



Programación

- 8-1 Pasos de programación básicos
- 8-2 Teclas de función del modo de programa
- 8-3 Editando los contenidos de un programa
- 8-4 Administración de archivos
- 8-5 Referencia de mandos
- 8-6 Usando las funciones de la calculadora en los programas
- 8-7 Lista de mandos del modo de programa
- 8-8 Biblioteca de programas

Esta unidad viene con aproximadamente 144 kbytes de memoria.

- Puede verificar la cantidad de memoria que se está usando y la cantidad restante, ingresando el modo SYSTEM desde el menú principal, y luego presionando **[F1]**(Mem). Para los detalles vea la parte titulada “9-2 Operaciones con la memoria”.

8-1 Pasos de programación básicos

Descripción

Los mandos y cálculos son ejecutados secuencialmente, exactamente como en las instrucciones múltiples del cálculo manual.

Ajustes básicos

1. Desde el menú principal, ingrese el modo PRGM. Al hacerlo, una lista de mandos y programas aparece sobre la presentación.

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

Los archivos se listan en la secuencia alfabética de sus nombres.

Program List		
AREA	*	34
GRAPHICS	:	56
MEASURE	:	66
OCTA	:	44
OCTONARY	:	89
TRIANGLE	:	69
EXE I EDIT NEW I DEL I DELA I D		

Ejecución

2. Registre un nombre de archivo.
3. Ingrese el programa.
4. Ejecute el programa.



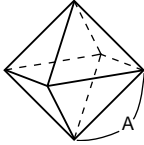
- # Si no hay programas almacenados en la memoria cuando ingresa el modo PRGM, el mensaje **"No Programs"** aparece sobre la presentación y en el menú de funciones solamente se muestra el ítem NEW (F3).
- # Los valores a la derecha de la lista de programas indican el número de bytes usados por cada programa.
- # Un nombre de archivo puede tener hasta ocho caracteres de longitud.

- # Los siguientes son los caracteres que pueden usarse en el nombre de un archivo:
A hasta la Z, r, θ , espacios, [,] , { , } , ' , " , ~ , 0 al 9, . , + , - , X , \div
- # El registro de un nombre de archivo utiliza 24 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 y retornar a la lista de programas sin ingresar un nombre de archivo, presione **ESC**.



Ejemplo 1 Calcular el área de superficie (cm²) y volumen (cm³) de tres octaedros regulares, cuando la longitud de un lado es de 7 cm, 10 cm y 15 cm.

Almacenar el programa bajo el nombre de archivo "OCTA".



Las siguientes son las fórmulas para el cálculo del área de superficie S y volumen V de un octaedro regular cuando se conoce la longitud de lado A.

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

Procedimiento

- ① **MENU** PRGM
- ② **F3** (NEW) **O** **C** **T** **A** **EXE** *1
- ③ **SHIFT** **VARS** (PRGM) **F3** (?) **→** **ALPHA** **X.θ.T** (A) **F6** (▷) **F6** (▷) **F3** (:)*2
2 **X** **SHIFT** **x²** ($\sqrt{\quad}$) **3** **X** **ALPHA** **X.θ.T** (A) **x²** **F6** (▷) **F4** (▲)
SHIFT **x²** ($\sqrt{\quad}$) **2** **÷** **3** **X** **ALPHA** **X.θ.T** (A) **△** **3**
ESC **ESC**

- ④ **F1** (EXE) o **EXE**
7 **EXE** (Valor de A)
EXE

?	7
S cuando A = 7	169.7409791
V cuando A = 7	161.6917506

- EXE**
EXE **1** **0** **EXE**
EXE

?	10
S cuando A = 10	346.4101615
V cuando A = 10	471.4045208

- EXE**
EXE **1** **5** **EXE**
EXE *3

?	15
S cuando A = 15	779.4228634
V cuando A = 15	1590.990258



*1 Presione **F3** (NEW) y el cursor cambia de forma para indicar que se ingresa un carácter alfabético.

*2 A continuación se muestra cómo el cálculo del área de superficie y volumen de un octaedro regular podrían calcularse usando un cálculo manual.

Área de superficie S . **2** **X** **SHIFT** **x²** ($\sqrt{\quad}$) **3** **X**
 <valor de A> **x²** **EXE**

Volumen V **SHIFT** **x²** ($\sqrt{\quad}$) **2** **÷** **3** **X**
 <valor de A> **△** **3** **EXE**

*3 Presionando **EXE** mientras el resultado final de un programa se encuentra sobre la presentación cambia a la lista de programas.

También puede ejecutar un programa mientras se encuentra en el modo **RUN•MAT** ingresando: Prog "<nombre de archivo>" **EXE**.

Presionando **EXE** mientras el resultado final del programa se encuentra sobre la presentación vuelve a ejecutar el programa.

Si el programa especificado por Prog "<nombre de archivo>" no puede encontrarse se producirá un error.

8-2 Teclas de función del modo de programa

- **{NEW}** ... {programa nuevo}

• Cuando está registrando un nombre de archivo

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

• Cuando está ingresando un programa — **F1(RUN)** ... fijado por omisión

- **{JUMP}** ... {parte superior}/{parte inferior} de un programa
- **{SRC}** ...{búsqueda}
- **{MAT}/{STAT}/{LIST}/{GRPH}/{DYNA}/{RECR}**
...menú de {matriz}/{estadísticas}/{lista}/{gráfico dinámico}/{recursión}
- Presionando **SHIFT VARS** (PRGM) visualiza el menú PRGM(PROGRAMA) siguiente.
 - **{Prog}** ...{llamado de programa}
 - **{JUMP}** ... {menú de mando de salto}
 - **{?}/{ \blacktriangle }** ... mandos de {entrada}/{salida}
 - **{I/O}** ... {control I/O / menú de mando de transferencia}
 - **{IF}/{FOR}/{WHLE}/{CTRL}/{LOGIC}**
... menú de mando de {salto condicional}/{control de ciclo}/{control de ciclo condicional}/{control de programa}/{operación lógica}
 - **{CLR}/{DISP}** ... menú de mando de {borrado}/{presentación}
 - **{:}** {separador de expresiones y mandos}

Para detalles en cada uno de estos mandos, vea la parte titulada “8-5 Referencia de mandos”.

- Presionando **CTRL F3** (SET UP) visualiza el menú de mandos mostrado a continuación.
 - **{ANGL}/{DISP}/{CPLX}/{GRPH}/{STAT}/{DERIV}/{T-VAR}/{ Σ DSP}**

Para detalles en cada uno de estos mandos, vea la parte titulada “Menús de teclas de funciones en la presentación de ajustes básicos” en la página 1-7-1.



• Cuando está ingresando un programa — **F2** (BASE)*¹

- **{JUMP}/{SRC}**
 - **{d-o}** ... ingreso de valor {decimal}/{hexadecimal}/{binario}/{octal}
 - **{LOG}** ... {operadores lógicos}
 - **{DISP}** ... conversión del valor visualizado a {decimal}/{hexadecimal}/{binario}/{octal}
 - **{SYBL}** ... {menú de símbolos}

 - Presiones **SHIFT** **VAR** (PRGM) visualiza el menú PRGM(PROGRAMA) siguiente.
 - **{Prog}/{JUMP}/{?}/{▲}**
 - **{= ≠ <}** ... {menú de operadores lógicos}
 - **{:}** {separador para expresiones y mandos}

 - Presionando **CTRL** **F3** (SET UP) visualiza el menú de mandos de modo mostrado a continuación.
 - **{Dec}/{Hex}/{Bin}/{Oct}**
-

- **{EXE}/{EDIT}**
... {ejecución}/{edición} de programa
- **{NEW}** ...{programa nuevo}
- **{DEL}/{DEL-A}**
... borrar {programa específico}/{todos los programas}
- **{SRC}/{REN}**
... {búsqueda}/{cambio} de nombre de programa



*¹ Los programas ingresados después de presionar **F2** (BASE) se indican mediante **B** a la derecha del nombre del archivo.

8-3 Editando los contenidos de un programa

■ 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 los mensajes de error

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

```
Ma ERROR
Press: [ESC]
```

Cuando aparece tal mensaje, presione **[ESC]** para visualizar la ubicación en donde se genera el error. El cursor estará destellando en la ubicación del problema. Compruebe la tabla “Tabla de mensajes de error” (página α-1-1) para los pasos que debe tomar para corregir la situación.

- Tenga en cuenta que presionando **[ESC]** no visualizará la ubicación del error si el programa está protegido mediante una contraseña. En su lugar, retorna a la pantalla de lista de programas.

● 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.

La tecla **[F1]** (JUMP) también es práctica cuando edita los contenidos del programa.

[F1] (JUMP) **[1]** (Parte superior) ... Mueve el cursor a la parte superior del programa.

```
=====OCTA=====
?→A: 2×√3×A²,
√2/3×A³
```

[F1] (JUMP) **[2]** (Parte inferior) ... Mueve el cursor a la parte inferior del programa.

```
=====OCTA=====
?→A: 2×√3×A²,
√2/3×A³
```

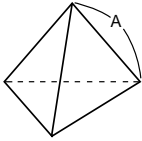
■ Usando un programa existente para crear un programa nuevo

Algunas veces puede ingresar un programa nuevo usando un programa que ya existe en la memoria como una base.



Ejemplo 2 Usar el programa llamado OCTA (página 8-1-2) que calcula el área de superficie (cm²) y volumen (cm³) de tetraedros regulares, cuando la longitud de un lado es 7 cm, 10 cm y 15 cm.

Utilice TETRA como el nombre de archivo.



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

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

Al ingresar el programa utilice las operaciones de tecla siguientes.

Longitud de un lado A $\boxed{\text{SHIFT}} \boxed{\text{VARS}} \boxed{\text{(PRGM)}} \boxed{\text{F3}} \boxed{?} \boxed{\rightarrow} \boxed{\text{ALPHA}} \boxed{\text{X,}\theta\text{T}} \boxed{\text{(A)}} \boxed{\text{F6}} \boxed{\triangleright} \boxed{\text{F6}} \boxed{\triangleright} \boxed{\text{F3}} \boxed{:}$

Area de superficie S $\boxed{\text{SHIFT}} \boxed{\text{x}^2} \boxed{(\sqrt{\quad})} \boxed{3} \boxed{\times} \boxed{\text{ALPHA}} \boxed{\text{X,}\theta\text{T}} \boxed{\text{(A)}} \boxed{\text{x}^2} \boxed{\text{F6}} \boxed{\triangleright} \boxed{\text{F4}} \boxed{\blacktriangle}$

Volumen V $\boxed{\text{SHIFT}} \boxed{\text{x}^2} \boxed{(\sqrt{\quad})} \boxed{2} \boxed{\div} \boxed{1} \boxed{2} \boxed{\times} \boxed{\text{ALPHA}} \boxed{\text{X,}\theta\text{T}} \boxed{\text{(A)}} \boxed{\wedge} \boxed{3}$

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

Longitud de un lado A $\boxed{\text{SHIFT}} \boxed{\text{VARS}} \boxed{\text{(PRGM)}} \boxed{\text{F3}} \boxed{?} \boxed{\rightarrow} \boxed{\text{ALPHA}} \boxed{\text{X,}\theta\text{T}} \boxed{\text{(A)}} \boxed{\text{F6}} \boxed{\triangleright} \boxed{\text{F6}} \boxed{\triangleright} \boxed{\text{F3}} \boxed{:}$

Area de superficie S $\boxed{2} \boxed{\times} \boxed{\text{SHIFT}} \boxed{\text{x}^2} \boxed{(\sqrt{\quad})} \boxed{3} \boxed{\times} \boxed{\text{ALPHA}} \boxed{\text{X,}\theta\text{T}} \boxed{\text{(A)}} \boxed{\text{x}^2} \boxed{\text{F6}} \boxed{\triangleright} \boxed{\text{F4}} \boxed{\blacktriangle}$

Volumen V $\boxed{\text{SHIFT}} \boxed{\text{x}^2} \boxed{(\sqrt{\quad})} \boxed{2} \boxed{\div} \boxed{3} \boxed{\times} \boxed{\text{ALPHA}} \boxed{\text{X,}\theta\text{T}} \boxed{\text{(A)}} \boxed{\wedge} \boxed{3}$

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

- Borrando $\boxed{2} \boxed{\times}$ (subrayado usando una línea ondulada encima).
- Cambiando $\boxed{3}$ por $\boxed{1} \boxed{2}$ (subrayado usando una línea sólida encima).

Ahora edite OCTA para producir el programa TETRA.

1. Edite el nombre del programa.

F6 (>) **F2** (REN) **AC** **T** **E** **T** **R** **A** **EXE**

```
Program List
TETRA      : 447
TRIANGLE   : 69
```

2. Edite los contenidos del programa.

F2 (EDIT)

```
====TETRA====
?>A:2*√3×A²,
√2/3×A³
```

▶▶▶▶ **DEL** **DEL**

▼ **◀** **DEL** **1** **2**

```
====TETRA====
?>A:√3×A²,
√2/12×A³
```

ESC

3. Trate de ejecutar el programa.

F1 (EXE) o **EXE**

7 **EXE** (Valor de A)

EXE

```
?
7
84.87048957
40.42293766
```

EXE

EXE **1** **0** **EXE**

EXE

```
?
10
173.2050808
117.8511302
```

EXE

EXE **1** **5** **EXE**

EXE

```
?
15
389.7114317
397.7475644
```

■ Buscando datos dentro de un programa



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

1. Llame el programa.
2. Presione **F2**(SRC) o **EXE** e ingrese los datos que desea encontrar.

F2(SRC)
ALPHA **X.0T** (A)

```
=====OCTA=====
?→A: 2×√3×A²,
√2/3×A³
```

```
Search For Text
-----
AC
-----
MAT ISTAT LIST GRPHIDYNARECR
```

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.*1

```
=====OCTA=====
?→A: 2×√3×A²,
√2/3×A³
SRC|
```

4. A cada presión de **EXE** o **F1**(SRC) ocasiona que el cursor salte a la instancia siguiente del dato especificado*2

```
=====OCTA=====
?→A: 2×√3×A²,
√2/3×A³
```



*1 El mensaje "Not Found" aparece cuando los datos de la búsqueda que ha especificado no pueden encontrarse en el programa.

*2 Si no hay más instancias de los datos que ha especificado, la operación de búsqueda finaliza y el cursor retorna al punto desde donde se inició la búsqueda.

No se puede especificar el símbolo de línea nueva (↵) o mando de presentación (▲) para el dato buscado.

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.

Si comete una equivocación mientras ingresa los caracteres para la búsqueda, presione **AC** para borrar el ingreso y vuelva a ingresar desde el principio.

8-4 Administración de archivos

■ Buscando un archivo

- Para encontrar un archivo usando la búsqueda por carácter inicial



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

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

F6 (>) **F1** (SRC)

O **C** **T**

```
Search For Program
[OCTA
```

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

```
Program List
OCTA      : 44↑
DICTIONARY : 89
TRIANGLE  : 69
```

- El nombre que comienza con los caracteres que ha ingresado se realzan en brillante.



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 **ESC** para borrar el mensaje de error.

■ Editando 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 y para mover la parte realzada en brillante al archivo cuyo nombre desea editar, y luego presione (>) (REN).

```
Rename
[&TRIANGLE]
```

2. Realice los cambios que desea.

```
Rename
[&ANGLE ]
```

3. Presione para registrar el nombre nuevo y retornar a la lista de programas.

La lista de programas se clasifica de acuerdo a los cambios que realiza en el nombre de archivo.

■ Borrando un programa

• Para borrar un programa específico

1. Mientras la lista de programas se encuentra sobre la presentación, utilice y para mover la parte realzada en brillante al nombre del programa que desea borrar.
2. Presione (DEL).
3. Presione (Yes) para borrar el programa seleccionado o (No) para cancelar la operación sin borrar nada.



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, aparecerá el mensaje "**Already Exists**". Cuando esto suceda, puede realizar cualquiera de las dos operaciones siguientes.

- Presione para borrar el error y retornar a la pantalla de edición de nombre de archivo.
- Presione para borrar el nombre de archivo nuevo e ingresar uno nuevo.

• Para borrar todos los programas

1. Mientras la lista de programas se encuentra sobre la presentación, presione **[F5]** (DEL-A).
 2. Presione **[EXE]** (Yes) para borrar todos los programas en la lista o **[ESC]** (No) para cancelar la operación sin borrar nada.
- También puede borrar todos los programas ingresando el modo SYSTEM desde el menú principal, y luego presionando **[F1]** (Mem) para visualizar la pantalla de administración de memoria.
Para los detalles vea la parte titulada "9-2 Operaciones con la memoria".

■ Registrando una contraseña

Cuando ingresa un programa, lo puede proteger con una contraseña que limita el acceso a los contenidos del programa para solamente aquéllos que conocen la contraseña.

- No necesita ingresar la contraseña para ejecutar un programa.



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]** (π 0) y luego ingrese la contraseña.

[F5] (π 0)

[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.
4. Luego de ingresar el programa, presione **[SHIFT] [ESC]** (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
AREA * : 34
GRAPHICS : 56

```

■ Llamando un programa protegido con una contraseña



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

1. En la lista de programas, utilice **[▲]** y **[▼]** para mover la parte realzada en brillante al nombre del programa que desea recuperar.
2. Presione **[F2]** (EDIT).
3. Ingrese la contraseña y presione **[EXE]** para llamar el programa.

```

Program Name
[AREA ]
Password?
[A ]

```



Presionando **[EXE]** sin ingresar una contraseña mientras almacena un programa nuevo, ocasiona que el archivo sea almacenado sin ninguna contraseña. Presionando **[EXE]** sin ingresar una contraseña registra solamente el nombre de archivo, sin una contraseña.

Ingresando una contraseña equivocada cuando llama un programa protegido mediante una contraseña, ocasiona que aparezca el mensaje "Mismatch". Presione **[ESC]** para retornar a la pantalla de ingreso de contraseña.

8-5 Referencia de mandos

■ Índice de mandos

Break	8-5-6
ClrGraph	8-5-11
ClrList	8-5-11
ClrMat	8-5-12
ClrText	8-5-12
DispF-Tbl, DispR-Tbl	8-5-12
Do~LpWhile	8-5-5
DrawDyna	8-5-12
DrawFTG-Con, DrawFTG-Plt	8-5-13
DrawGraph	8-5-13
DrawR-Con, DrawR-Plt	8-5-13
DrawR Σ -Con, DrawR Σ -Plt	8-5-14
DrawStat	8-5-14
DrawWeb	8-5-14
Dsz	8-5-9
For~To~(Step~)Next	8-5-4
Getkey	8-5-15
Goto~Lbl	8-5-10
If~Then~(Else~)IfEnd	8-5-4
Isz	8-5-11
Locate	8-5-16
Prog	8-5-7
Receive (/ Send (.....	8-5-17
Return	8-5-8
Stop	8-5-8
While~WhileEnd	8-5-6
? (Mando de ingreso)	8-5-2
▲ (Mando de salida)	8-5-3
: (Mando de instrucción múltiple)	8-5-3
↵ (Retorno de carro)	8-5-3
' (Delimitador de texto de comentario)	8-5-3
=, ≠, >, <, ≥, ≤ (Operadores de relación)	8-5-18



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 ítemes que siempre deben ser ingresados se muestran en negrita.
{Llaves}	Las llaves se usan para encerrar un número de ítemes, 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 ítemes 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>,"<indicador>" ? → <nombre de variable>

Ejemplo: ? → A

Descripción:

- Este mando interrumpe momentáneamente la ejecución del programa e indica solicitando el ingreso de un valor o expresión para la asignación a una variable. Si no especifica un indicador, la ejecución de este mando ocasiona que aparezca "?" indicando que la calculadora está esperando para el ingreso. Si se especifica un indicador, "<indicador>?" aparece para indicar el ingreso. No hay límite al número de caracteres que pueden especificarse para un indicador.
- 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:

- Este mando interrumpe momentáneamente la ejecución de un programa y visualiza el texto de caracteres alfabéticos o el resultado de un cálculo inmediatamente anterior.
- 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:

- Diferente al mando de salida (▲), las instrucciones conectadas con el mando de instrucción múltiple se ejecutan sin parar.
- El mando de instrucción múltiple puede usarse para enlazar dos expresiones de cálculo o dos mandos.
- 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:

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

' (Delimitador de texto de comentario)

Función: Indica un texto de comentario insertado dentro de un programa.

Descripción: Todo lo que siga al apóstrofe es tratado como un texto de comentario no ejecutable.



■ Mandos de programa (COM)

If~Then~(Else~)IfEnd

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). La instrucción "IfEnd" se ejecuta siempre siguiendo a la instrucción "Then" o instrucción "Else".

Sintaxis:

$$\text{If } \begin{array}{c} \langle \text{condición} \rangle \\ \text{expresión numérica} \end{array} \left\{ \begin{array}{c} \leftarrow \\ : \\ \blacktriangle \end{array} \right\} \text{ Then } \langle \text{instrucción} \rangle \left[\left\{ \begin{array}{c} \leftarrow \\ : \\ \blacktriangle \end{array} \right\} \langle \text{instrucción} \rangle \right]$$

$$\left\{ \begin{array}{c} \leftarrow \\ : \\ \blacktriangle \end{array} \right\} \left(\text{Else } \langle \text{instrucción} \rangle \left[\left\{ \begin{array}{c} \leftarrow \\ : \\ \blacktriangle \end{array} \right\} \langle \text{instrucción} \rangle \right] \left\{ \begin{array}{c} \leftarrow \\ : \\ \blacktriangle \end{array} \right\} \right) \text{ IfEnd}$$

Parámetros: condición, expresión numérica

Descripción:

(1) If ~ Then ~ IfEnd

- Cuando la condición es verdadera, la ejecución procede con la instrucción "Then" y luego continúa con la instrucción siguiendo "IfEnd".
- Cuando la condición es falsa, la ejecución salta a la instrucción siguiente "IfEnd".

(2) If ~ Then ~ Else ~ IfEnd

- Cuando la condición es verdadera, la ejecución procede con la instrucción "Then" y luego continúa con la instrucción siguiendo "IfEnd".
- Cuando la condición es falsa, la ejecución salta a la instrucción "Else" y luego continúa con la instrucción siguiente "IfEnd".

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 incrementado de acuerdo al valor del intervalo con cada ejecución. La ejecución continúa hasta que el valor de la variable de control excede el valor final.

Sintaxis:

$$\text{For } \langle \text{valor inicial} \rangle \rightarrow \langle \text{nombre de variable de control} \rangle \text{ To } \langle \text{valor final} \rangle \left(\text{Step } \langle \text{valor de intervalo} \rangle \right) \left\{ \begin{array}{c} \leftarrow \\ : \\ \blacktriangle \end{array} \right\}$$

Next

Parámetros:

- nombre de variable de control: A hasta la Z
- valor inicial: valor o expresión que produzca un valor (es decir $\sin x$, A, etc.)
- valor final: valor o expresión que produzca un valor (es decir $\sin x$, A, etc.)
- valor de intervalo: valor numérico (fijado por omisión: 1)

While~WhileEnd

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

Sintaxis:

$$\text{While } \begin{array}{c} \text{<condición>} \\ \text{expresión numérica} \end{array} \left\{ \begin{array}{c} \blacktriangleleft \\ : \\ \blacktriangleright \end{array} \right\} \text{<instrucción>} \left\{ \begin{array}{c} \blacktriangleleft \\ : \\ \blacktriangleright \end{array} \right\} \text{WhileEnd}$$

Parámetros: expresión

Descripción:

- Este mando repite los mandos contenidos en el ciclo en tanto su condición es verdadera (diferente de cero). Cuando la condición se convierte en falsa (0), 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 ciclo son ejecutados.

■ Mandos de control de programa (CTL)

Break

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

Sintaxis: Break

Descripción:

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



Prog

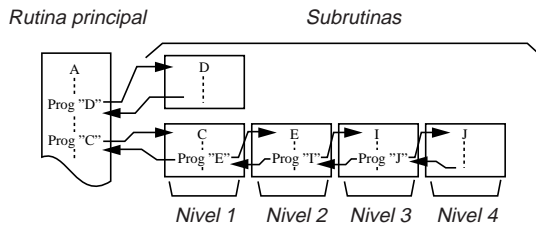
Función: Este mando especifica la ejecución de otro programa como una subrutina. En el modo RUN•MAT, este mando ejecuta un programa nuevo.

Sintaxis: Prog "nombre de archivo"

Ejemplo: Prog "ABC"

Descripción:

- Aun cuando este programa se ubica dentro de un ciclo, su ejecución rompe inmediatamente el ciclo 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.



- 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.
- Un mando "Goto~Lbl" dentro de una subrutina es válida solamente dentro de esa subrutina. No puede usarse para saltar a un rótulo fuera de la subrutina.
- Si una subrutina con el nombre de archivo especificado por el mando Prog no existe, se produce un error.
- En el modo **RUN•MAT**, 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 del mando Return dentro de una rutina principal, ocasiona la finalización de la ejecución del programa. La ejecución de un mando Return dentro de una subrutina finaliza la subrutina y retorna al programa desde donde se saltó a la subrutina.

Stop

Función: Este mando finaliza la ejecución de un programa.

Sintaxis: Stop

Descripción:

- Este mando finaliza la ejecución de un programa.
- La ejecución de este mando dentro de un ciclo finaliza la ejecución sin que se genere un mensaje de error.

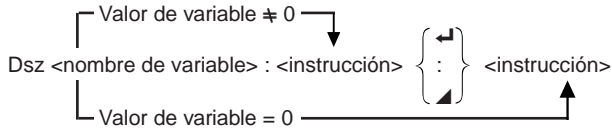


■ Mandos de salto (JUMP)

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: Disminuir 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 de instrucción múltiple (:), mando de presentación (▲), o retorno de carro (↵).



Goto~Lbl

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

Sintaxis: Goto <nombre de rótulo> ~Lbl <nombre de rótulo>

Parámetros: nombre de rótulo: valor (0 al 9), variable (A hasta la Z, r, θ)

Descripción:

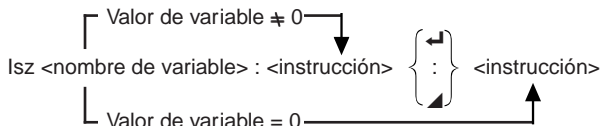
- Este mando consiste de dos partes: Goto n (en donde n es un parámetro como se ha descrito) y Lbl n (en donde n es el parámetro especificado por Goto n). Este mando ocasiona que la ejecución del programa salte a la instrucción "Lbl" cuyo parámetro n coincide con el valor especificado por la instrucción "Goto".
- Este mando puede usarse para hacer un ciclo de nuevo al comienzo de un programa o para saltar a cualquier ubicación dentro del programa.
- Este mando puede usarse en combinación con saltos condicionales y saltos de cuenta.
- Si no hay ninguna instrucción "Lbl" cuyo valor coincida con el valor especificado por la instrucción "Goto", se producirá un error.



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 : Aumentar 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 (\blacktriangleleft), o retorno de carro (\blacktriangleleft).

■ Mandos de borrado (CLR)

ClrGraph

Función: Este mando borra la pantalla de gráfico y retorna el ajuste de la ventana de visualización a los valores iniciales INIT.

Sintaxis: ClrGraph

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 <nombre de lista>

ClrList

Parámetros: nombre de lista: 1 a 20, Ans

Descripción: Este mando borra los datos en la lista especificada por el "nombre de lista". Todos los datos de lista son borrados si no se especifica nada para el "nombre de lista".

ClrText

Función: Este mando borra la pantalla de texto.

Sintaxis: ClrText

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

ClrMat

Función: Este mando borra los datos de matrices.

Sintaxis: ClrMat <nombre de matriz>

ClrMat

Parámetros: nombre de matriz: A hasta la Z, Ans

Descripción: Este mando borra los datos en la matriz especificada por el “nombre de matriz”. Todos los nombres de matrices serán borrados si no se especifica nada para el “nombre de matriz”.

■ Mandos de presentación (DISP)

DispF-Tbl, DispR-Tbl**Sin parámetros**

Función: Estos mandos visualizan tablas numéricas.

Descripción:

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

DrawDyna**Sin parámetros**

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

Descripción: Este mando delinea un gráfico dinámico durante la ejecución de un programa, de acuerdo con los parámetros de delineado definidos dentro del programa.



DrawFTG-Con, DrawFTG-Plt**Sin parámetros**

Función: Estos mandos utilizan valores en una tabla generada para graficar una función.

Descripción:

- Estos mandos delinear un gráfico de funciones de acuerdo con las condiciones actuales.
- DrawFTG-Con produce un gráfico de tipo conectado, mientras DrawFTG-Plt produce un gráfico de tipo de marcación de puntos.

DrawGraph**Sin parámetros**

Función: Este mando delinea un gráfico.

Descripción:

- Este mando traza un gráfico de acuerdo con las condiciones actuales.

DrawR-Con, DrawR-Plt**Sin parámetros**

Función: Estos mandos utilizan valores en una tabla generada para graficar una expresión de recursión con $a_n(b_n$ o $c_n)$ como el eje vertical y n como el eje horizontal.

Descripción:

- Estos mandos grafican expresiones de recursión de acuerdo a las condiciones actuales, con $a_n(b_n$ o $c_n)$ como el eje vertical y n como el eje horizontal.
- DrawR-Con produce un gráfico de tipo conectado, mientras DrawR-Plt produce un gráfico de tipo de marcación de puntos.



Draw Σ -Con, Draw Σ -Plt

Sin parámetros

Función: Estos mandos utilizan valores en una tabla generada para graficar expresiones de recursión, con Σa_n (Σb_n o Σc_n) como el eje vertical y n como el eje horizontal.

Descripción:

- Estos mandos grafican expresiones de recursión de acuerdo a las condiciones actuales, con Σa_n (Σb_n o Σc_n) como el eje vertical y n como el eje horizontal.
- Draw Σ -Con produce un gráfico de tipo conectado, mientras Draw Σ -Plt produce un gráfico de tipo de marcación de puntos.

DrawStat

Función: Este mando delinea un gráfico estadístico.

Sintaxis: Vea la parte titulada "8-6-9 Usando gráficos y cálculos estadísticos dentro de un programa".

Descripción:

Este mando delinea un gráfico estadístico de acuerdo con las condiciones del gráfico estadístico.

DrawWeb

Función: Este mando grafica convergencias/divergencias de una expresión de recursión (gráfico WEB).

Sintaxis: DrawWeb <expresión de recursión>, <número de líneas>

Ejemplo: DrawWeb a_{n+1} (b_{n+1} o c_{n+1}), 5

Descripción:

- Este mando grafica convergencias/divergencias de una expresión de recursión (gráfico WEB).
- Omitiendo la especificación del número de líneas automáticamente se especifica el valor fijado por omisión de 30.



■ Mandos de entrada/salida (I/O)

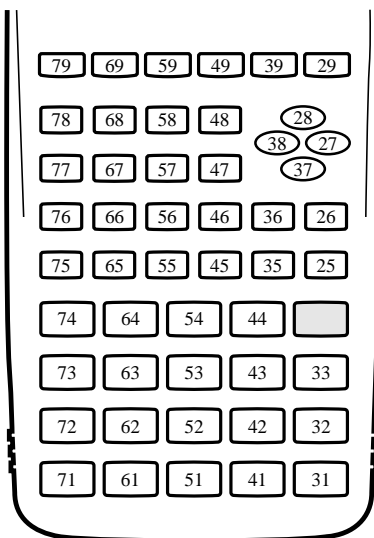
Getkey

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

Sintaxis: Getkey

Descripción:

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



- Se retorna un valor de cero, si previamente no se ha presionado ninguna tecla antes de la ejecución de este mando.
- Este mando puede usarse dentro de un ciclo.

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>, <expresión numérica>

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 y expresión numérica.
- serie: serie de caracteres.

Descripción:

- Este mando visualiza valores (incluyendo contenidos de variables) o texto en una posición especificada sobre la pantalla de texto. Si hay un cálculo ingresado, se visualiza ese resultado de cálculo.
- La fila está designada por un valor de 1 al 7, y la columna se designa por un valor de 1 al 21.



Ejemplo: Cls ↵

Locate 7, 1, "CASIO FX"

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

- En algunos casos, el mando ClrText debe ser ejecutado antes de realizar la ejecución del programa anterior.

Receive (/ Send (

Función: Este mando recibe y envía datos desde un dispositivo conectado externo.

Sintaxis: Receive (<dato>) / Send (<dato>)

Descripción:

- Este mando recibe y envía datos desde a un dispositivo conectado.
- Los siguientes tipos de mandos pueden ser recibidos por este mando.
 - Valores individuales asignados a variables.
 - Datos de matrices (todos los valores - excepto los valores individuales que no pueden ser especificados)
 - Datos de lista (todos los valores - excepto los valores individuales que no pueden ser especificados)



■ Operadores de relación (REL)

$=, \neq, >, <, \geq, \leq$

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

Sintaxis:

<lado izquierdo> <operador de relación> <lado derecho>

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: $=, \neq, >, <, \geq, \leq$



8-6 Usando las funciones de la calculadora en los programas

■ Presentación de texto

Puede incluirse 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
"CASIO"	CASIO
? → X	?
"X =" ? → X	X = ?

- Si el texto es seguido por una fórmula de cálculo, asegúrese de insertar un mando de presentación (\blacktriangle) 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 se desplaza automáticamente si el texto ocasiona que la pantalla se llene.

■ 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 **RUN • MAT** y luego usar el editor 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 \leftarrow Filas a ser transpuestas
 Nombre de matriz

Mat A

Ejecutando este programa produce el resultado siguiente.

Ans	1	2
1	1	2
2	5	6
3	3	4

• Para sumar dos filas (Row+)



Ejemplo 4 Sumar la fila 2 a la fila 3 de la matriz del Ejemplo 1.

La siguiente es la sintaxis para usar con este programa.

Row+ A, 2, 3 ↵

└─── Filas a ser sumadas
└─── Filas para las cuales se calcula el producto escalar.
└─── Nombre de matriz

Mat A

Ejecutando este programa produce el resultado siguiente.

Ans	1	2
1	1	2
2	3	4
3	8	10

■ Usando las funciones gráficas dentro de un programa

Se pueden incorporar funciones gráficas dentro de un programa para delinear gráficos complejos y para superponer gráficos uno encima de otro. A continuación se muestran los variados tipos de sintaxis que se usan cuando se programa con funciones gráficas.

- Ventana de visualización

View Window -5, 5, 1, -5, 5, 1 ↵

- Ingreso de función de gráfico

Y = Type ↵ Especifica el tipo de gráfico.

"X² - 3" → Y1 ↵

- Operación de delineado de gráfico

DrawGraph ↵

Programa de ejemplo

① ClrGraph ↵

② View Window -10, 10, 2, -120, 150, 50 ↵

③ Y = Type ↵

"X⁴ - X³ - 24X² + 4X + 80" → Y1 ↵

⑤ G SelOn 1 ↵

⑥ DrawGraph

① SHIFT VARS F6 F6 F1 2 ESC

② SHIFT OPTN F1 ESC

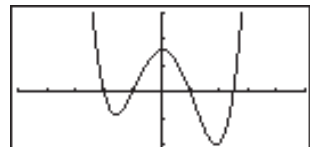
③ F6 F1 3 1

④ VARS F4 1 ESC

⑤ F6 F1 1

⑥ SHIFT VARS F6 F6 F2 2

Ejecutando este programa produce el resultado que se muestra aquí.



• Sintaxis de otras funciones gráficas

• V-Window

View Window <Xmin>, <Xmax>, <Xscale>, <Ymin>, <Ymax>, <Yscale>, <Tθmin>, <Tθmax>, <Tθpitch>

StoV-Win <área de V-Win> área: 1 a 6

RclV-Win <área de V-Win> área: 1 a 6

• Zoom

Factor <factor de X>, <factor de Y>

ZoomAuto Sin parámetro

• Pict

StoPict <área de imagen> área: 1 a 20

RclPict <área de imagen> área: 1 a 20

• Sketch

PlotOn <coordenada de X>, <coordenada de Y>

PlotOff <coordenada de X>, <coordenada de Y>

PlotChg <coordenada de X>, <coordenada de Y>

PxlOn<número de línea>, <número de columna>

PxlOff<número de línea>, <número de columna>

PxlChg<número de línea>, <número de columna>

PxlTest(<número de línea>, <número de columna>[])

F-Line <coordenada de X 1>, <coordenada de Y 1>, <coordenada de X 2>, <coordenada de Y 2>

Text <número de línea>, <número de columna>, "<texto>"

Text <número de línea>, <número de columna>, <expresión>

Tangent <función>, <coordenada de X>

Normal <función>, <coordenada de X>

Inverse <función>

Circle <coordenada de X de punto central>, <coordenada de Y de punto central>, <valor de radio R>

Vertical <coordenada de X>

Horizontal <coordenada de Y>



■ Usando las funciones del gráfico dinámico dentro de un programa

El uso de las funciones del 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 del gráfico dinámico

1 → D Start ↵

5 → D End ↵

1 → D pitch ↵

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 1 ESC

② F6 F2 1

③ F2 3

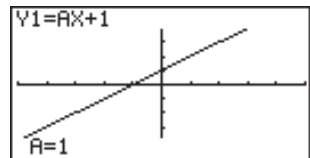
④ VARS F5 1

⑤ F5 2

⑥ F5 3

⑦ SHIFT VARS F6 F6 F2 3

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 Start ↵
 - 5 → F End ↵
 - 1 → F pitch ↵
- Generación de tabla numérica
 - DispF-Tbl ↵
- Operación de delineado de gráfico
 - Tipo conectado: DrawFTG-Con ↵
 - Tipo de marcación de puntos: DrawFTG-Plt ↵

Programa de ejemplo

ClrGraph ↵

ClrText ↵

View Window 0, 6, 1, -20, 106, 10 ↵

Y = Type ↵

"3X² - 2" → Y1 ↵

① G SelOn 1 ↵

0 → ② F Start ↵

6 → ③ F End ↵

1 → ④ F pitch ↵

⑤ DispF-Tbl ↵

⑥ DrawFTG-Con

① **F6** **F1** **1**

② **VAR** **F6** **F1** **1**

③ **F1** **2**

④ **F1** **3**

⑤ **SHIFT** **VAR** **F6** **F6** **F2** **4** **1**

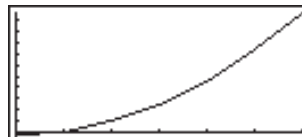
⑥ **SHIFT** **VAR** **F6** **F6** **F2** **4** **2**

Ejecutando este programa produce el resultado que se muestra aquí.

Tabla numérica

X	Y1
0	-2
1	1
2	10
3	25

Gráfico



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

La incorporación de las funciones de tabla y gráfico de recursión 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 recursión.

- Ingreso de fórmula de recursión

a_{n+1} Type ↵ ... Especifica el tipo de recursión.

" $3a_n + 2$ " → a_{n+1} ↵

" $4b_n + 6$ " → b_{n+1} ↵

- Ajuste de gama de tabla

1 → R Start ↵

5 → R End ↵

1 → a_0 ↵

2 → b_0 ↵

1 → a_n Start ↵

3 → b_n Start ↵

- Generación de tabla numérica

DispR-Tbl ↵

- Operación de delineado de gráfico

Tipo conectado: DrawR-Con ↵, DrawRΣ-Con ↵

Tipo de marcación de puntos: DrawR-Plt ↵, DrawRΣ-Plt ↵

- Gráfico de convergencia/divergencia estadística (gráfico WEB)

DrawWeb a_{n+1} , 10 ↵



Programa de ejemplo

View Window 0, 1, 1, -0.2, 1, 1 ↵

① a_{n+1} Type ↵② $-3a_n^2 + 3a_n$ → a_{n+1} ↵

④ 0 → R Start ↵

⑤ 6 → R End ↵

⑥ 0.01 → a_0 ↵⑦ 0.01 → a_n Start ↵

⑧ DispR-Tbl ▲

⑨ DrawWeb a_{n+1} , 30

① F6 F3 6 2

② F3 1 2

③ F3 1 3

④ VARS F6 F2 2 1

⑤ F2 2 2

⑥ F2 2 3

⑦ F2 2 C

⑧ SHIFT VARS F6 F6 F2 5 1

⑨ F2 5 2 ESC

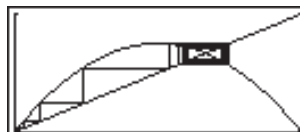
⑩ F6 F3 1 3

Ejecutando este programa produce el resultado que se muestra aquí.

Tabla numérica

$n+1$	$3n+1$
0	0.01
1	0.0297
2	0.0864
3	0.2369

Gráfico de recursión

**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 F5 1 ② F4 4

- Orden descendente

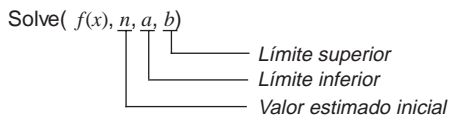
③ SortD (List 1, List 2, List 3)

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

③ F5 2

■ Usando la función de cálculo de resolución en un programa

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



Programa de ejemplo

①Solve($2X^2 + 7X - 9$, 1, 0, 1)

① OPTN F4 7

- 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 aplica durante el cálculo.
- El ingreso de los cierres de paréntesis, límite inferior a y límite superior b pueden omitirse.

■ 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 al gráfico "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



Las soluciones obtenidas usando la resolución (Solve) puede incluir errores.

No puede usar una diferencial, diferencial cuadrática, integración, Σ , valor máximo/mínimo o expresión de cálculo de resolución dentro de un término de cálculo de resolución.

Las condiciones gráficas que se requieren dependen en el tipo de gráfico. Vea la parte titulada “Cambiando los parámetros de un gráfico” (página 6-1-2).

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

S-Gph1 DrawOn, Scatter, List 1, List 2, 1, Square ↵

En el caso de un gráfico lineal xy , reemplace “Scatter” en la especificación anterior con “xyLine”.

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

S-Gph1 DrawOn, NPPlot, List 1, Square ↵

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

S-Gph1 DrawOn, Hist, List 1, List 2 ↵

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 Modified
 Distribución normal N-Dist
 Línea discontinua Broken

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

S-Gph1 DrawOn, Linear, List 1, List 2, List 3 ↵

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 cuártica 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 sinusoidal.

S-Gph1 DrawOn, Sinusoidal, List 1, List 2 ↵

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

S-Gph1 DrawOn, Logistic, List 1, List 2 ↵

Programa de ejemplo

ClrGraph ↵

① S-Wind Auto ↵

{1, 2, 3} → List 1 ↵

{1, 2, 3} → List 2 ↵

② S-Gph1 DrawOn, Scatter, List 1, List 2, 1, Square ↵

③ DrawStat

① CTRL F3 F5 1 1 ESC

② F4 1 1

③ F4 2 1

④ F4 3 1

⑤ F4 5 1

⑥ SHIFT VARS F6 F6 F2 1

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)

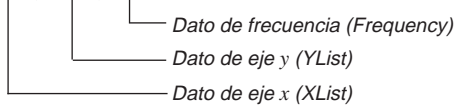
① F4 6 1

```

1-Variable
Σx      =2.33333333
Σx      =14
Σx²     =36
x̄σn     =0.74535599
x̄σn-1   =0.81649658
n        =6
↓
  
```

- Cálculos estadísticos con dos variables

① 2-Variable List 1, List 2, List 3



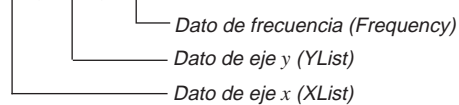
① [F4] [6] [2]

```
2-Variable
Σx =2
Σy =6
Σx² =14
Σxy =0.81649658
x̄ = 2/3
ȳ = 3
n =3
```

- Cálculos estadísticos de regresión

① LinearReg List 1, List 2, List 3

Tipo de
cálculo*



① [F4] [6] [3]

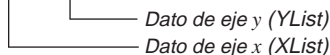
```
LinearReg
a =1
b =0
r =1
r² =1
y=ax+b
```

- * 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

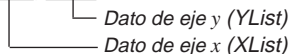
- Cálculo estadístico de regresión sinusoidal

SinReg List 1, List 2



- Cálculo estadístico de regresión logística

LogisticReg List 1, List 2



8-7 Lista de mandos del modo de programa

Programa RUN

Nivel 1	Nivel 2	Nivel 3	Mando	
MAT	Swap		Swap_	
	*Row		*Row_	
	*Row+		*Row+_	
	Row+		Row+_	
STAT	S-GPH	S-Gph1	S-Gph1_	
		S-Gph2	S-Gph2_	
		S-Gph3	S-Gph3_	
	DRAW	On		DrawOn
		Off		DrawOff
	GRAPH	Scat		Scatter
		xyLine		xyLine
		NPPlot		NPPlot
		Hist		Hist
		Box		MedBox
		ModBox		ModifiedBox
		N-Dist		N-Dist
		Broken		Broken
		Linear		Linear
		MedMed		Med-Med
		Quad		Quad
		Cubic		Cubic
		Quart		Quart
		Log		Log
	Exp		Exp	
	Power		Power	
	Sin		Sinusoidal	
	Lgstic		Logistic	
	List		List_	
	MARK	<input type="checkbox"/>		Square
		<input type="checkbox"/>		Cross
		.		Dot
CALC	1VAR		1-Variable	
	2VAR		2-Variable	
	Linear		LinearReg	
	MedMed		Med-MedLine	
	Quad		QuadReg	
	Cubic		CubicReg	
	Quart		QuartReg	
	Log		LogReg	
	Exp		ExpReg	
	Power		PowerReg	
	Sin		SinReg	
Lgstic		LogisticReg		
LIST	SortA		SortA(
	SortD		SortD(

GRP	SelOn	G_SelOn_		
		SelOff	G_SelOff_	
TYPE	Y=		Y=TYPE	
	r=		r=TYPE	
	Param		ParamTYPE	
	X=C		X=CTYPE	
	Y>		Y>Type	
	Y<		Y<Type	
	Y≥		Y≥Type	
	Y≤		Y≤Type	
	GMEM	Store		StoGMEM
		Recall		RclGMEM
DYNA	SelOn		D_SelOn_	
	SelOff		D_SelOff_	
	Var		D_Var	
	TYPE	Y=	Y=Type	
RECR	n,an..	n	n	
		an	an	
		an+1	an+1	
		bn	bn	
		bn+1	bn+1	
		cn	cn	
		cn+1	cn+1	
		SelOn		R_SelOn_
		SelOff		R_SelOff_
		Sel a0		Sel_a0
Sel a1		Sel_a1		
TYPE	an		anType	
	an+1		an+1Type	
	an+2		an+2Type	

Tecla[OPTN]				
Nivel 1	Nivel 2	Nivel 3	Mando	
LIST	List		List	
	Dim		Dim_	
	Seq		Seq(
	Min		Min(
	Max		Max(
	Mean		Mean(
	Median		Median(
	Sum		Sum_	
	Prod		Prod_	
	CumL		CumL	
	%		Percent_	
	ΔList		ΔList	
	Augmnt		Augment(
	Fill		Fill(
L→Mat		List→Mat(
MAT	Mat		Mat_	
	Dim		Dim_	
	Det		Det_	
	Trn		Trn_	
	Augmnt		Augment(
	Ident		Identity_	
	Fill		Fill(
	M→List		Mat→List(
	CPLX	Abs		Abs_
		Arg		Arg_
Conjg			Conjg_	
ReP			ReP_	
ImP			ImP_	
re^θi			re^θi	
CALC	a+bi		a+bi	
	d/dx		d/dx(
	d²/dx²		d²/dx²(
	∫dx		∫(
	Σ		Σ(
	FMin		FMin(
NUM	FMax		FMax(
	Solve		Solve(
	Abs		Abs_	
	Int		Int_	
	Frac		Frac_	
	Rnd		Rnd_	
E-SYM	Intg		Intg_	
	μ		μ	
	n		n	
	p		p	
	f		f	
	k		k	
	M		M	
	G		G	
	T		T	
	P		P	
	E		E	

PROB	xl	I		
		nPr	P	
LIST	nCr		C	
	Ran#		Ran#	
	P(P(
	Q(Q(
	R(R(
	t(t(
HYP	sinh		sinh_	
	cosh		cosh_	
	tanh		tanh_	
	sinh ⁻¹		sinh ⁻¹ _	
	cosh ⁻¹		cosh ⁻¹ _	
ANGL	tanh ⁻¹		tanh ⁻¹ _	
	°		°	
	r		r	
	g		g	
	° ' "		° ' "	
STAT	►DMS		►DMS	
	Pol(Pol(
	Rec(Rec(
FMEM	Σ		Σ	
	fn		fn	
ZOOM	Factor		Factor_	
	Auto		ZoomAuto	
SKTCH	Cls		Cls	
	PLOT	On		PlotOn_
		Off		PlotOff_
		Change		PlotChg_
	Plot		Plot_	
	LINE	F-Line		F-Line_
		Line		Line_
	GRAPH	Y=		Graph_Y=
		/dx		Graph_/
	Text		Text_	
	PIXEL	On		PxlOn_
		Off		PxlOff_
		Change		PxlChg_
		Test		PxlTest(
Tangnt		Tangent_		
Normal		Normal_		
Invrse		Inverse_		
Circle		Circle_		
Vert		Vertical_		
Horz		Horizontal_		
PICT	Store		StoPict_	
	Recall		RclPict_	
SYBL	'		'	
	"		"	
	~		~	
	*		*	
	#		#	
	° ' "		° ' "	

Lista de mandos del modo de programa

Tecla [VARS]			
Nivel 1	Nivel 2	Nivel 3	Mando
V-WIN	Xmin	Xmin	
	Xmax	Xmax	
	Xscale	Xscl	
	Xdot	Xdot	
	Ymin	Ymin	
	Ymax	Ymax	
	Yscale	Yscl	
	Tθmin	Tθmin	
	Tθmax	Tθmax	
	Tθptch	Tθptch	
	R-Xmin	RightXmin	
	R-Xmax	RightXmax	
	R-Xscl	RightXscl	
	R-Xdot	RightXdot	
	R-Ymin	RightYmin	
	R-Ymax	RightYmax	
	R-Yscl	RightYscl	
	R-Tmin	RightTθmin	
	R-Tmax	RightTθmax	
	R-Tpch	RightTθptch	
FACT	Xfact	Xfct	
	Yfact	Yfct	
STAT	n	n	
	X	\bar{x}	
		Σx	
		Σx^2	
	xσn	xσn	
	xσn-1	xσn-1	
	minX	minX	
	maxX	maxX	
	Y	\bar{y}	
		Σy	
		Σy^2	
		Σxy	
	yσn	yσn	
	yσn-1	yσn-1	
	minY	minY	
	maxY	maxY	
GRAPH	a	a	
	b	b	
	c	c	
	d	d	
	e	e	
	r	r	
	r ²	r ²	
	Q1	Q1	
	Med	Med	
	Q3	Q3	
	Mod	Mod	
	H-Strt	H_Start	
	H-ptch	H_ptch	

GRPH	PTS	x1	x1	
		y1	y1	
		x2	x2	
		y2	y2	
		x3	x3	
			y3	y3
	Yn	Y	Y	
		rn	r	
		Xtn	Xt	
		Ytn	Yt	
Xn		X		
DYNA	Start	D_Start		
	End	D_End		
	Pitch	D_pitch		
TABL	Start	F_Start		
	End	F_End		
	Pitch	F_pitch		
	Result	F_Result		
RECR	FORM	an	an	
		an+1	an+1	
		an+2	an+2	
		bn	bn	
		bn+1	bn+1	
		bn+2	bn+2	
	cn	cn		
	cn+1	cn+1		
	cn+2	cn+2		
	RANGE	R-Strt	R_Start	
		R-End	R_End	
		a0	a0	
		a1	a1	
		a2	a2	
		b0	b0	
	b1	b1		
	b2	b2		
	c0	c0		
	c1	c1		
	c2	c2		
	anStrt	anStart		
	bnStrt	bnStart		
	cnStrt	cnStart		
	Result	R_Result		
EQUA	S-Rslt	Sim_Result		
	S-Coef	Sim_Coef		
	P-Rslt	Ply_Result		
	P-Coef	Ply_Coef		

Tecla [SHIFT][VARS](PRGM)			
Nivel 1	Nivel 2	Nivel 3	Mando
Prog			Prog_
JUMP	Lbl		Lbl_
	Goto		Goto_
	Isz		Isz_
	Dsz		Dsz_
?			?
▲			▲
I/O	Locate		Locate_
	Getkey		Getkey_
	Send		Send(
	Receiv		Receive(
IF	If		If_
	Then		Then_
	Else		Else_
	IfEnd		IfEnd_
FOR	For		For_
	To		To_
	Step		Step_
	Next		Next_
WHILE	While		While_
	WhileEnd		WhileEnd_
	Do		Do_
	LpWhile		LpWhile_
CTRL	Prog		Prog_
	Return		Return_
	Break		Break_
	Stop		Stop_
LOGIC	=	≠	=
	≠	≠	≠
	>	>	>
	<	<	<
	≧	≧	≧
	≦	≦	≦
	And		And_
	Or		Or_
	Not		Not_
CLR	Text		ClrText
	Graph		ClrGraph
	List		ClrList_
	Matrix		ClrMat_
DISP	Stat		DrawStat
	Graph		DrawGraph
	Dyna		DrawDyna
	F-TBL	Table	DispF-Tbl
	G-Con		DrawFTG-Con
	G-Plot		DrawFTG-Plt
	R-TBL	Table	DispR-Tbl
	Web		DrawWeb_
	R-Con		DrawR-Con
	RE-Con		DrawRE-Con
	R-Plot		DrawR-Plt
	RE-Plt		DrawRE-Plt
:			:

Tecla [CTRL][F3](SET UP)			
Nivel 1	Nivel 2	Nivel 3	Mando
ANGL	Deg		Deg
	Rad		Rad
	Gra		Gra
DISP	Fix		Fix_
	Sci		Sci_
	Norm		Norm
	EngOn		EngOn
	EngOff		EngOff
CPLX	Real		Real
	a+bi		a+bi
	re^θi		re^θi
GRPH	G-FUNC	On	FuncOn
		Off	FuncOff
	D-TYPE	G-Con	G-Connect
		G-Plot	G-Plot
	BG	None	BG-None
		Pict	BG-Pict
	SIMUL	On	SimulOn
		Off	SimulOff
	COORD	On	CoordOn
		Off	CoordOff
GRID	On	GridOn	
	Off	GridOff	
AXES	On	AxesOn	
	Off	AxesOff	
LABEL	On	LabelOn	
	Off	LabelOff	
STAT	S-WIN	Auto	S-WindAuto
		Manual	S-WindMan
	File		File_
RESID	None	Resid-None	
	List	Resid-List	
DERIV	On	DerivOn	
	Off	DerivOff	
T-VAR	Range	VarRange	
	List	VarList_	
Σ•DSP	On	ΣdispOn	
	Off	ΣdispOff	



Lista de mandos del modo de programa

Programa BASE

Tecla [SHIFT][OPTN](V-Window)			
Nivel 1	Nivel 2	Nivel 3	Mando
V-Win			ViewWindow_
Sto			StoV-Win_
Rcl			RclV-Win_

Nivel 1	Nivel 2	Nivel 3	Mando
d-o	d		d
	h		h
	b		b
	o		o
LOG	Neg		Neg_
	Not		Not_
	and		and
	or		or
	xor		xor
	xnor		xnor
DISP	►Dec		►Dec
	►Hex		►Hex
	►Bin		►Bin
	►Oct		►Oct

Tecla [CTRL][F3](SET UP)			
Nivel 1	Nivel 2	Nivel 3	Mando
Dec			Dec
Hex			Hex
Bin			Bin
Oct			Oct

Tecla [SHIFT][VARS](PRGM)			
Nivel 1	Nivel 2	Nivel 3	Mando
Prog			Prog_
JUMP	Lbl		Lbl_
	Goto		Goto
	Isz		Isz_
	Dsz		Dsz_
?			?
▲			▲
= ≠ <	=		=
	≠		≠
	>		>
	<		<
	≧		≧
	≦		≦
:			:



8-8 Biblioteca de programas

- Cerciórese de comprobar la cantidad de bytes de memoria libre sin usar que tiene antes de intentar la realización de cualquier programación.

Nombre de programa	Factorización prima
--------------------	---------------------

Descripción

Este programa divide continuamente un número natural por factores hasta que se producen sus factores primos.

Propósito

Este programa acepta el ingreso de un número natural A, y lo divide por B (2, 3, 5, 7...) para hallar los factores primos de A.

- Si una operación de división no produce un resto, el resultado de la operación se asigna a A.
- El procedimiento anterior se repite hasta que $B > A$.

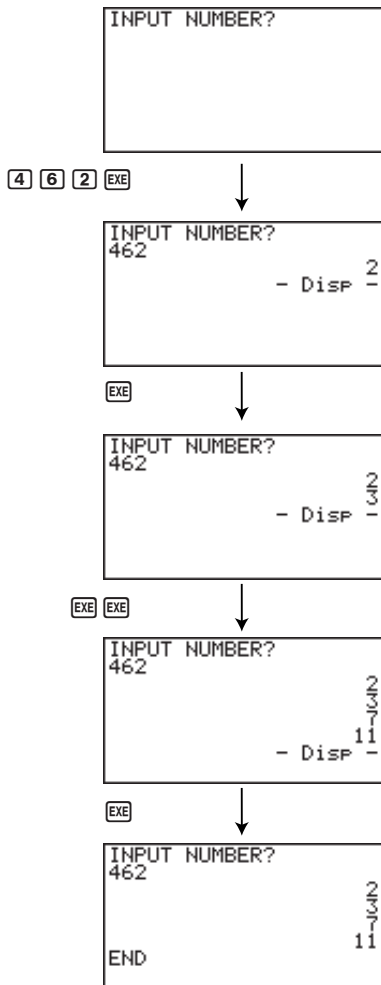


Ejemplo

$$440730 = 2 \times 3 \times 3 \times 5 \times 59 \times 83$$

```
ClrText↵
"INPUT NUMBER" ?→A↵
2→B↵
Do↵
While Frac (A/B)=0↵
B↵
A/B→A↵
WhileEnd↵
If B=2↵
Then 3→B↵
Else B+2→B↵
IfEnd↵
LpWhile B≤A↵
"END"
```





Nombre de programa	Diferenciación de secuencia aritmética-geométrica
---------------------------	--

Descripción

Luego de ingresar los términos en secuencia 1, 2 y 3, este programa determina si una secuencia es una secuencia aritmética o geométrica basado en las diferencias y relaciones de términos.

Propósito

Este programa determina si una secuencia específica es una secuencia aritmética o geométrica.

● ● ● ● ●

Ejemplo 1 5, 10, 15, ... Secuencia aritmética

● ● ● ● ●

Ejemplo 2 5, 10, 20, ... Secuencia geométrica

```

ClrText↓
"A1"→A↓
"A2"→B↓
"A3"→C↓
B-A→D↓
C-B→E↓
If D=E↓
Then ClrText↓
"AN = A1 + (N-1)D"↓
" "↓
"A1 ="↓
"D ="↓
Locate 6,3,A↓
Locate 6,4,D↓
IfEnd↓
B/A→F↓
C/B→G↓
If F=G↓
Then ClrText↓
"AN = A1×r^(N-1)"↓
" "↓
"A1 ="↓
"r ="↓
Locate 6,3,A↓
Locate 6,4,F↓
IfEnd↓
"END"

```



Ejemplo 1

```
A1?
```

5 EXE



```
A1?  
5  
A2?
```

1 0 EXE



```
A1?  
5  
A2?  
10  
A3?
```

1 5



```
A1?  
5  
A2?  
10  
A3?  
15
```

EXE



```
AN = A1 + (N-1)D  
A1 = 5  
D = 5  
- Disp -
```

Ejemplo 2

```
A1?
```

5 EXE



```
A1?  
5  
A2?
```

1 0 EXE



```
A1?  
5  
A2?  
10  
A3?
```

2 0



```
A1?  
5  
A2?  
10  
A3?  
20
```

EXE



```
AN = A1*r^(N-1)  
A1 = 5  
r = 2  
END
```



Nombre de programa

Elipse

Descripción

Este programa visualiza una tabla numérica de los valores siguientes basados en el ingreso del foco de una elipse, la suma de la distancia entre el foco y lugar geométrico, y el intervalo (tamaño de extensión) de X.

- Y1: Valores de coordenada de mitad superior de elipse.
- Y2: Valores de coordenada de mitad inferior de elipse.
- Y3: Distancia entre el foco derecho y lugar geométrico.
- Y4: Distancia entre el foco izquierdo y lugar geométrico.
- Y5: Suma de Y3 y Y4.

Luego, el programa marca los puntos del foco y los valores en Y1 y Y2.

Propósito

Este programa también muestra que las sumas de las distancias entre el lugar geométrico y dos focos de un elipse son iguales.

```

Do↓
ClrText↓
"FOCUS (C,0),(-C,0)"↓
"C="?→C↓
"SUM DISTANCE"?→D↓
LpWhile 2Abs C≥D Or D≤0↓
D/2→A↓
√(A²-C²)→B↓
Y=Type↓
"B√(1-X²/A²)"→Y1↓
"-Y1"→Y2↓
"√((X-C)²+Y1²)"→Y3↓
"√((X+C)²+Y1²)"→Y4↓
"Y3+Y4"→Y5↓
For 1→E To 20↓
If E≤5↓
Then G SelOn E↓
Else G SelOff E↓
IfEnd↓
Next↓
-Int A→F Start↓
Int A→F End↓
"F pitch"?→F pitch↓
DispF-Tbl↓
ClrGraph↓
1.2A→Xmax↓
-1.2A→Xmin↓
1.2B→Ymax↓
-1.2B→Ymin↓
G SelOff 3↓
G SelOff 4↓
G SelOff 5↓
DispF-Tbl↓
DrawFTG-Plt↓
PlotOn C,0↓
PlotOn -C,0↓
"END"

```

```
FOCUS (C,0),(-C,0)
C=?
3
```



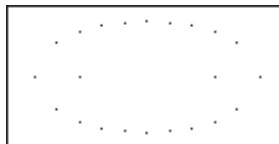
```
FOCUS (C,0),(-C,0)
C=?
3
SUM DISTANCE?
10
```



```
FOCUS (C,0),(-C,0)
C=?
3
SUM DISTANCE?
10
F pitch?
1
```



X	Y1	Y2	Y3
-5	0	0	8
-4	2.4	-2.4	7.4
-3	3.2	-3.2	6.8
-2	3.666	-3.666	6.2



Nombre de programa

Rotación

Descripción

Este programa delinea un ángulo en la coordenada definida por un vértice de ingreso, y luego lo rota en un ángulo especificado alrededor del vértice.

Propósito

Este programa demuestra la transformación de coordenada usando una matriz.

¡Importante!

Para este programa como unidad angular deberá ajustarse grados.

```

Do↓
ClrText↓
"VERTEX NUMBER"?→A↓
LpWhile A≤0 Or Frac A≠0↓
{2,A}→Dim Mat A↓
ClrGraph↓
For 1→B To A↓
Text 1,1,"VERTEX"↓
Text 1,30,B↓
If B=1↓
Then Plot ↓
PlotOn X,Y↓
X→Mat A[1,B]↓
Y→Mat A[2,B]↓
Else Plot C,D↓
F-Line C,D,X,Y↓
X→Mat A[1,B]↓
Y→Mat A[2,B]↓
IfEnd↓
Mat A[1,B]→C↓
Mat A[2,B]→D↓
Next↓
Mat A[1,1]→E↓
Mat A[2,1]→F↓
F-Line C,D,E,F↓
Text 1,1,"--AXIS--"↓
Plot ↓
PlotOn X,Y↓
X→C↓
Y→D↓
A→Dim List 1↓
A→Dim List 2↓
Fill(C,List 1)↓
Fill(D,List 2)↓
List→Mat(List 1,List 2)↓
Trn Mat Ans→Mat C↓
Mat A→Mat C→Mat A↓
ClrText↓
"ANGLE"?→E↓
[[cos E,-sin E][sin E,cos E]]→Mat B↓
Mat B×Mat A→Mat D↓
Mat D+Mat C→Mat D↓
If A=1↓
Then PlotOn Mat D[1,1],Mat D[2,1]↓
Else For 1→B To A-1↓
Mat D[1,B]→F↓
Mat D[2,B]→G↓
Mat D[1,B+1]→H↓
Mat D[2,B+1]→I↓
F-Line F,G,H,I↓
Next↓
If A>2↓
Then Mat D[1,1]→F↓
Mat D[2,1]→G↓

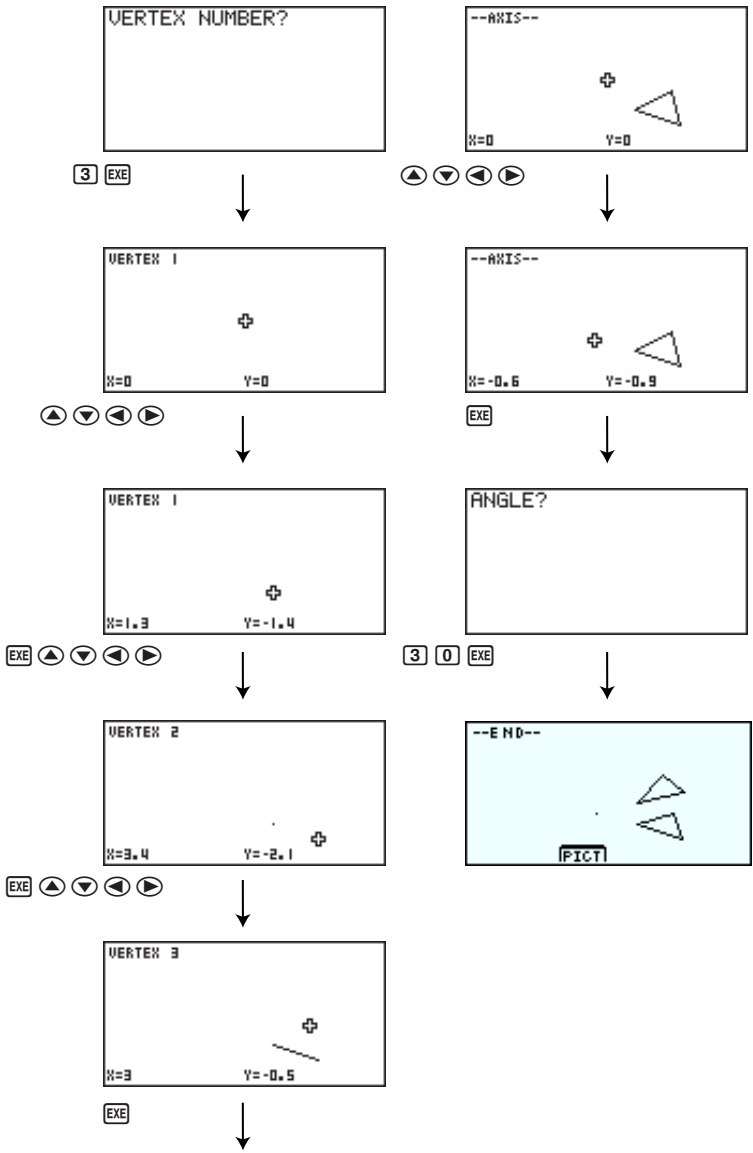
```

```

F-Line H,I,F,G↓
IfEnd↓
IfEnd↓
Text 1,1,"--END--"

```





Nombre de programa

Ángulos interiores y área de superficie de un triángulo

Descripción

Este programa calcula los ángulos interiores y el área de superficie de un triángulo definido por las coordenadas de ingreso para los ángulos A, B y C.

Propósito

Este programa calcula los ángulos interiores y el área de superficie de un triángulo definido por las coordenadas de ingreso para los ángulos A, B y C.

¡Importante!

El ingreso de las mismas coordenadas para cualquier par de ángulos (A, B, C) produce un error.

```

ClrText
"WHICH ANGLE?"
" 1. Deg"
" 2. Rad"
" 3. Gra"
Do
Getkey
LWhile ((Ans=72) Or (Ans=62) Or (Ans=52))=0
If Ans=72
Then 1→θ
Deg
" "
"-Deg-Deg-Deg-Deg-Deg-"
IfEnd
If Ans=62
Then 2→θ
Rad
" "
"-Rad-Rad-Rad-Rad-Rad-"
IfEnd
If Ans=52
Then 3→θ
Gra
" "
"-Gra-Gra-Gra-Gra-Gra-"
IfEnd
"AX"→A
"AY"→B
"BX"→C
"BY"→D
"CX"→E
"CY"→F
A-C→G
B-D→H
C-E→I
D-F→J
E-A→K
F-B→L
-GI-HJ→M
-IK-JL→N
-KG-LH→O
√(G²+H²)→P
√(I²+J²)→Q
√(K²+L²)→R
M/PQ→S
N/QR→T
O/PR→U
cos⁻¹ S→V
cos⁻¹ T→W
cos⁻¹ U→X
PQ√(1-S²)→Y
ClrText
" <ABC ="
Locate 9,1,V

```

```

" <ACB ="
Locate 9,2,W
" <BAC ="
Locate 9,3,X
If θ=1
Then " (Deg)"
IfEnd
If θ=2
Then " (Rad)"
IfEnd
If θ=3
Then " (Gra)"
IfEnd
" AREA ="
Locate 9,5,Y/2
" "
"END"

```



```
WHICH ANGLE?  
1. Des  
2. Rad  
3. Gra
```

1



```
2. Rad  
3. Gra  
-Des-Des-Des-Des-Des-  
AX?
```

0 EXE 0 EXE



```
AX?  
0  
AY?  
0  
BX?
```

1 EXE 0 EXE



```
0  
BX?  
1  
BY?  
0  
CX?
```

0 EXE 3



```
1  
BY?  
0  
CX?  
0  
CY?  
13
```

EXE



```
<ABC = 60  
<ACB = 30  
<BAC = 90  
                  (Dea)  
AREA = 0.8660254038  
END
```

