfica típica para una marcación de puntos de una probabilidad normal.

S-Gph1 DrawOn, NPPlot, List1, Square, Blue ↵ ↵

- La siguiente es una especificación de condición gráfica típica para un gráfico de una sola variable.

S-Gph1 DrawOn, Hist, List1, List2, Blue ↵ ↵

Para los siguientes tipos de gráficos puede usarse el mismo formato, reemplazando simplemente "Hist" en la especificación anterior con el tipo de gráfico aplicable.

Histograma: Hist
 Recuadro de mediana: MedBox
 Recuadro de media: MeanBox
 Distribución normal: N-Dist
 Línea de trazos: Broken



P.254



- La siguiente es una especificación de condición gráfica típica para un gráfico de regresión.

S-Gph1 DrawOn, Linear, List1, List2, List3, Blue, ↵

El mismo formato puede usarse para los tipos siguientes de gráficos, reemplazando simplemente "Linear" en la especificación anterior con el tipo de gráfico aplicable.

- Regresión lineal: Linear
- Med-Med: Med-Med
- Regresión cuadrática: Quad
- Regresión cúbica: Cubic
- Regresión de cuarto grado: ... Quart
- Regresión logarítmica: Log
- Regresión exponencial: Exp
- Regresión de potencia: Power

- La siguiente es una especificación de condición gráfica típica para un gráfico de regresión senoidal.

S-Gph1 DrawOn, Sinusoidal, List1, List2, Blue, ↵

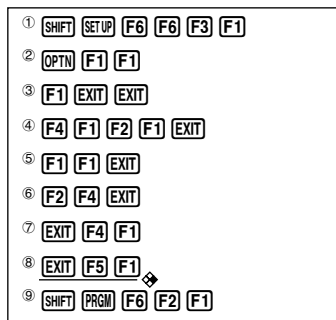
- La siguiente es una especificación de condición gráfico típica para un gráfico de regresión logística.

S-Gph1 DrawOn, Logistic, List1, List2, Blue, ↵

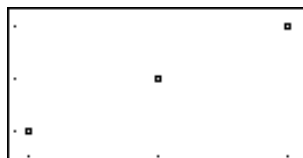
Programa de ejemplo

```

ClrGraph ↵
① S-Wind Auto ↵
② {1, 2, 3} → List 1 ↵
③ {1, 2, 3} → List 2 ↵
④ S-Gph1 DrawOn, ⑤ Scatter, List1, List2, 1, Square, ⑥ Blue, ↵
⑦ DrawStat
    
```



Ejecutando este programa produce el gráfico de dispersión mostrado aquí.



■ Realizando cálculos estadísticos

- Cálculos estadísticos con una sola variable

① 1-Variable List 1, List 2

Dato de frecuencia (Frequency)

Dato de eje x (XList)

① **F4** **F1** **F6** **F1**

```
1-Variable
x̄ =2.33333333
Σx =14
Σx² =36
x̄n =0.74535599
x̄n-1 =0.81649658
n =6
```

- Cálculos estadísticos con dos variables

2-Variable List 1, List 2, List 3

Dato de frecuencia (Frequency)

Dato de eje y (YList)

Dato de eje x (XList)

```
2-Variable
x̄ =2
Σx =6
Σx² =14
x̄n =0.81649658
x̄n-1 =1
n =3
```

- Cálculos de regresión estadística

① LinearReg List 1, List 2, List 3

Tipo de
cálculo*

Dato de frecuencia (Frequency)

Dato de eje y (YList)

Dato de eje x (XList)

① **F4** **F1** **F6** **F6** **F1**

```
LinearReg
a =1
b =0
r =1
r² =1
y=ax+b
```

- * Como tipo de cálculo puede especificarse cualquiera de los siguientes tipos.

LinearReg Regresión lineal.

Med-MedLine . Cálculo de Med-Med.

QuadReg Regresión cuadrática.

CubicReg Regresión cúbica.

QuartReg Regresión cuártica.

LogReg Regresión logarítmica.

ExpReg Regresión exponencial.

PowerReg Regresión de potencia.

- Cálculo estadístico de regresión senoidal

SinReg List 1, List 2

Dato de eje y (YList)

Dato de eje x (XList)

- Cálculo estadístico de regresión logística

LogisticReg List 1, List 2

Dato de eje y (YList)

Dato de eje x (XList)