



Computadora  
**Spectrum**

Manual de uso



Basado en el Manual de Uso  
de ZX Spectrum Plus

Autor: Neil Ardley

Copyright © 1984 por  
Sinclair Research Ltd.  
y Dorling Kindersley Ltd.  
Londres - Inglaterra

**CZ SPECTRUM**

**MANUAL DE USO**

**CONTENIDO**

**Primeros Pasos** **pág. 3**

---

**Programando su computadora Spectrum** **pág. 17**

---

**Aprendiendo sobre su Spectrum** **pág. 41**

---

**Nociones de Basic Spectrum** **pág. 49**

---

# COMO UTILIZAR ESTE LIBRO

---

Esta guía para su Spectrum contiene cuatro capítulos, cada uno de ellos identificado por un color. Para remitirse a uno de los capítulos bastará con dirigirse a la sección del color correspondiente.

---

## 1 PRIMEROS PASOS

- Como organizar su sistema Spectrum • Sintonizando la televisión
  - Como solucionar problemas • Cuáles son las capacidades de su Spectrum? • Como utilizar programas de computadora comerciales
  - El procedimiento para cargar un programa • Preguntas y respuestas acerca de los problemas que puedan producirse al cargar un programa
- 

## 2 PROGRAMANDO SU COMPUTADORA SPECTRUM

- El teclado —el panel de controles de su computadora • Como operar las teclas • Una calculadora en su televisor • Colores, como emplearlos
  - Gráficas sencillas • Dibujando con su computadora • Como diseñar sus propios dibujos • El procedimiento para crear caracteres
    - Dibujos animados • Un instrumento musical y efectos sonoros
  - Como grabar sus programas • Soluciones para los problemas que puedan producirse al grabar programas
- 

## 3 APRENDIENDO ACERCA DE SU SPECTRUM

- Cuáles son sus componentes? • Cómo funciona? • El procedimiento para conectar equipos accesorios • Información técnica
- 

## 4 NOCIONES ACERCA DEL LENGUAJE BASIC DE SPECTRUM

- Guía de referencia para el programador utilizando las palabras clave del BASIC Spectrum • Su Spectrum se comunica con usted mediante mensajes en la pantalla del televisor • Más allá de los confines del BASIC
  - La terminología de las computadoras: un breve diccionario

# 1. PRIMEROS PASOS

---

El propósito de este capítulo es explicarle como comenzar a explorar el potencial de su computadora Spectrum. Le indicaremos como armar y organizar su sistema para que pueda entrar en acción cuando usted se lo indique. Una vez completada esta etapa usted podrá elegir entre dos caminos. La primera posibilidad es ingresar algunos programas en su computadora Spectrum, lo cuales demostrarán su operación, incluyendo las capacidades del color y sonido, o, alternativamente, usted puede preferir aprender a utilizar programas comerciales, listos para ser procesados, tales como juegos de computadora ("computer games"). Cualquiera sea su elección, usted tendrá la oportunidad de disfrutar plenamente de las posibilidades ofrecidas por su nueva computadora.

## COMO ORGANIZAR SU SISTEMA SPECTRUM

El procedimiento para organizar su sistema Spectrum es el siguiente: en primer lugar verifique que tenga disponibles todos los elementos enumerados en la lista de control a continuación. En segundo lugar siga las

instrucciones detalladas en la página opuesta.

Conecte todos los componentes seguramente. Recuerde que si durante la operación de su Spectrum usted desconecta la electricidad o desenchufa la computadora, se perderán todos los programas, información o resultados almacenados en la memoria de la computadora.

Después de haber terminado de trabajar con su computadora, apáguela y desenchúfela.

### Lista de control: Tiene todos los elementos necesarios?

Al desembalar su computadora usted encontrará:

- 1 Su computadora Spectrum.
- 2 Su fuente de alimentación. Esta produce el suministro de 9 volt CC requeridos por la computadora Spectrum.
- 3 Cable de conexión a la antena de su televisor. Conecta su Spectrum con la televisión.
- 4 Cable del grabador de cassette. Conecta su Spectrum con el grabador de cassette.
- 5 La Tarjeta de Garantía. Complete la Tarjeta y remítala como se indica en ella.
- 6 Programa de introducción al uso de la computadora en cassette.
- 7 Programa de demostración sobre usos más frecuentes para su computadora en cassette.
- 8 Dos patas de plástico que le permitirán inclinar la computadora para mayor comodidad de uso. Colóquelas a presión sobre los apoyos de goma que lleva la máquina en su base.

- 9 En la parte exterior hay un espacio para alojar este manual. Bajo la máquina, a la derecha, hay una cavidad para guardar hasta cuatro cassettes adicionales. Usted deberá suministrar los

elementos siguientes:

- 1 Una televisión.
- 2 Un grabador de cassettes.
- 3 Un enchufe eléctrico para conectar el sistema a la red de electricidad.



### Organizando su equipo: preguntas y respuestas

**Es necesario que el televisor sea de colores?**

No, no es necesario, aunque si su televisor es blanco y negro entonces usted no podrá apreciar los colores producidos por su Spectrum.

**Sirve cualquier tipo de televisor?**

En principio su Spectrum le ofrecerá una imagen en cualquier televisor. Si éste no es el caso, la causa del problema puede ser que la computadora y el televisor tengan diferentes sistemas para generar las imágenes. Esto puede suceder si su televisor es muy antiguo, o si el televisor y su Spectrum han sido adquiridas en

países diferentes. En caso de duda le aconsejamos que consulte a su vendedor de televisores.

**Es posible emplear un monitor en lugar de un televisor?**

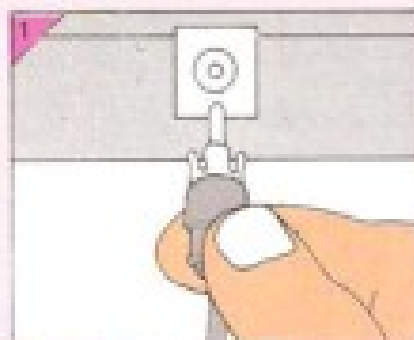
Si, es posible que el vendedor de la computadora también pueda suministrarle un monitor, el cual le ofrecerá una mejor imagen para su Spectrum.

**Puede utilizar el expansor de 16K de RAM?**

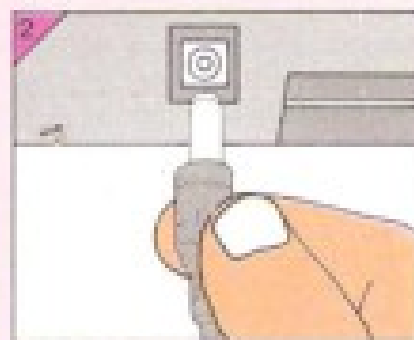
No. Este accesorio RAM (o "RAM pack") solamente puede ser utilizado con la computadora 1000 ó 1500.

## Conectando su Spectrum a la red eléctrica

Primero instale un enchufe eléctrico en el cable de suministro de electricidad de su computadora. En algunos países será necesario instalar un fusible de 3A en el enchufe eléctrico. Observe que su Spectrum no requiere una conexión a tierra, aun si el enchufe que usted instale sea de tres patas. A continuación, complete las siguientes etapas, siguiendo las secuencias de las ilustraciones, para conectar su Spectrum a la red eléctrica y su televisor. Después de haber conectado debidamente el sistema, diríjase a la página siguiente, donde encontrará las instrucciones para sintonizar.



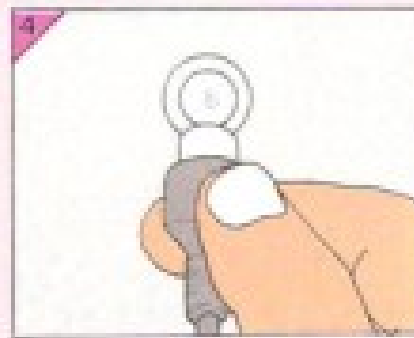
Insertar el cable del televisor en el enchufe marcado TV de su Spectrum. Solamente uno de los enchufes en el cable del televisor se ajustará a este punto de conexión.



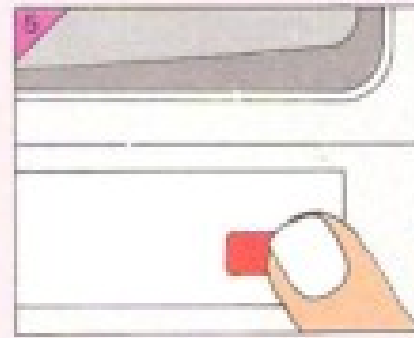
Inserte el pequeño enchufe del cable de alimentación de electricidad en el enchufe marcado 9VDC de su Spectrum.



Conecte la fuente a la red domiciliaria. Recuerde que su computadora Spectrum no tiene llave de encendido (ON/OFF).



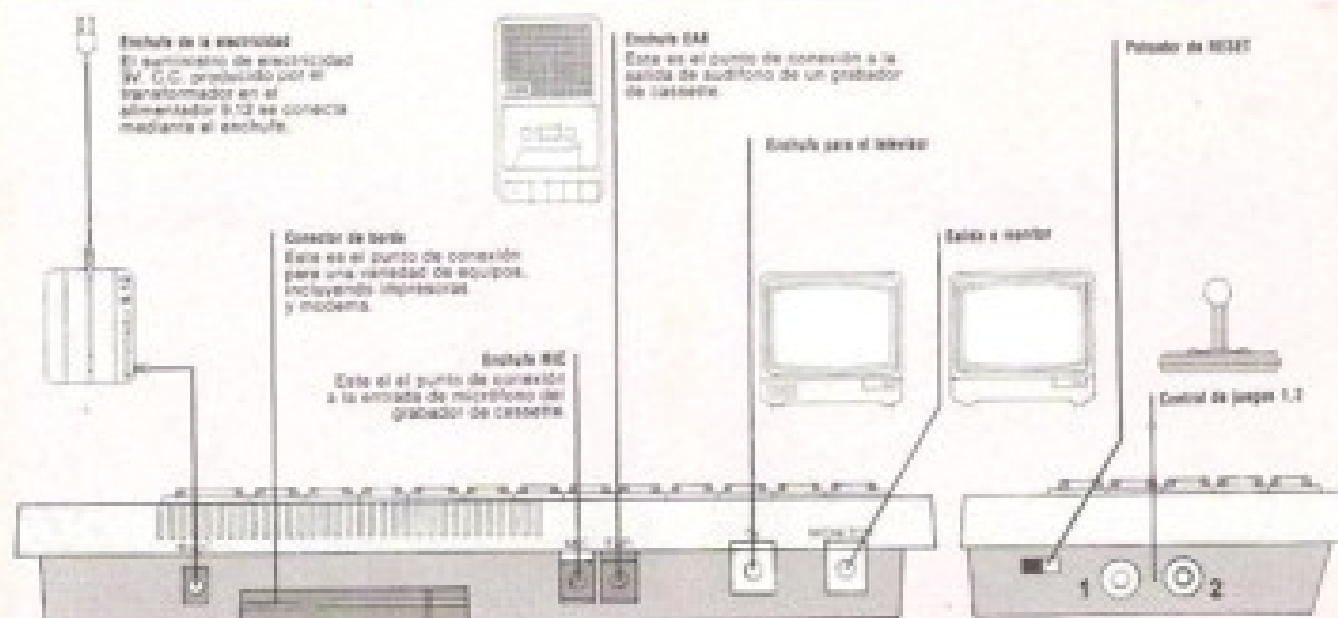
Retire el cable que conecta la antena de su televisor. Ahora conecte el cable del televisor de su Spectrum, en el conector de la antena de su televisor.



Conecte su televisor y baje su volumen al mínimo.

Ahora todo está pronto para proceder a la siguiente etapa: sintonizar el televisor para que reciba las señales de su Spectrum.

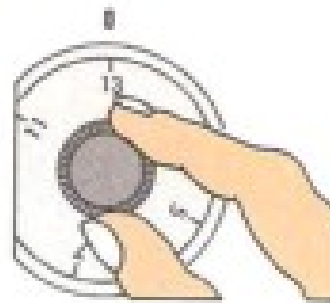
## Los enchufes y conectores de la Spectrum



## SINTONIZANDO SU TV

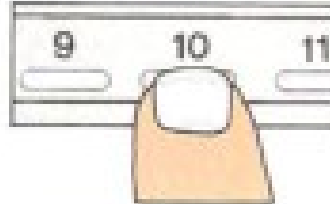
Su Spectrum emite una señal de video de televisor en colores en las frecuencias correspondientes al canal 12 de la banda de VHF. Por lo tanto, es necesario sintonizar su televisión en este canal para que proyecte la imagen generada por la computadora.

Después de haber conectado su computadora al televisor, la siguiente etapa es sintonizar el televisor hasta que la pantalla de este último muestre el mensaje de Spectrum, tal como en la ilustración. Si no puede recibir esta imagen en su televisor, o si los colores no están bien, le sugerimos que lea las instrucciones a continuación.

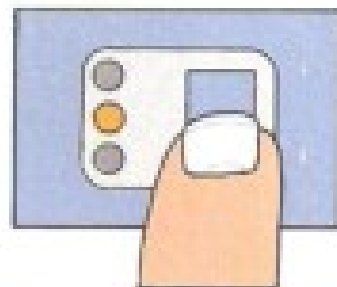
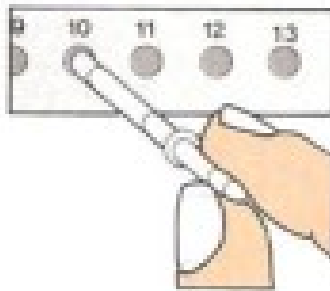


**Tipos de controles de sintonización**

**Perilla Selectora de Canales**  
Conecte el canal 12. Luego accione la perilla de sintonía fina hasta obtener el mensaje de copyright en forma nítida.



**Sintonización de botón**  
Seleccione uno de los botones de sintonización para ser utilizado como el botón de sintonización de su computadora. Ajuste el botón hasta obtener el mensaje en la pantalla. Si fuera posible elija un botón que no sea utilizado para sintonizar canales de televisión, de esta forma evitará tener que reajustarlo cada vez que desee utilizar su televisor para operar la computadora.



**Sintonizador a botones**  
Este método de sintonización es similar al de botón, excepto que el equipo se ajusta automáticamente en el canal seleccionado.

### Como verificar los colores de la Spectrum

Para verificar los colores de la Spectrum basta con oprimir la tecla B y a continuación un número entre 1 y 6. Ahora se producen los siguientes eventos: (1) el mensaje desaparece de la pantalla; (2) aparece la palabra BORDER (borde) y el número correspondiente. Ahora oprima la tecla ENTER. El área del "borde" de la pantalla debe

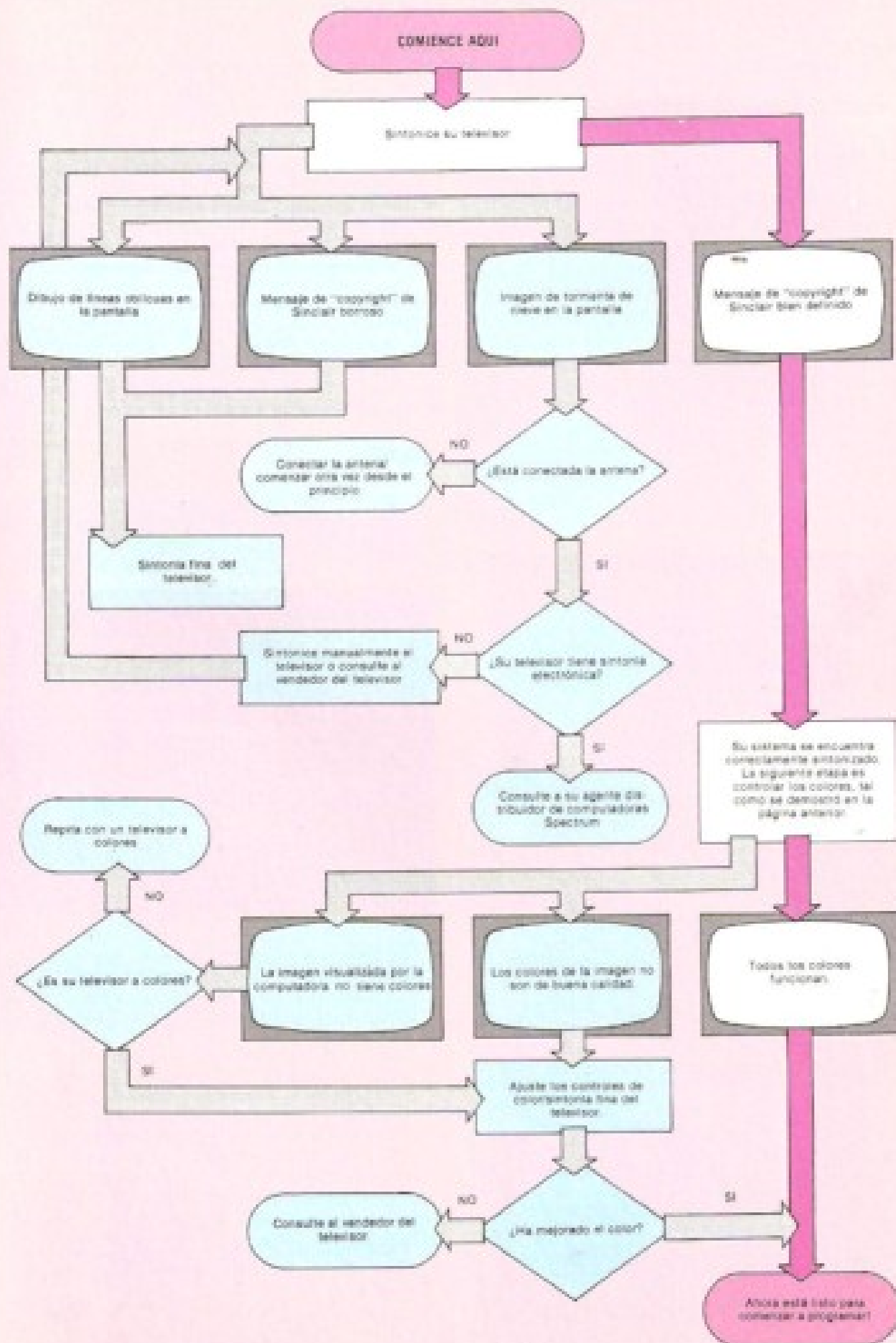
cambiar al color indicado en la tecla-número seleccionada. Las pantallas en las ilustraciones a continuación muestran que cuando usted ingresa en la computadora (en el orden indicado) las instrucciones BORDER 4 —ENTER, BORDER 3—ENTER, y finalmente BORDER 7, el color del borde cambia a blanco.

**BORDER 3**



**BORDER 4**





# CUALES SON LAS CAPACIDADES DE SU SPECTRUM

## Primero, algunos experimentos

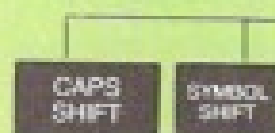
Ahora que su Spectrum está conectada y el televisor ha sido sintonizado correctamente, haga la prueba de oprimir algunas de las teclas de la computadora. Como puede apreciar, inmediatamente aparecen palabras y letras en la pantalla del televisor y también algunos números.

Sin embargo, usted debe saber como programar la Spectrum para darle instrucciones sobre qué hacer. Pero, no tenga miedo, pruebe las teclas sin temor.

## Como ingresar información o instrucciones en su computador

Para ingresar cualquier palabra, letra o número en su computadora, primero tome nota su posición sobre la tecla y a continuación utilice la misma secuencia de teclas selectoras que se indica a continuación:

Primero oprima estas 2 teclas juntas y luego la tecla deseada



Oprima la tecla junto con SYMBOL SHIFT

SYMBOL SHIFT

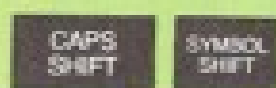
Oprima la tecla. Utilice la tecla CAPS SHIFT para elegir mayúsculas

CAPS SHIFT



Oprima la tecla

Primero oprima estas dos teclas juntas, luego la tecla deseada junto con SYMBOL SHIFT.



SYMBOL SHIFT

Los detalles sobre como operar las teclas pueden verse en las páginas 20-21.

## A continuación: cómo programar su Spectrum

Su Spectrum puede hacer muchas cosas. Pero para hacerla funcionar es necesario que usted le comunique una serie de instrucciones, que se denominan programa de computadora.

Los siguientes son una colección de programas breves que le demostrarán algunas de las capacidades de su Spectrum, incluyendo colores, sonidos y gráficas.

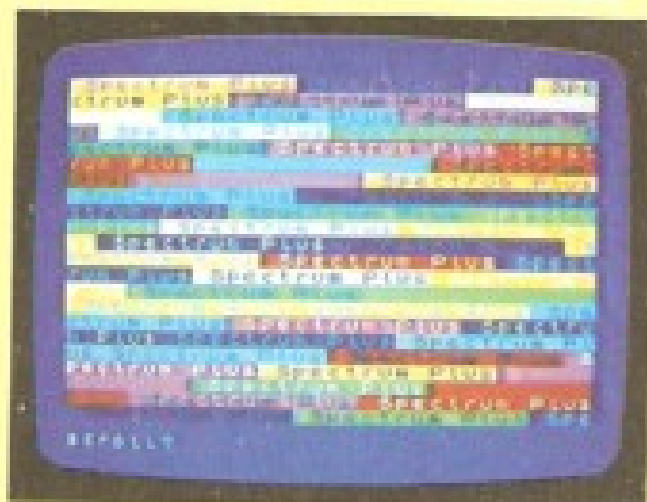
Todo lo que usted debe hacer es ingresar el programa en la computadora exactamente como se indica. Las imágenes en la pantalla de el televisor le indican el resultado de sus operaciones con el teclado. Si desea experimentar con los programas, puede hacerlo siguiendo las instrucciones en la página opuesta sobre *Cómo alterar un programa*.

## Como ingresar y ejecutar un programa

Cada serie de instrucciones es mostrada en una lista, denominada listado. Los listados de los programas contienen varias secciones, cada una de las cuales comienza con un número, por ejemplo 10, 20, etc. Cada una de estas secciones es

## NOMBRES

```
10 BORDER 1: INK AND+7
20 PAPER AND+7
30 PRINT "SPECTRUM +",
40 GO TO 10
```



El nombre Spectrum aparece en la pantalla en una variedad de colores. La computadora se detiene y le pregunta mediante un mensaje en la parte inferior de la pantalla: *scroll?* (en inglés "scroll" significa rebobinar). Para desplazar la imagen en la pantalla hacia arriba, bastará con oprimir cualquier tecla excepto: N, BREAK o STOP. Si usted detiene el movimiento de la pantalla y oprime las teclas (en el orden indicado): primero BREAK, luego R (RUN) seguida de ENTER, los nombres aparecerán en un patrón de colores diferentes.

## Otro experimento

En la línea del programa sustituya "Spectrum Plus" por su nombre entre paréntesis, por ejemplo

```
30 PRINT "JUAN";
```

No se olvide de incluir el punto y coma (;).

denominada línea del programa.

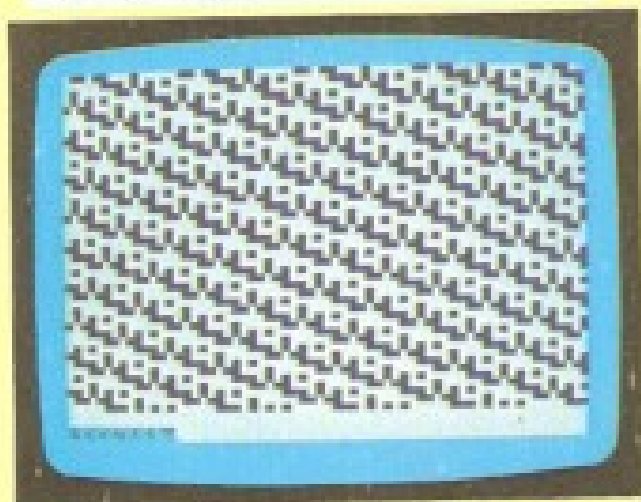
En cada una de las líneas del programa, se encontrarán palabras completas o abreviaciones compuestas de dos o más letras, tales como PRINT, RND, PI, PAPER, LED y GOTO. Estas son denominadas palabras claves y no es necesario ingresarlas letra por letra. Bastará utilizar las teclas correspondientes, cada una de las cuales tiene escrita una palabra clave (por ejemplo la palabra clave PRINT se encuentra sobre la tecla P). Para ingresar una de estas instrucciones busque la palabra clave correspondiente en el teclado y siga las instrucciones en la sección sobre cómo ingresar información a la computadora en el capítulo 2.

A medida que usted ingresa la información correspondiente a una línea, mediante el teclado, esta va apareciendo en la parte inferior de la pantalla del televisor. Al terminar la línea del programa, oprima la tecla ENTER. La línea cambiará de posición, desplazándose a la parte superior de la pantalla. Repetir el mismo procedimiento para las demás líneas. Si comete un error, consulte la sección sobre Cómo corregir errores en la página siguiente.

Después de haber ingresado todas las líneas del programa en su computadora oprima la tecla R. La palabra clave RUN aparece en la pantalla. Ahora oprima la tecla ENTER y su Spectrum entrará en acción, ejecutando el programa.

## FIGURAS

```
10 LET A=1
20 FOR K=1 TO 7
30 LET A=18+OHMS (RND*14+120)
40 NEXT K
50 INK RND*7
60 BORDER RND*7
70 PRINT A$
80 GO TO 10
```



Este programa genera una serie de figuras de colores. Cuando la pantalla está llena, el proceso se detiene y aparece el mensaje "scroll?". Si desea continuar oprima cualquier tecla (excepto N, SPACE, BREAK o STOP) para desplazar la imagen hacia arriba. Para ver un nuevo tipo de dibujo con una combinación de colores diferente, cuando su computadora le pregunte "scroll?" ingrese las siguientes instrucciones en el orden indicado: oprima la tecla N, luego BREAK, seguido por R (RUN) y finalmente ENTER.

### Pruebe esto

En la línea 20 cambie el número 7 por otro número, para obtener un patrón diferente.

## Cómo alterar un programa

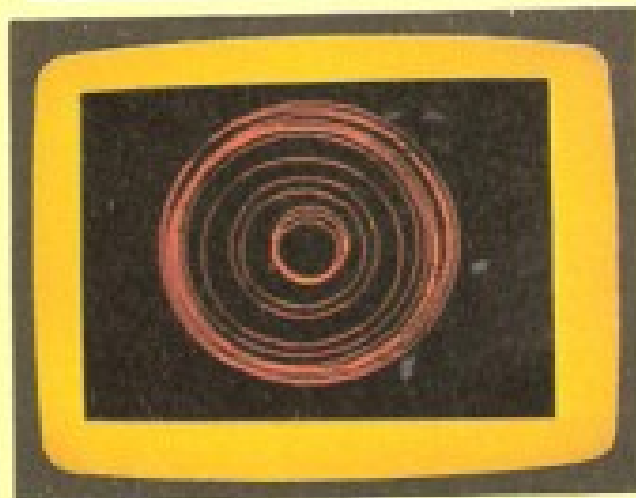
Espere hasta el final del programa o interrúmpalo mediante la instrucción BREAK. A continuación oprima V (CLS), seguida de ENTER, K (LIST) y finalmente de ENTER. Como resultado de estas instrucciones el listado de programa aparecerá en la pantalla.



Estudie el listado y seleccione la línea del programa que desea modificar, ingrese la línea del programa incorporando las nuevas instrucciones o información que desea incluir por el teclado, incluyendo su número, y finalmente oprima ENTER. La nueva línea aparecerá en el listado. Oprima R (RUN) y ENTER, el nuevo programa entrará en operación.

## CIRCULOS LUMINOSOS

```
10 BORDER 0: PAPER 0: CLS
20 CIRCLE INK RND*5: FLASH RND
30 RND*5: RND*5: RND*5: RND*5
40 BEEP 0:1: RND*50
50 IF RND*5 THEN GO TO 60
60 GO TO 20
70 FOR Y=-5 TO 4
80 FOR X=0 TO 5
90 BORDER X
100 BEEP .05:Y*Y
110 NEXT X
120 NEXT Y
130 RUN
```



El programa genera una serie de círculos luminosos casi concéntricos de una variedad de colores. De pronto, el borde cambia de colores, la computadora produce un sonido y aparece un nuevo juego de círculos.

### Pruebe esto

Después de listar el programa (LIST), ingrese la instrucción PAPER 7 y oprima la tecla ENTER. A continuación ingrese una nueva línea 20, pero sin las palabras claves FLASH RND. Como resultado los círculos no brillarán.

MOSAICO LOCO

```
1000 REM ***** MOSAICO LOCO *****
1010 CLS
1020 LET Y=0
1030 FOR X=0 TO 20
1040   FOR Y=0 TO 20
1050     LET V=INT(RND*16)
1060     LET C=V/4
1070     LET R=V/4
1080     LET G=V/4
1090     LET B=V/4
1100     PRINT USING "##";C;R;G;B;
1110   NEXT Y
1120 NEXT X
1130 GOTO 1030
```



Un cuadrado de colores se mueve en la pantalla desarrollando un patrón de colores. Cada vez que el programa es reiniciado el patrón producido es diferente.

Otro experimento

Cambie el número 143 por 42, en la línea 50 del programa, y el cuadrado se transformará en estrellas! Pruebe otros números entre 33 y 142. Consulte la tabla de juego de caracteres en la página 51 para ver que sucederá.

Como reiniciar un programa

Algunos de estos programas, por ejemplo Ramas y Estrellas, llegan al final y producen el reporte @ OK y el número de la última línea del programa. Esto sirve para indicarle que la computadora ha ejecutado la totalidad del programa. Para reiniciarlo solamente se requiere oprimir R (RUN) y ENTER.

Otros programas continúan operando, por ejemplo, el del Mosaico loco, o se reinician automáticamente como Amanecer.



Para detener estos programas, oprima simultáneamente CAPS SHIFT y SPACE (BREAK). Oprima esta tecla hasta que el programa se interrumpa y aparezca el informe BREAK. Para reiniciar el programa oprima R (RUN) y ENTER

Como corregir errores

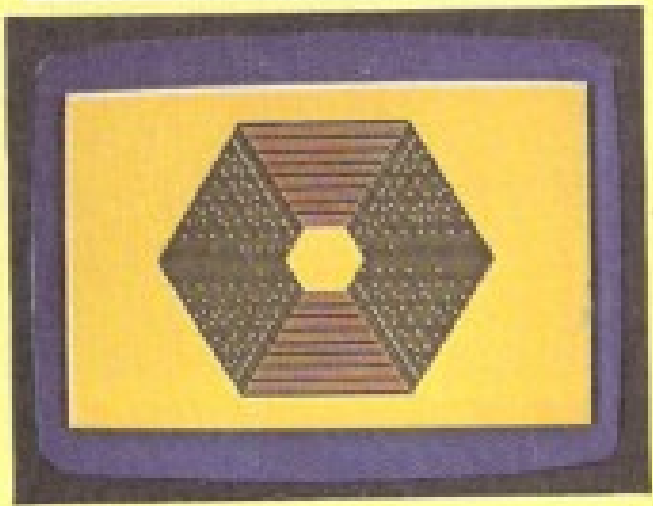
Si oprimió una tecla equivocada, o se equivocó al oprimir la combinación CAPS SHIFT - SYMBOL SHIFT, no se preocupe. Oprima simultáneamente CAPS SHIFT y @ (DELETE) y la última palabra clave, signo, letra o número desaparecerá. Siga oprimiendo estas dos teclas para borrar más.



Si cometió un error después de oprimir ENTER, en ese lugar aparecerá un signo de interrogación. Oprima DELETE para borrar la línea hasta el signo de interrogación y a continuación complete la línea correctamente. Finalmente oprima ENTER. Es posible que usted ingrese una línea con errores. En este caso la computadora se detendrá y producirá un informe de error. El informe (abajo de su pantalla) le indicará cual es la línea incorrecta. A continuación usted debe ingresar nuevamente toda la línea correctamente y oprimir las teclas ENTER, R(RUN) y ENTER. El programa debería ejecutarse sin problemas.

POLIEDROS

```
1000 REM ***** POLIEDROS *****
1010 CLS
1020 LET N=5
1030 FOR I=0 TO N
1040   FOR J=0 TO N
1050     LET V=INT(RND*16)
1060     LET C=V/4
1070     LET R=V/4
1080     LET G=V/4
1090     LET B=V/4
1100     PRINT USING "##";C;R;G;B;
1110   NEXT J
1120 NEXT I
1130 GOTO 1030
```



Al principio, este programa, produce una pantalla vacía. Oprima el número 5 seguido de ENTER. Se generará una figura de seis lados. Reinicie el programa y esta vez ingrese otro número para crear una figura diferente.

Pruebe esto

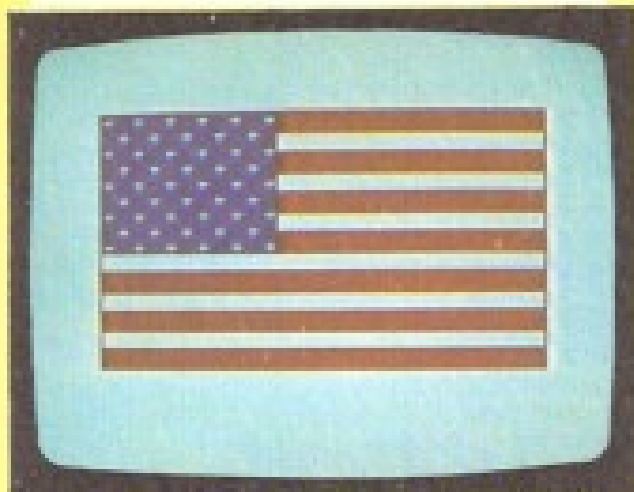
En la línea número 20, del programa cambie el número 2 por otro número. La computadora aumenta su velocidad si el número es más alto y los poliedros se separarán.

## FRANJAS Y ESTRELLAS

```

10 DEF FN C=0
20 PAPER 7
30 CLS
40 FOR X=10 TO 140 STEP 20
50 FOR Y=10 TO 110
60 PLOT 10+X, Y DRAW 210,0
70 NEXT Y
80 NEXT X
90 PLOT 150,20 DRAW 0,130
100 PLOT 230,20 DRAW 0,130
110 PAPER 1
120 DEF FN C=0
130 FOR X=10 TO 0 STEP -2
140 PRINT AT 1,2,"* * * * *"
150 NEXT X
160 PRINT AT 1,2,"* * * * *"
170 PRINT AT 1,2,"* * * * *"

```



Este programa genera en la pantalla la bandera de los Estados Unidos.

### Un experimento

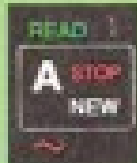
Pruebe cambiar los números de los colores de la bandera. El color de las franjas es indicado en la línea 10, el de las estrellas en la línea 120, y el del fondo de las estrellas en la línea 110.

### Como iniciar un nuevo programa

Después de haber terminado un programa desea ingresar otro totalmente nuevo. En este caso el procedimiento es esperar hasta que el programa corriente se haya completado, o, también, es posible interrumpirlo mediante el comando BREAK.

Es posible elegir entre dos procedimientos para borrar el programa antiguo de la memoria de la computadora. Un procedimiento es oprimir dos teclas: primero A (NEW) y luego ENTER. La pantalla del televisor se apagará por unos instantes y luego aparecerá el mensaje de Copyright.

BREAK  
SPACE



ENTER

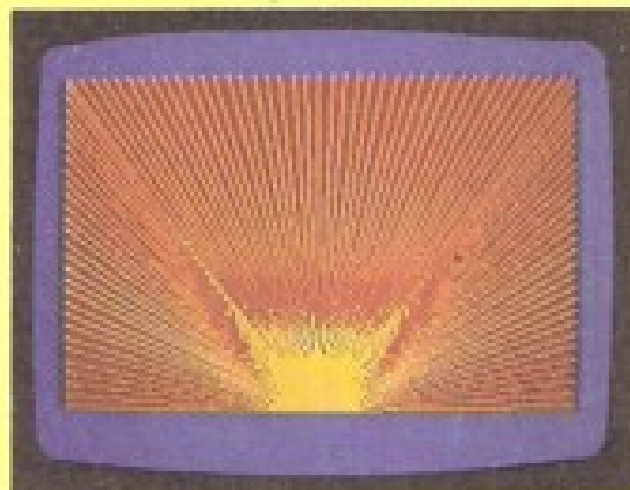
La otra manera, más fácil es desconectar el cable de alimentación (marcado 9VDC) y volver a conectarlo. Esto tiene el mismo efecto que desconectar la fuente de poder del enchufe y volver a conectarla.

## AMANECEER

```

10 PAPER 0:DEF FN C=0
20 DEF FN C=0
30 PAPER 0:DEF FN C=0
40 CLS
50 LET Y=100-10-2
60 FOR X=10 TO 170 STEP 2
70 PLOT 100,0
80 DRAW 100,0
90 DEEP .01-X/70
100 NEXT X
110 FOR X=127 TO 127 STEP 2
120 PLOT 100,0
130 DRAW 1,0
140 DEEP .01-X/70
150 NEXT X
160 FOR X=170 TO 0 STEP -2
170 PLOT 100,0
180 DRAW 1,0
190 DEEP .01-X/70
200 NEXT X
210 PAUSE 200
220 GO TO 10

```



Este programa genera un amanecer de variados colores en la pantalla. Si la pantalla queda en blanco, no se asuste, espere un poco y tendrá un nuevo amanecer.

### Experimento

Modifique el contenido de la línea 20 del programa, cambiando el número 200 por otro, para así alterar el tiempo de duración de cada amanecer. 200 es equivalente a 4 segundos.

### Cuál es la siguiente etapa?

Ahora usted puede elegir. Si desea conservar algunos de estos programas para volverlos a usar en el futuro, es posible almacenarlos, grabándolos en cintas de cassette. Para averiguar cual es el procedimiento, consulte la sección Como almacenar sus programas en la página 38.

Si desea continuar experimentando con su Spectrum, el Capítulo 2 Programando su computadora Spectrum le ofrecerá toda la información necesaria. Hasta ahora usted ha probado los programas sin realmente comprender como es que funcionan. El Capítulo 2 le explicará varios de los aspectos de la programación de su Spectrum.

Por último, si desea probar algunas cintas conteniendo programas pre-grabados le sugerimos estudiar la sección Como utilizar programas comerciales.

## COMO UTILIZAR PROGRAMAS COMERCIALES

Cuando ingresa un programa en la Spectrum desde el teclado, usted produce una secuencia de señales código electrónicas, a medida que oprime cada tecla. Estas señales son dirigidas a la memoria de la Spectrum donde son almacenadas, para ser utilizadas por la computadora después de que éste ha recibido la instrucción de ejecutar el programa. Estas instrucciones permanecerán en la memoria de la computadora hasta que usted las borre (por ejemplo, ingresando la instrucción NEW, presionando el botón de reajuste (reset) o apagando la computadora.

Sin embargo, no siempre es necesario ingresar el programa a través del teclado. Es posible comprar programas de computadora pre-grabados comerciales. Estos son alimentados directamente a la computadora en forma automática. El empleo de programas pre-grabados, listos para ser utilizados, le ahorrará el trabajo de tener que ingresar la instrucciones para su computadora a través del teclado, cada vez que desee utilizar su Spectrum.

Otra ventaja es que podrá formar su propia biblioteca de programas listos para ser utilizados, en lugar de tener que escribirlos y cargarlos en la computadora usted mismo. Los fabricantes de programas de computadora producen programas de todas clases, los cuales son escritos por los mejores programadores. Existe disponible una amplia variedad de programas para la Spectrum. Le sugerimos que examine el catálogo de programas de Spectrum para establecer cuales son los adecuados a sus necesidades.

### El procedimiento para cargar un programa en su Spectrum

Las señales código en una cinta conteniendo un programa, consisten en una serie de notas agudas o graves, registradas a una velocidad de aproximadamente 1500 notas por segundo. Cuando usted pasa una cinta magnética conteniendo un programa en su grabador, el grabador produce una secuencia de notas que componen el programa. Al conectar el grabador a la computadora Spectrum las señales código en la cinta son transmitidas directamente a la memoria de la Spectrum donde son almacenadas. Este proceso se denomina "cargar un programa".

### Preguntas acerca de los programas pregrabados.

¿Qué es un "programa pregrabado" (en inglés "software")?

"Programas pregrabados" o en inglés "software" son instrucciones que se ingresan en la computadora para hacerla operar. La expresión inglesa hardware, en cambio se refiere a la computadora y otros equipos auxiliares.

¿Por qué los programas pregrabados son producidos en cassettes?

Las cintas de cassettes son fáciles de utilizar y no requieren ningún tipo de equipo especial. Solamente se necesita un grabador de cassette para este tipo de programas.

¿Cómo suenan los programas pregrabados en cassettes?

Pase uno de sus cassettes en el grabador sin conectar ésta última a la computadora Spectrum. El sonido es inconfundible y es causado por las señales código que son transmitidas al altavoz del grabador en lugar de la computadora. Las señales son transmitidas desde el cassette a la Spectrum a una velocidad tan alta que es imposible distinguir entre los sonidos individuales.

¿Cuál es el mejor grabador de cassette?

La Spectrum puede operar correctamente con un grabador de cassettes portátil económico, preferiblemente conectado a la red eléctrica principal y no utilizando pilas. El grabador debe tener su propio control de volumen, pero no es esencial que tenga un control de tono. También existen disponibles grabadores de cassette especiales para funcionar con computadoras. Estos equipos han sido diseñados para almacenar y cargar programas con más seguridad que los equipos ordinarios.

¿Los programas grabados en cinta requieren cuidados especiales?

Como cualquier otra forma de almacenamiento mediante impulsos magnéticos, los programas en el cassette son afectados por campos magnéticos de gran intensidad. Por lo tanto, es aconsejable mantener los cassettes lejos de cualquier fuente de electricidad o equipo utilizando energía eléctrica. Los cassettes también deben ser mantenidos limpios y protegidos del polvo.

¿Es posible utilizar cualquier tipo de programas pregrabados con su Spectrum?

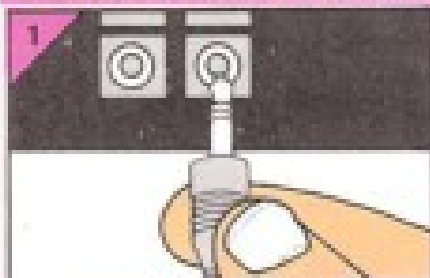
No. Su computadora solamente puede procesar programas pregrabados producidos especialmente para computadoras Spectrum.

## Conectando su grabador de cassettes

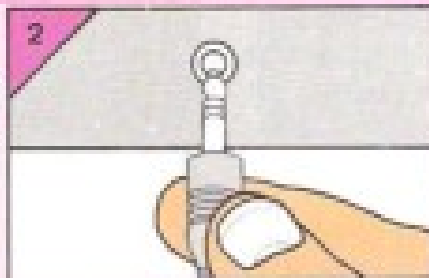
Su computadora viene con un cable especial para conectarlo al grabador de cassettes. Es el cable con los dos enchufes pequeños. Ubique al grabador de cassettes junto a su Spectrum y enchufe el cable tal como se muestra. El grabador de cassettes y la Spectrum pueden

estar encendidos o apagados durante esta operación, aunque es una buena idea verificar que el grabador no contenga ningún cassette al encender o apagar la computadora para proteger los programas grabados en la cinta.

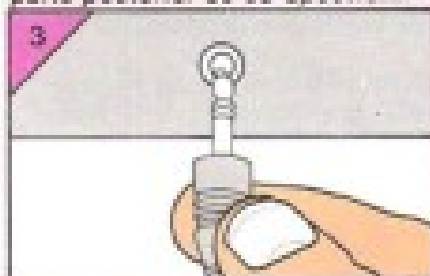
### Haciendo las conexiones correctas



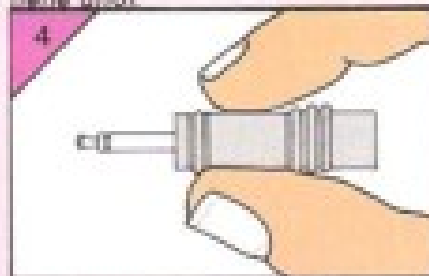
1  
Introduzca cualquiera de los dos enchufes en el punto de conexión EAR situado en la parte posterior de su Spectrum.



2  
Introduzca el otro enchufe en el punto de conexión EAR del grabador de cassettes (si este tiene uno).



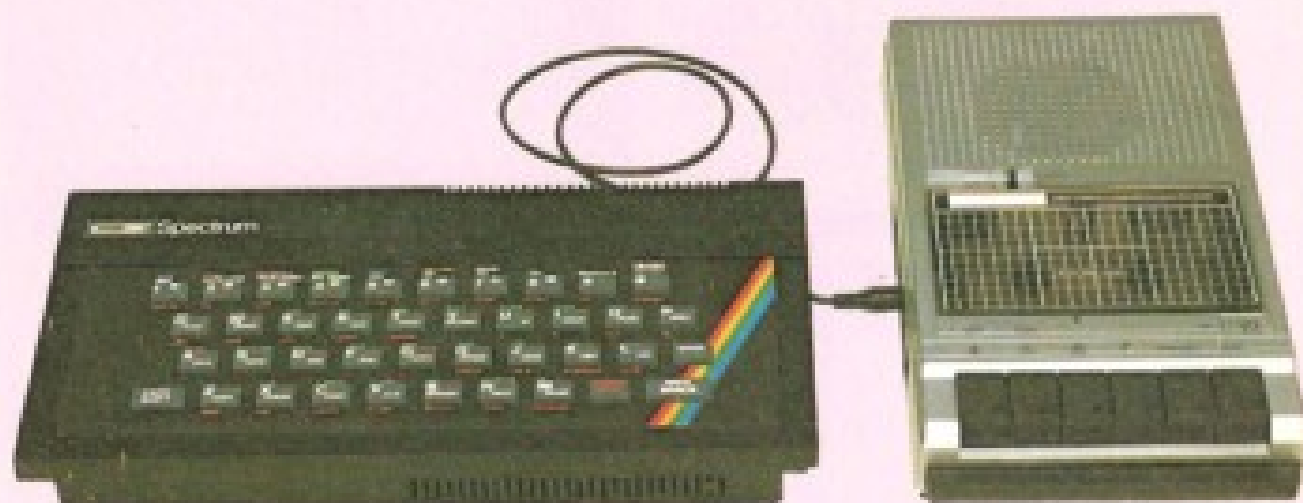
3  
Si su grabador no cuenta con un punto de conexión EAR conecte el enchufe en uno de los puntos de conexión para auriculares (si el grabador tiene uno). También puede tratar de conectarlo al punto de conexión de altavoz externo.



4  
Si el enchufe del cable de conexión del grabador no encaja en el punto de conexión necesitará un adaptador o un cable especial con los enchufes correctos. El punto de conexión EAR de la Spectrum requiere un enchufe macho de 3.5 mm y una señal de entrada de alrededor de 1 volt.

### Recuerde que:

Algunos grabadores son afectados por otros equipos eléctricos en sus cercanías, lo que puede distorsionar las señales transmitidas entre la computadora y el grabador, con el resultado de que no será posible cargar correctamente los programas. Si su grabador parece no funcionar, puede alejarlo del televisor o de la computadora.



## EL PROCEDIMIENTO PARA CARGAR UN PROGRAMA

Después de haber conectado el grabador de cassettes a su Spectrum todo se encuentra dispuesto para cargar y ejecutar un programa.

Encienda el grabador. Verifique que la Spectrum también esté debidamente enchufada y encendida. Coloque el cassette en el grabador.

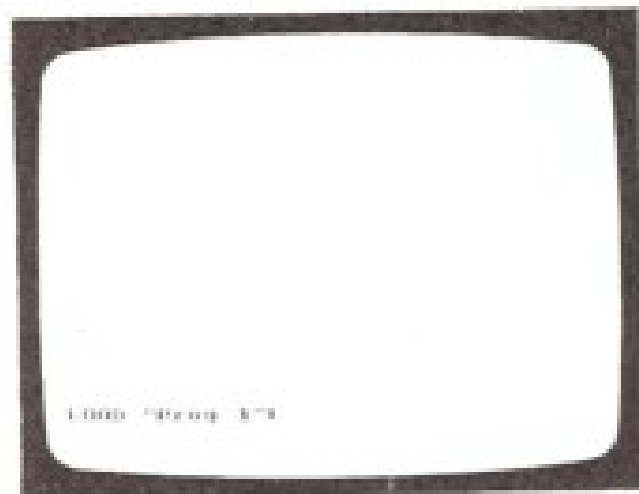
Si la computadora se encuentra en proceso de ejecutar otro programa espere hasta que sea completado o interrúmpalo, oprimiendo la tecla BREAK. A continuación ingrese en la computadora la instrucción NEW para limpiar la memoria de la Spectrum aunque esto no es necesario, porque cargar un nuevo programa tiene el efecto de borrar el contenido de la memoria.

Ahora siga las instrucciones numeradas. En caso de algún problema, diríjase a la página 16 y consulte la sección sobre **Soluciones para los problemas que pueden encontrarse al cargar programas pregrabados en la computadora.**

**1** Coloque el cassette y rebobine hasta el principio de la cinta.

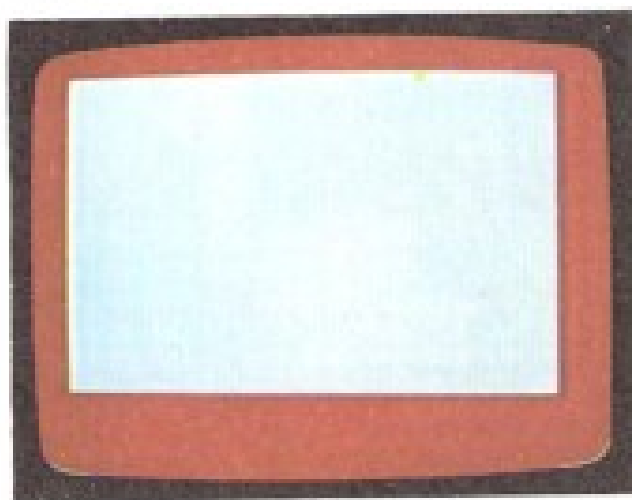
**2** Ajuste los controles de volumen y tono del grabador en los niveles requeridos. Pruebe el nivel de volumen correspondiente a dos tercios del máximo, y si su grabador cuenta con un control de tono, ajústelo al tono agudo máximo.

**3** Oprima la tecla J y el mensaje LOAD debería aparecer en la pantalla. A continuación ingrese el nombre del programa desde el teclado de la computadora entre comillas por ejemplo: LOAD "prog1"



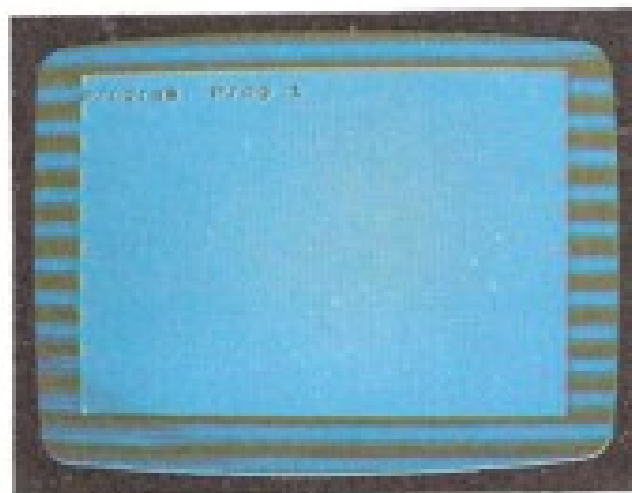
**4** Oprima la tecla ENTER. La pantalla se pondrá en blanco.

**5** Encienda el grabador. El borde de la pantalla cambiará de color, tornándose rojo o azul o una combinación de los dos colores. Esta señal le indica que su Spectrum está buscando el programa.



**6** Pocos segundos después unas franjas azules y rojas comenzarán a desplazarse a lo largo del borde. Esta señal indica que la Spectrum ha comenzado a recibir una señal.

**7** La palabra Program aparece en la pantalla, seguida por el nombre del programa o Bytes seguida de un nombre o letra. Esta señal le informa que la computadora ha ubicado el programa que usted le indica.



**8** Las franjas rojas y azules reaparecen, mientras la computadora espera para cargar el programa.

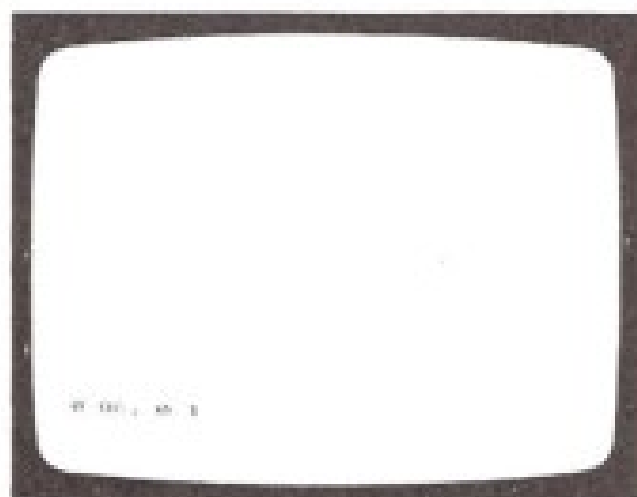
**9** Ahora se visualiza en la pantalla un patrón de líneas amarillas y azules en el borde. Esta señal indica que su Spectrum se encuentra cargando el programa. Esta operación puede tomar varios minutos, según la extensión del programa.

**10** Las operaciones 7, 8 y 9 pueden ser repetidas varias veces si el programa está dividido en sectores.



**11** El programa puede comenzar a operar inmediatamente, una vez cargado. Recuerde parar el grabador.

**12** Cuando el programa no se inicia automáticamente, una vez cargado, la pantalla se pondrá en blanco y aparecerá el reporte `OK`, `0:1`. Pare el grabador.



**13** Oprima las teclas `R` (`RUN`) y `ENTER`. Ahora es posible ejecutar el programa.

### Consejos útiles sobre como cargar programas

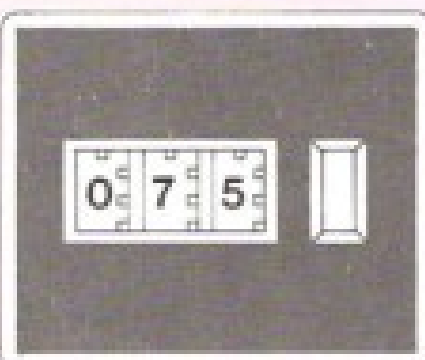
Los siguientes son algunos consejos útiles que le ayudarán a ahorrar tiempo al cargar un programa.

**1** Para ubicar los programas rápidamente identifique todas las cintas claramente con rótulos adecuados. Si el cassette contiene más de un programa, anote en el rótulo los nombres de los programas en el orden en que fueron grabados. Escriba el nombre del programa exactamente como lo usará la computadora.

1	2
1. Los programas	
2	
3. Los programas	
4. Los programas	
5. Los programas	
6. Los programas	
7. Los programas	
8. Los programas	
9. Los programas	
10. Los programas	

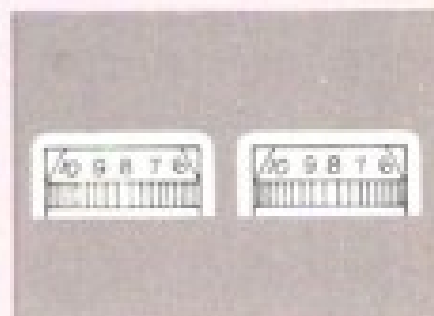
**2** Si su grabador de cassettes tiene un contador de vueltas, este se puede utilizar para ubicar la

posición del programa requerido en la cinta con varios programas grabados. Rebobine la cinta y ponga el contador en cero. A continuación ingrese la instrucción `LOAD` seguida del nombre de cualquier programa (entre comillas) que **NO** se encuentre grabado en esta cinta. Encienda el grabador y la Spectrum Plus irá nombrando cada programa grabado en la cinta, sin cargarlo. Anote el número de revoluciones frente al nombre del programa. Utilizando esta lista usted podrá ubicar fácilmente el programa requerido en el futuro.

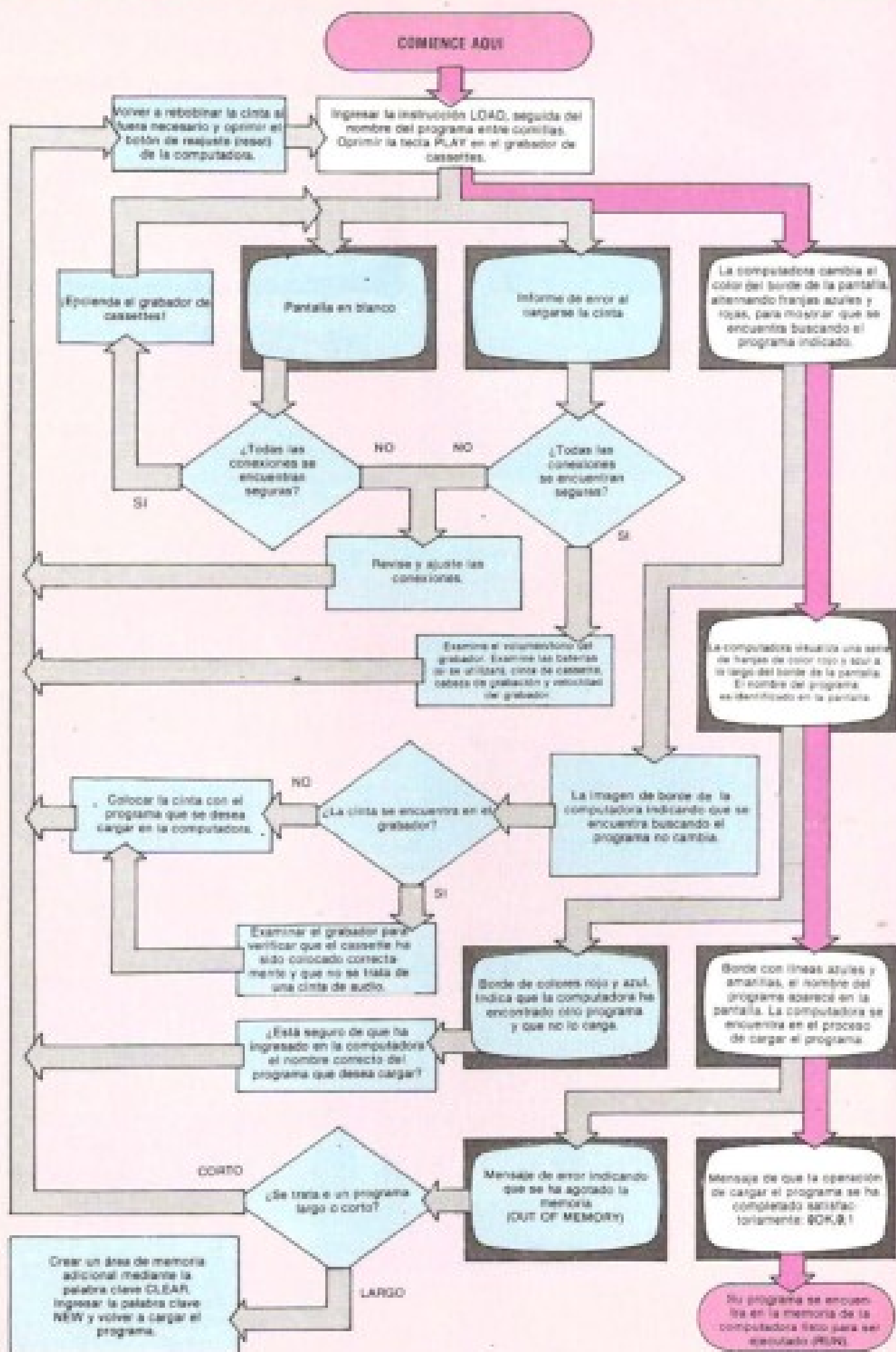


**3** Si la cinta se encuentra justo antes del principio del

programa requerido, o si no conoce el nombre del programa, entonces basta con ingresar la instrucción `LOAD` en lugar de `LOAD` seguido del nombre del programa entre comillas. Cuidé que no exista un espacio entre las dos comillas. Su Spectrum cargará el primer programa que encuentre en la cinta. Si el programa que aparece en la pantalla no es el que usted desea, bastará con oprimir la tecla `BREAK`, avanzar la cinta y volver a intentar.



**4** Pruebe diferentes niveles de sonido y tono para determinar cuáles son los que ayudan a su Spectrum a cargar los programas. Ajuste los controles del grabador a estos niveles antes de comenzar a cargar el programa.



## 2. PROGRAMANDO SU COMPUTADORA SPECTRUM

---

Este capítulo es una introducción a la programación para la computadora Spectrum. Su propósito es explicarle cómo dar instrucciones utilizando el teclado. Mediante estos controles usted podrá poner a trabajar la computadora. Se incluyen varios programas breves que le demostrarán las características especiales ofrecidas por la Spectrum. En el futuro podrá incorporar estas características en sus propios programas.

La computadora Spectrum tiene su propio lenguaje, el lenguaje de computadoras llamado BASIC. Para hacer que la computadora obedezca sus instrucciones, usted debe redactar un programa en BASIC utilizando con tal propósito las teclas en el teclado de su Spectrum. Además, el teclado le permite controlar la computadora mientras

La computadora Spectrum utiliza un dialecto o versión particular del idioma BASIC, que es sencillo pero muy poderoso. Además la Spectrum tiene una característica muy importante que facilita la tarea de redactar un programa: el sistema de introducción de palabras clave mediante una tecla única.

Las palabras clave son palabras con un significado especialmente definido en BASIC, cada una de ellas es una instrucción específica para la computadora, como, por ejemplo, PRINT o INPUT. En la mayoría de las computadoras es necesario ingresar las

palabras clave letra por letra, como si fuera una máquina de escribir, y es necesario verificar que no se cometan errores. Pero en la Spectrum basta con oprimir la tecla indicada para obtener una palabra clave completa.

El BASIC de la Spectrum tiene más de 86 palabras clave, las cuales son ingresadas mediante un total de 40 teclas (30 teclas-letra y 10 teclas-número). Muchas de las teclas tienen la capacidad de producir varias palabras clave que serán reconocidas por la computadora. La mayoría de las teclas le suministran una palabra clave determinada o una letra, número, signo o aún una forma (caracteres gráficos), todos los cuales pueden ser utilizados en sus programas.

### Seleccionando palabras clave y símbolos

En el teclado de la Spectrum se encuentran dos teclas que utilizará con mucha frecuencia, estas son: CAPS SHIFT y SYMBOL SHIFT. El propósito de estas teclas es ayudar a seleccionar las palabras clave y signos que usted desea hacer aparecer en la pantalla, operándolas en combinación con las demás teclas. En primer lugar, le sugerimos que estudie la distribución del teclado. Las siguientes dos páginas le demostrarán exactamente como seleccionar cualquiera de las instrucciones que aparecen indicadas en el teclado de la computadora. Después de haber aprendido todo esto, usted se encontrará preparado para escribir sus propios programas.

Generan número y también pueden introducir códigos de control para los colores en el programa. Ver la página 33.

Esta tecla produce letras mayúsculas. Para cambiar de mayúsculas a minúsculas bastará con volver a oprimir.

Estas teclas insertan  
códigos en las líneas del  
programa para generar  
colores normales o  
invertidos.

Son las teclas que controlan el movimiento del cursor, es lo que hace desplazarse en la dirección de la flecha. Estas teclas son utilizadas frecuentemente en los programas, para controlar el movimiento de figuras en la pantalla. También se las usa para editar programas.

Esta tecla es utilizada para seleccionar formas o caracteres gráficos en las teclas 1 a 8. Si usted oprime esta tecla y a continuación algunas de las teclas número (con o sin la tecla CAPS SHIFT) aparecerá un carácter gráfico en la pantalla. Si desea retornar la computadora a su modo de operación normal bastará con volver a oprimir la tecla GRAPHICS.

Estas 8 teclas ingresan palabras clave que controlan la visualización de los colores en la pantalla.

utilizada para  
cambiar el contenido de  
una línea en el programa  
sin que sea necesario  
volverlo a escribir  
Ver la página 21.

Esta tecla limpia el área de BASIC de la memoria, borrando cualquier programa que se encuentre almacenado en ella.

Al oprimir esta tecla conjuntamente con una tecla correspondiente a una letra se producirá una mayúscula. Si desea generar una serie de letras mayúsculas utilice CAPS LOCK.

Esta tecla produce la palabra clave que controla el sintetizador de sonido de la Spectrum.

Mantenga oprimida esta tecla y oprima una tecla letra o tecla-número para seleccionar la palabra clave o signo indicado en el tope de dicha tecla. (Ilustrada

con CAPS SHIFT  
seleccionará el símbolo  
o palabra clave indicada  
inmediatamente encima  
del tope de la tecla. Ver  
la páginas 20-21.

Tecla utilizada cuando se ha cometido un error, por ejemplo si se ha oprimido una tecla equivocada, para borrar una palabra clave, letra, número o signo —ver la página 10 por más detalles—.

Optima esta tecla para ingresar una linea de programa en la memoria de la computadora. Esta tecla se utiliza además para alimentar información a la computadora durante un programa.

- Esta tecla detiene el programa que está siendo ejecutado. No borrar el programa de la memoria de la computadora.

Produce un espacio  
como en una máquina  
de escribir.

## COMO OPERAR LAS TECLAS

Es posible obtener hasta seis palabras claves, letras, números o signos de la mayoría de las teclas de su Spectrum. Sin embargo, seleccionar un caracter o palabra clave en el teclado no es una operación difícil, una vez que usted se familiarice con las características especiales de su computadora. El resultado obtenido al oprimir una tecla dependerá del "modo" en el cual la computadora se encuentra en ese momento. Cada modo determina que la tecla produzca una información diferente como, por ejemplo, una letra, una palabra clave o un caracter gráfico. La ventaja de este sistema es que la Spectrum le ayudará a seleccionar el modo apropiado, para ingresar en el orden correcto, instrucciones e información en la computadora.

### Modo de palabra clave

Conecte y reajuste su Spectrum para que aparezca el mensaje inicial. Ahora oprima la tecla ENTER. La computadora hará aparecer una letra K parpadeante en el rincón de abajo a la izquierda. Este cuadro luminoso es el cursor. El cursor le señala el punto donde algo aparecerá en la pantalla, y la letra K le advierte que la computadora se encuentra en el modo de palabra clave. Oprima cualquier tecla-letra y la palabra clave situada en el tope de la sección elevada de la tecla aparecerá en la pantalla. Por ejemplo, oprima la letra Q, y la palabra clave PLOT aparece en la pantalla. Oprima las teclas CAPS SHIFT y 0 (DELETE) para borrarla e inserte con otras teclas.

Las teclas-número le darán números, pero tan pronto como usted oprima una tecla-letra, aparecerá la palabra clave superior en la sección elevada de la tecla.

Utilice DELETE otra vez para volver a hacer aparecer el cursor K. A continuación oprima cualquiera de las teclas SYMBOL SHIFT, manténgala deprimida y oprima cualquier otra tecla-letra. Ahora aparece la palabra clave o signo que está en rojo sobre la tecla en sí. O sea que el modo "palabra clave" está relacionado con el ingreso sobre la tecla misma.

### Como seleccionar una palabra clave, símbolo o caracter

Aquí veremos cómo seleccionar cualquier palabra clave, signo o caracter en una tecla numérica o de letra. Al seleccionar una función en una tecla, note donde está impreso sobre la tecla y de qué color, y usando los dos ejemplos

a continuación, decida qué otra tecla necesitará oprimir para cambiar al modo correcto. Siempre fíjese en el cursor sobre la pantalla primero, para saber en qué modo se encuentra la computadora.

	Modo "Palabra Clave" (K)		
	Tecla solo		BORDER
Tecla-letra	SYMBOL SHIFT	y tecla	*
	CAPS SHIFT	y	SYMBOL SHIFT
Tecla-letra	CAPS SHIFT	y	SYMBOL SHIFT
	SYMBOL SHIFT	y tecla	BRIGHT
Modo "Extensible" (E)			
Tecla-letra	CAPS SHIFT	y	SYMBOL SHIFT
	SYMBOL SHIFT	y tecla	BRIGHT
Modo "Letra" (L)			
Tecla solo	CAPS SHIFT	y tecla	B
	SYMBOL SHIFT	y tecla	*
Modo "Mayúsculas" (C)			
Tecla-letra	CAPS SHIFT	y	SYMBOL SHIFT
	SYMBOL SHIFT	y tecla	B
Modo "Gráfico" (G)			
Tecla-letra	CAPS SHIFT	y	SYMBOL SHIFT
	SYMBOL SHIFT	y tecla	*

\* Tecla solo de 1 a 12 con los gráficos definidos por el usuario.

	Modo "Palabra Clave" (K)		
	Tecla solo		3
Tecla-número	SYMBOL SHIFT	y tecla	0
	CAPS SHIFT	y tecla	TRUE VIDEO
Modo "Extensible" (E)			
Tecla-número	CAPS SHIFT	y	SYMBOL SHIFT
	SYMBOL SHIFT	y tecla	TRUE VIDEO
Modo "Letra" (L)			
Tecla solo	CAPS SHIFT	y tecla	3
	SYMBOL SHIFT	y tecla	0
Modo "Mayúsculas" (C)			
Tecla-número	CAPS SHIFT	y	SYMBOL SHIFT
	SYMBOL SHIFT	y tecla	0
Modo "Gráfico" (G)			
Tecla-número	CAPS SHIFT	y	SYMBOL SHIFT
	SYMBOL SHIFT	y tecla	0

## Modo de letras y mayúsculas

Después de haber producido una palabra clave o signo operando en el modo Palabra Clave, la computadora automáticamente cambia el modo del cursor al Modo L. El Modo L es también conocido como el Modo Letra (letter mode). Cuando en el Modo L, si usted oprime una tecla-letra, inmediatamente se visualiza dicha letra en minúscula. Igualmente, si oprime una tecla-número, aparecerá el número correspondiente. Si desea letra mayúscula deberá apretar simultáneamente la tecla CAPS SHIFT y la tecla de letra. Si quiere todas letras mayúsculas, oprima simultáneamente las teclas CAPS SHIFT y 2 (CAPS LOCK). El cursor cambia a C. Su Spectrum se encuentra en modo "mayúsculas", y Ud. obtendrá una letra mayúscula cada vez que oprima una tecla de letra.

Todavía puede obtener los números en modo mayúsculas. Para volver al modo "letra" oprima de nuevo CAPS SHIFT y 2 (CAPS LOCK).

## Modo extendido:

El siguiente modo se llama "extendido" y se obtiene oprimiendo simultáneamente las teclas CAPS SHIFT y SYMBOL SHIFT. El cursor cambia a una E. Oprima

cualquier tecla de letra, aparecerá la palabra clave en color verde encima de la tecla. Por ejemplo con la tecla B obtendrá la palabra clave BIN. Para obtener la palabra clave en rojo debajo de la tecla debe oprimir simultáneamente las teclas SYMBOL SHIFT y la deseada con la tecla B, por ejemplo, ahora obtendrá la palabra clave BRIGHT. De manera que, en el modo "extendido" puede obtener las palabras claves que hay arriba (en verde) y abajo (en rojo) de la tecla en cuestión. Luego de apretar una tecla en modo extendido (o de nuevo la combinación CAPS SHIFT-SYMBOL SHIFT) la computadora vuelve automáticamente al modo "letra" o mayúscula."

## Modo gráficos:

El quinto modo se llama modo "gráficos" y se obtiene presionando simultáneamente las teclas CAPS SHIFT y 9 (GRAPHICS). El cursor cambia a una G. Oprima las teclas de 1 a 8 y vea como aparecen los caracteres gráficos marcados sobre esas teclas. Ahora mantenga apretada la tecla CAPS SHIFT y oprima las teclas del 1 al 8. Los gráficos aparecen nuevamente, pero esta vez invertidos los colores blanco y negro. Para abandonar el modo "gráficos" debe volver a oprimir las teclas CAPS SHIFT y 9 (GRAPHICS) simultáneamente, ya que la computadora no lo hace en forma automática.

## Editando programas

Cuando usted escriba sus propios programas para su Spectrum deseará corregir los errores, ya sea en comandos o en líneas del programa, o efectuar alteraciones. Usted puede hacerlo fácilmente editando el programa.

### Como corregir un error

Cuando se ingresa en la computadora una línea en BASIC con error o un comando equivocado, la Spectrum mostrará en la pantalla un ? destellante delante del error. Para corregir el error, mantenga apretadas las teclas CAPS SHIFT y la tecla de control de cursor izquierdo o derecho (teclas 5 y 8), para mover el cursor a la derecha del error. Luego bórre el error usando DELETE, o agregue los elementos necesarios. Luego oprima ENTER.

Por ejemplo, suponga que desea que su computadora multiplique 7 por 8 pero se olvida de oprimir la tecla SYMBOL SHIFT para obtener el símbolo \*. Como consecuencia, usted habrá ingresado:

PRINT 7\*8

Su Spectrum no puede obedecer esta instrucción, por lo tanto cuando usted oprime la tecla ENTER, la computadora mostrará un signo de interrogación destellante delante de la b, el lugar donde se encuentra el error. Todo lo que

usted tiene que hacer es: mover el cursor justo a la derecha del error; oprimir la tecla DELETE para borrar la b; oprimir la tecla SYMBOL SHIFT y la tecla B para obtener \*; y oprimir la tecla ENTER para que la computadora obedezca el comando correcto. No es necesario mover el cursor hasta el final de la línea.



### Como editar la línea de programa

Al escribir un programa, usted construye una secuencia de líneas numeradas dando instrucciones, la cual se denomina listado. Si después de haber escrito su programa, efectúa el listado oprimiendo las teclas K (LIST) y ENTER, verá el signo > (próximo a una de las líneas del programa). Si no, use las teclas de control de cursor ↓ o ↑. Si ahora oprime las teclas CAPS SHIFT y 1 (EDIT), esta línea aparece copiada en la parte inferior de la pantalla y puede ser modificada como antes con las teclas del cursor y DELETE. Oprima ENTER para colocar la nueva línea dentro del programa. Si desea editar otra línea, mueva el signo > mediante las teclas de control de movimiento vertical del cursor, hasta la línea que desea modificar y oprima la tecla EDIT. Si esta operación toma demasiado tiempo, ingrese LIST seguido del número de línea y oprima EDIT. En cada caso, la línea resalta, aparecerá al fondo de la pantalla y se la puede cambiar.

Para borrar una línea completa del programa, ingrese el número de la línea y oprima la tecla ENTER. Si ingresa un programa que contiene un error, un informe de error aparecerá en la pantalla. Ver la página 74 por más detalles sobre estos informes.

# UNA CALCULADORA EN SU TELEVISOR

Su Spectrum puede hacer cálculos muy rápidamente y con gran precisión. Todo lo que necesita son algunos números y signos, tales como + y -, que le indiquen cuáles son las operaciones que debe efectuar con los números.

Primero ingrese la siguiente instrucción (el signo de + se encuentra en la tecla-letra K):

```
PRINT 6 + 2
```

Esta instrucción es un "comando". Cuando usted oprime la tecla ENTER, el comando desaparece de la pantalla y es sustituido por el resultado. En este ejemplo el resultado es el número 8.

Su Spectrum utiliza cinco signos conocidos como operadores aritméticos para los cálculos. El panel en la parte inferior de esta página explica la función de cada uno de ellos. El procedimiento para todos ellos es similar, siempre mediante la instrucción PRINT.

La Spectrum también puede visualizar en la pantalla los cálculos y su resultado. Ingrese el siguiente comando en su calculadora:

```
PRINT "6 + 2 = "; 6 + 2
```

La computadora responderá, visualizando:

6 + 2 = 8

Ha sucedido lo siguiente: el comando PRINT determina que todo lo que se encuentra entre ambas comillas (") se visualice en la pantalla, en nuestro ejemplo: 6 + 2 = .

El punto y coma le indica a la computadora que el resultado debe ser ubicado inmediatamente después del signo de igual (=).

## Los signos utilizados por la Spectrum para sus cálculos

La computadora emplea los siguientes operadores aritméticos para efectuar operaciones matemáticas. Observe que la computadora no utiliza los símbolos x o +.

Símbolo	Tecla	Función	Ejemplo
+	K	Suma dos números.	8 + 2 = 10
-	J	Resta dos números.	8 - 2 = 6
*	B	Multiplica dos números.	8 * 2 = 16
/	V	Divide dos números.	8 / 2 = 4
^	H	Eleva un número al cuadrado.	8 ^ 2 = 64

## Su primer programa

Después de haber ejecutado un comando, su Spectrum lo olvida. Si usted desea que su computadora repita los cálculos, entonces le sugerimos escribir un programa. Ingrese las instrucciones en la computadora y luego oprima la tecla ENTER.

```
10 PRINT 6 + 2
```

Esta vez sus intrucciones no son obedecidas inmediatamente. En cambio, la computadora visualiza la instrucción en la pantalla. A continuación oprima R (RUN) y la tecla ENTER. Inmediatamente aparece el resultado de la operación: 8.

La totalidad de la instrucción es ahora un programa de computadora. Al poner un número al principio, su Spectrum almacena la instrucción en su memoria pero no la ejecuta hasta que se indique. Cuando ingresa un programa oprimiendo R (RUN) seguido de ENTER, su Spectrum ejecutará la instrucción. La instrucción se transforma en una sentencia en lugar de un comando y forma una línea numerada en un programa. Las sentencias en un programa son siempre ejecutadas de acuerdo al número de línea. Estos normalmente van de diez en diez.

Ahora podemos poner a trabajar la Spectrum. Ingrese el programa a continuación. Recuerde oprimir la tecla ENTER después de haber ingresado cada línea en la computadora y al terminar oprima la tecla R (RUN) y ENTER. Una vez ejecutado el programa, el resultado debería ser el siguiente:

## TABLA DE NUMEROS



La tabla visualiza todos los números entre 1 y 255. Ahora oprima cualquier tecla excepto N, la barra espaciadora, STOP o BREAK. A continuación aparece un juego completo de otros números.

Este programa utiliza una "variable". En este ejemplo la variable se denomina n. Es posible utilizar cualquier letra o palabra; en nuestro ejemplo n es la abreviación por el término número. Se ha

adjudicado un valor a esta variable que cambia a medida que se ejecuta el programa. En la línea de programa 10, se ha incluido la palabra clave LET para adjudicar el valor 1 a la variable n. La línea de programa 20 visualiza el valor seguido de un espacio. Seguidamente, en la línea 30 volvemos a utilizar la palabra clave LET, esta vez para incrementar el valor de la variable en 1. Como resultado de esta instrucción, el valor de la variable ahora es 2 (1 + 1). La línea 40 contiene la palabra clave GOTO (en inglés: vé a) para dirigir la computadora nuevamente a la línea 20 que ahora visualiza el número 2.

Este ciclo se repite una y otra vez hasta que la pantalla de su televisor está llena de números.

### Como enseñarle a su computadora a pedir números

Interrumpa el programa oprimiendo la tecla BREAK. Ahora ingrese una nueva línea.

10 INPUT n

Esta nueva línea de programa sustituye la línea 10 anterior. Cuando se ejecuta el programa, la computadora esperará a que usted ingrese un número (INPUT en inglés significa ingresar). Ingrese cualquier número desde el teclado de su computadora y oprima la tecla ENTER. Ahora, los números de la tabla visualizada en la pantalla del televisor, comienzan a partir del número que usted ingresara en la computadora. Esto se debe a que INPUT n determina que el valor de la variable n sea idéntico al número que usted ingresara.

### Programa para una tabla de multiplicar

Oprima el botón de resta para borrar de la memoria de la computadora el programa anterior. Ingrese en la computadora el siguiente programa que hace que su Spectrum multiplique. Ingrese cualquier número mediante el teclado en la computadora y este producirá la tabla de multiplicar para ese número en la pantalla. Si desea que la tabla de multiplicar continúe oprima cualquier tecla, excepto N, BREAK, o la barra espaciadora. Oprima BREAK y vuelva a ejecutar el programa para crear una nueva tabla. Ahora le mostramos el programa para la tabla y los

### TABLA DE MULTIPLICAR

```
10 LET x=1
20 INPUT n
30 PRINT "1"-"+"; x; "="; n * x
40 LET x=x+1
50 GO TO 30
```



resultados que se obtienen al ingresar 3 ya continuación 145.



### ¿Por qué es necesario utilizar paréntesis?

En ciertos casos será necesario emplear paréntesis al realizar cálculos. Por ejemplo, ingrese los siguientes dos comandos y compare los resultados:

```
PRINT 6 + 2/4
PRINT (6 + 2)/4
```

En el primer caso el resultado obtenido será 6.5 en el segundo el resultado será 2. La causa de estos resultados diferentes es que la computadora tiene incorporado un sistema de prioridades que utiliza durante sus cálculos. El orden de prioridades es el siguiente: primero ejecuta el signo  $\frac{1}{2}$ ; en segundo lugar  $*$  o  $/$ ; y finalmente  $+$  o  $-$ . Pero en todos los casos siempre ejecutará primero las operaciones contenidas entre paréntesis. Por lo tanto, en el caso del primer ejemplo la computadora primero dividió 4 por 2, y luego le sumó el resultado de la división al número 6. En el segundo ejemplo, la computadora efectuó en primer lugar la operación entre paréntesis (sumó 6 más 2) y luego dividió el resultado por 4.

### La puntuación en su Spectrum

La Spectrum hace uso de una variedad de signos de puntuación. Todos ellos cumplen papeles muy importantes debido a que eran como instrucciones para la computadora, afectando la forma en que ésta interpreta las líneas de programa o visualiza información en la pantalla de la televisión.

- **Punto y coma** Cuando utilizada conjuntamente con la palabra clave PRINT, el punto y coma le indica a la computadora que debe visualizar en la pantalla dos elementos uno al lado del otro.
- **Dos puntos** En una línea de programa señala el final de una sentencia y el comienzo de la próxima.
- **Comillas** Cualquier carácter o serie de caracteres contenidos entre comillas son tratados como texto y visualizados en la pantalla, no son considerados como números o variables. Las comillas marcan el principio y el final de una "cadena" (string).
- **Coma** Cuando utilizada con la palabra clave PRINT, la coma le indica a la computadora que visualice el elemento siguiente, ya sea en el centro de la línea o al principio de la línea siguiente.
- **Punto** Punto decimal o punto.

# COLORES, COMO EMPLEARLOS

Su Spectrum puede producir ocho colores diferentes. Cada color tiene su propio número código. Es posible utilizar cada color en tres formas diferentes: como un color de borde, como un color de tinta, y finalmente como un color de papel.

## Spectrum . Códigos de los colores

La tabla enumera los colores utilizados por la Spectrum y sus números código respectivos. No es necesario memorizar los números porque las teclas-número que los producen indican cual es el color correspondiente a cada una de ellas.

Número	Color
0	Negro
1	Azul
2	Rojo
3	Magenta (morado)
4	Verde
5	Celeste
6	Amarillo
7	Blanco

El tono de los colores en la pantalla de su televisor dependerá del equipo y el ajuste de sus controles de color, contraste y brillo. Recuerde que se requiere un televisor de colores para este propósito.

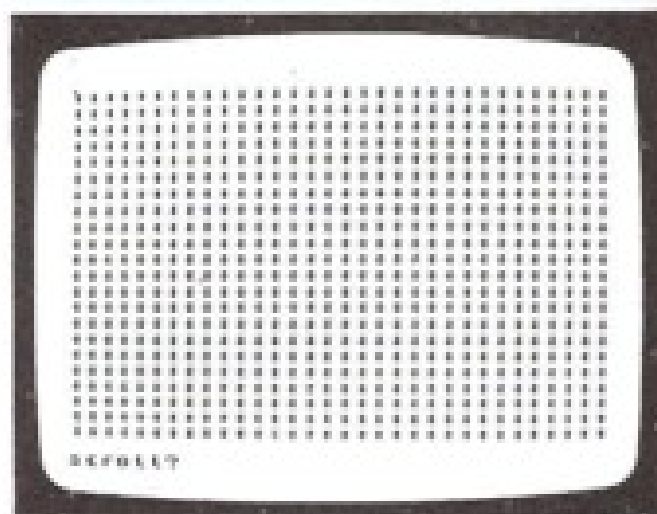
## Los tres modos de empleo de color en su Spectrum

Los colores de su computadora pueden ser controlados de tres maneras diferentes. El color de borde (border color) es el color del área en torno del sector central de la pantalla. Los colores de tinta (ink colours) son los colores de los caracteres en la pantalla (letras, números, signos, y formas gráficas), puntos o líneas. El color de papel (paper color) es el color del fondo, ya sea en la totalidad del área de visualización o encuadrado en torno a cada uno de los caracteres.

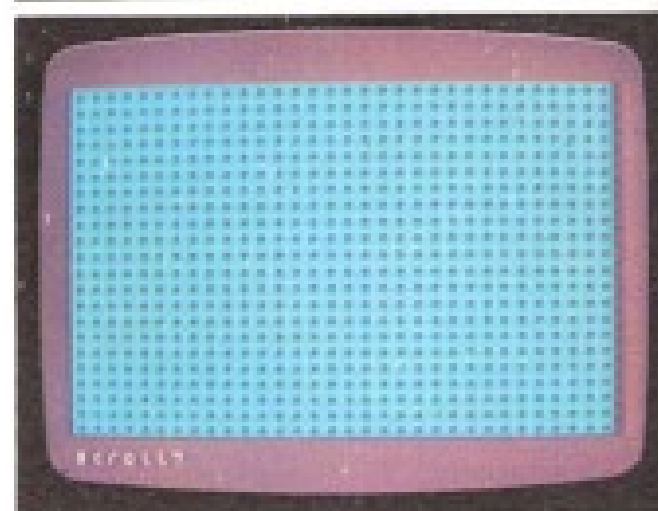
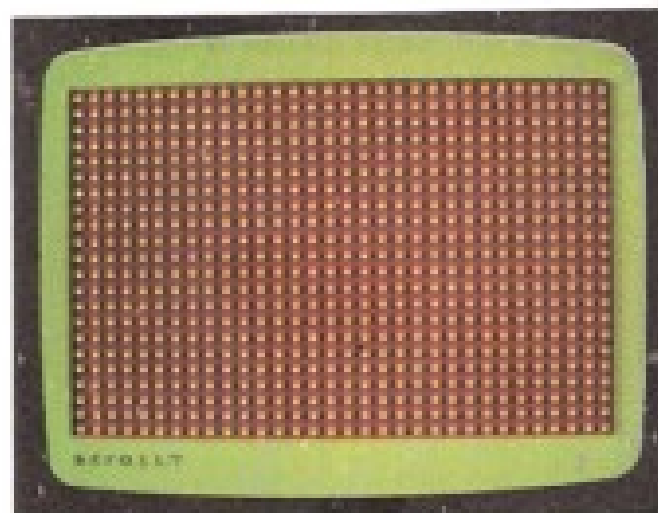
Cuando usted enciende su Spectrum la computadora emplea una serie de colores preestablecidos. La tinta es de color negro, mientras que borde y papel son de color blanco. Es posible cambiar estos colores simplemente, mediante comandos directamente desde el teclado. Ya hemos visto antes como cambian los colores en el televisor, en las páginas 6 y 7. Ingrese el siguiente sencillo programa en la computadora.

## PRUEBA DE COLORES

```
10 PRINT "4"
20 GO TO 10
```



Al ejecutar este programa la computadora produce un patrón de estrellas en blanco y negro. Ahora oprima la tecla de BREAK e ingrese algunos nuevos comandos de control para los colores. Ingrese las palabras clave BORDER, INK, y PAPER, cada una de ellas seguida de un número de 0 a 7. después de cada instrucción oprima la tecla ENTER y vuelva a ejecutar el programa. Los siguientes son dos ejemplos, el primero con la instrucción BORDER 4, PAPER 2, e INK 7. El segundo ejemplo contiene las instrucciones BORDER 3, PAPER 5, e INK 1.



### Como utilizar colores en sus programas

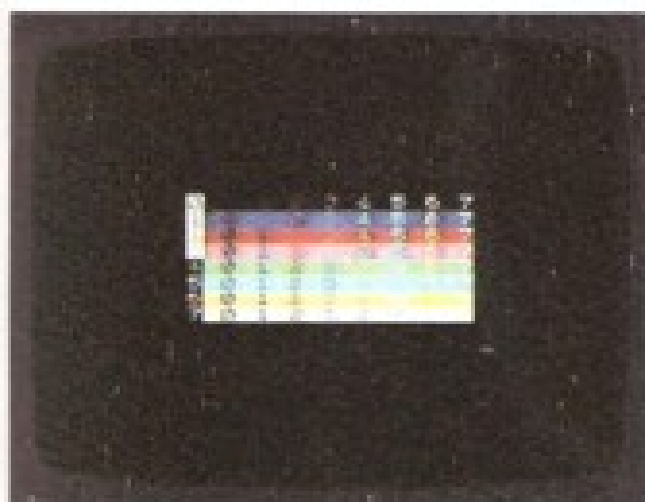
Es posible utilizar las palabras clave BORDER, PAPER e INK en su programa, para crear textos, tablas, patrones, y dibujos de una variedad de colores.

Cuando se incluye la palabra clave BORDER en una línea de programa, el color del borde cambia cuando la computadora llega a ese punto. La palabra clave INK independientemente cambia el color de los caracteres o líneas la próxima vez que éstas aparezcan en la pantalla. Finalmente, la palabra clave PAPER incluida independientemente en una línea de programa cambia el color del papel, pero solamente en torno de los caracteres (incluyendo puntos y líneas) en la pantalla. Si desea que la totalidad del fondo del área de visualización tenga un color determinado, lo que debe hacer es ingresar las palabras clave PAPER y CLS.

También es posible emplear INK y PAPER después de PRINT. En este caso, dichas instrucciones solamente afectarán los caracteres específicos controlados por PRINT. El programa siguiente demuestra todos los colores de borde (border), ink (tinta) y de papel (paper). El programa también le demuestra cómo utilizar las palabras clave INK y PAPER después de PRINT.

### COMBINACIONES DE COLORES

```
10 FOR b=0 TO 7
20 BORDER b: PAPER b: CLS
30 PRINT AT 5,10: INK 9:b
40 FOR p=0 TO 7
50 PRINT AT p+5,5: INK p: PAPER p
60 BEEP 0.5:b+2-20+p
70 FOR i=0 TO 7
80 PRINT INK i: PAPER p: " ";i
90 BEEP 0.01:++
100 NEXT i
110 NEXT p
120 NEXT b
```



Al ejecutar este programa, usted verá todas las combinaciones de borde, papel y tinta (o en inglés: border, paper e ink). El programa incluye tres variables: b para el número del color correspondiente al borde; i controlando el número correspondiente a la tinta (ink); y p, que es el número del papel (paper). La palabra clave BEEP produce el sonido, y las líneas del programa comenzando con FOR y NEXT marcan el principio y el final de un ciclo del programa. El propósito de este ciclo es cambiar todos los números de los colores desde 0 hasta 7 consecutivamente. Observe que INK y PAPER pueden tener un valor de 9, este

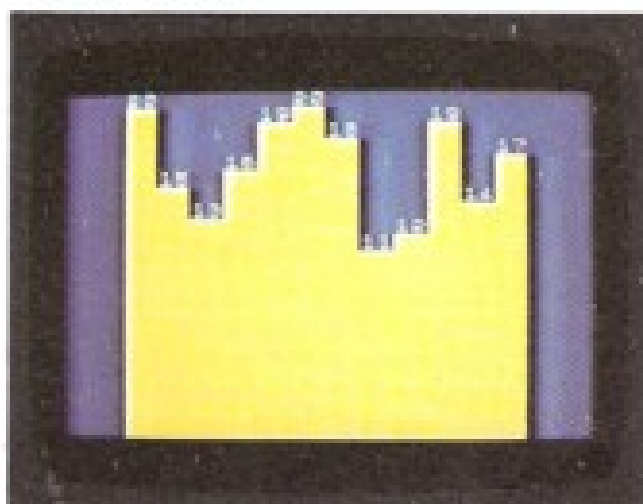
número código causa que la tinta o el papel se tornen de color blanco o negro, para que resalten contra el fondo o un carácter.

### Programa para crear un gráfico de colores

El programa siguiente utiliza los colores de su Spectrum para dibujar un gráfico de barras. El propósito del gráfico es mostrar las temperaturas diurnas producidas durante un período determinado. Las temperaturas son mostradas gráficamente como barras amarillas y con números. En la línea del programa 50 tenga cuidado de dejar dos espacios en blanco entre las comillas (" ").

### GRAFICOS DE BARRAS

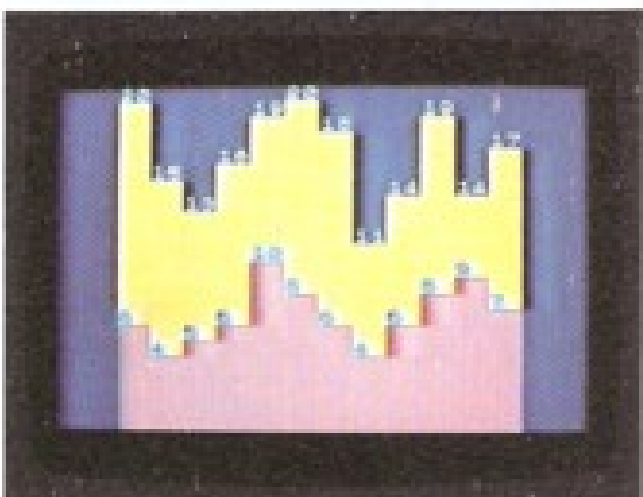
```
10 BORDER 0: PAPER 1: CLS
20 LET i=4
30 FOR x=1 TO 12
40 READ i
50 FOR l=21 TO 21-i STEP -1
60 PRINT PAPER 0:AT l,c: " "
70 NEXT l
80 PRINT INK 2:AT 20-i,c:i
90 LET c=c+2
100 NEXT x
110 DATA 20,15,13,16,19,20,18,1
1,12,19,14,17
```



Ahora le sugerimos agregar las siguientes líneas de programa, incluyendo la nueva línea 110, tal como se muestra. Como resultado de estas nuevas instrucciones, el gráfico será de dos colores.

### GRAFICO DE BARRAS DOBLE

```
55 READ i
56 FOR l=21 TO 21-i STEP -1
57 PRINT PAPER 3:AT l,c: " "
58 NEXT l
59 PRINT INK 1: PAPER 5:AT 20-i,c:i
110 DATA 20,6,15,4,13,5,16,6,19,
10,20,6,16,6,11,4,14,6,19,6,14,
6,17,7
```



## GRAFICAS SENCILLAS

Su computadora Spectrum le ofrece facilidades para gráficas de alta resolución y de baja resolución. Ambos tipos de gráficas pueden aparecer en la pantalla al mismo tiempo. Las imágenes gráficas en el modo de baja resolución consisten en bloques de color. El propósito de estas dos gráficas es explicar como producir estos bloques de color y como ubicarlos en la pantalla de su computadora.

### La pantalla de baja resolución

En el modo de baja resolución la pantalla del televisor se divide en una cuadrícula, cuyas dimensiones son 32 líneas verticales o columnas y 22 líneas horizontales. Cada posición en la pantalla es identificada por dos números, que son sus coordenadas. En primer lugar se indica el número de la línea en la cual se encuentra el punto. Las líneas se cuentan a partir de la parte superior de la pantalla, y el valor de la línea en el tope es 0, el valor de la línea en la parte inferior de la pantalla es 21.

A continuación se indica el número de la columna, éstas se cuentan de izquierda a derecha. La primera columna a la izquierda tiene el número 0 y la última a la derecha tiene el número 31. (En la página 8 encontrará un diagrama de la cuadrícula de baja resolución).

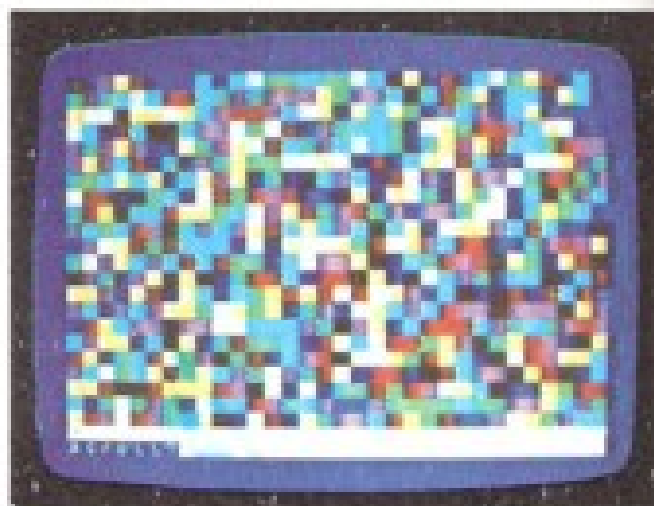
El siguiente programa llenará las posiciones de los caracteres con colores. La palabra clave RND (la encontrará en la tecla R) seleccionará al azar un color de tinta.

### Cómo seleccionar caracteres gráficos

Su Spectrum trae un conjunto de caracteres gráficos en el teclado, que hace sencilla la programación de gráficos de baja resolución. Están sobre las teclas 1 a 8. Para producir estos caracteres gráficos en la pantalla mantenga apretada la tecla CAPS SHIFT y oprima la tecla 9 (GRAPHICS) para seleccionar el modo "gráfico". Luego oprima las teclas del 1 al 8, usando la tecla SPACE entre una y otra. Los gráficos

### CUADRADOS

```
10 BORDER 1 INK END:7
20 PRINT
30 GO TO 10
```



El programa genera cuadrados en toda la pantalla de el televisor. Para indicarle a la computadora que visualice un carácter en una posición determinada de la pantalla, deberá insertar las palabras clave PRINT y AT. La palabra clave AT es insertada después de PRINT y es seguida por el número de líneas, una coma, el número de la columna, y un punto y coma.

Por ejemplo, el comando

```
PRINT AT 11,15,"*"
```

visualizará un asterisco en el punto situado en la intersección de las líneas 11 con la columna 15, esto es, el centro de la pantalla.

### Cómo dibujar un arco iris

Un buen procedimiento para producir imágenes de colores es incluir ciclos FOR NEXT en los programas de gráficas. Los ciclos FOR NEXT son grupos de instrucciones contenidas en un programa, que la computadora repite una cantidad determinada de veces.



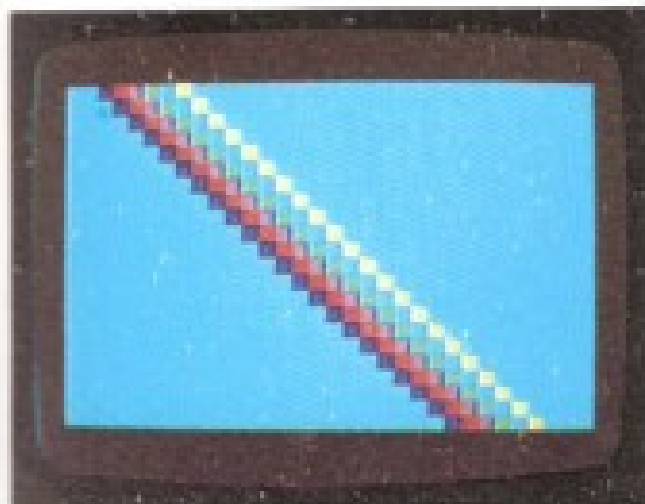
aparecen en la parte inferior de la pantalla. La parte negra de cada carácter en la tecla es del color de la tinta, y la parte blanca es del color del papel. Oprima las mismas teclas de nuevo, manteniendo apretada la tecla CAPS SHIFT. Esta vez aparecerán con los colores de tinta y papel invertidos. De esta manera se ponen los caracteres gráficos en líneas de programa. Para obtener el modo "gráfico" vuelva a apretar CAPS SHIFT y 9 simultáneamente.

El programador le indicará a la computadora la cantidad de veces que ésta debe ejecutar el ciclo mediante una instrucción contenida en la primera línea del ciclo. Este método puede ser utilizado, por ejemplo, para visualizar caracteres en la pantalla del televisor.

Es posible programar en la pantalla más de un ciclo al mismo tiempo. También es posible ubicar un ciclo dentro de otro, con resultados muy útiles. El siguiente programa le enseñará cómo se utilizan dos ciclos (uno "anidado" dentro del otro) para cambiar los colores y las posiciones producidas en la pantalla por las palabras clave INK y AT. Las explicaciones al final de la página le indicarán cómo programar estos ciclos.

### ARCO IRIS

```
5 BORDER 0: PAPER 5: CLS
10 LET X=1
20 FOR I=0 TO 31
30 FOR J=1 TO 6
40 PRINT INK (J) AT I,X+J: "■"
50 NEXT J
60 LET X=X+1
70 NEXT I
```



### Programando imágenes

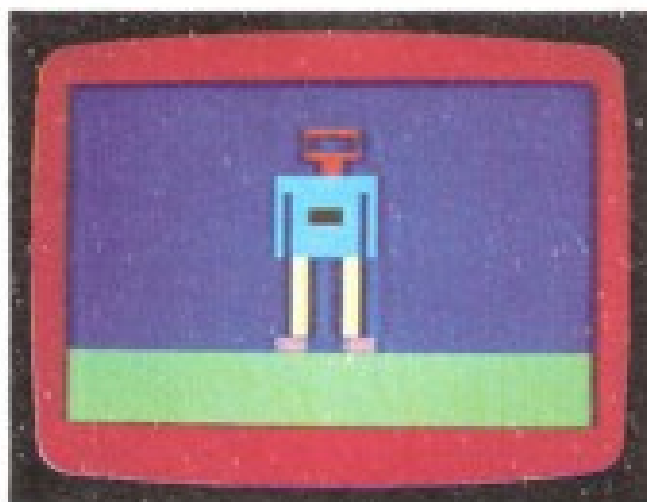
Cuando en el modo de gráficas de baja resolución, es posible "pintar" imágenes mediante el procedimiento de determinar las coordenadas de las posiciones y los colores de los caracteres gráficos. Para planear una figura le sugerimos que utilice una cuadrícula para imágenes de baja resolución como la incluida en la página 80.

A continuación, seleccione los caracteres gráficos, tal como se explica en la página opuesta las líneas del programa, una a una, para construir la imagen.

El programa siguiente le demuestra el tipo de resultados que puede producir con sus programas. Todas las formas incluidas en la imagen pueden encontrarse en las teclas-número del teclado. Es posible ingresar todas las líneas del programa antes de ejecutarlo. También puede observar cómo el robot es constituido elemento tras elemento, al ingresarse cada nueva línea de programa, si le indica a la computadora que ejecute las líneas del programa en su memoria, después de ingresar una nueva línea.

### EL ROBOT

```
5 BORDER 2: PAPER 1: CLS
10 PRINT INK 5: AT 3,15: "■"
20 PRINT INK 5: AT 4,15: "■"
30 PRINT INK 5: AT 5,15: "■"
40 FOR I=7 TO 10: PRINT INK 5: AT I,15: "■": NEXT I
50 PRINT INK 6: PAPER 0: AT 6,15: "■"
60 PRINT INK 2: AT 11,13: "■"
70 FOR I=11 TO 15: PRINT INK 6: AT I,14: "■": NEXT I
80 PRINT INK 3: AT 16,13: "■": AT 17,13: "■"
90 FOR I=17 TO 21: FOR J=0 TO 31: PRINT INK 4: AT I,J: "■": NEXT J
100 NEXT I: NEXT I
```



En la línea 70 del programa, la palabra clave TAB, después de PRINT, ubica un carácter a lo largo de la línea donde la computadora se encuentra trabajando en ese momento. TAB es seguida por número entre 0 y 31, que especifica la posición de una columna.

### Cómo emplear los ciclos FOR NEXT

Un ciclo FOR NEXT siempre se inicia con una línea de programa conteniendo las palabras clave FOR y TO, junto con la variable y su valor inicial y final. Por ejemplo:

```
30 FOR C = 1 TO 6
```

En el ejemplo, la variable es C. El ciclo iniciado de esta forma contendrá una línea (o líneas) que le indican a la computadora que repita una operación. Las líneas también pueden incluir la variable C. Los ciclos FOR NEXT siempre terminan con la palabra clave NEXT seguida por la variable. Por ejemplo:

```
50 NEXT C
```

La computadora ejecuta el programa repitiendo el ciclo FOR NEXT completo una cantidad indicada de veces. El valor inicial de la variable, indicado antes de TO, se incrementa en pasos de 1, hasta que alcanza el límite establecido después de TO. En el ejemplo, el ciclo se repite seis veces, el valor inicial de C es 1 y cambia sucesivamente a 2, 3, 4, 5 y finalmente a 6.

En el primer programa de página 25, se utilizan tres ciclos en un "nido". Esto significa que cada valor del ciclo 'exterior', el ciclo 'intermedio' completa su proceso, y el ciclo 'interior' completa sus procesos con más frecuencia.

## DIBUJANDO CON SU COMPUTADORA

Las posibilidades de su Spectrum son enormes. Gracias a su capacidad de generar figuras sumamente nítidas, de gran resolución (high-resolution), es posible utilizar la Spectrum para crear imágenes detalladas con líneas claras, y empleando líneas rectas o curvas.

Los dibujos y demás gráficas de gran nitidez se componen de una cantidad de pequeños puntos situados uno junto al otro, formando líneas, o relleno de espacios (por ejemplo un rectángulo). El tamaño de cada uno de estos puntos es 1/64 parte del tamaño de los cuadrados empleados en gráficas de baja nitidez o resolución (low-resolution). Cuando se ingrese el siguiente comando en su computadora.

**PLOT 128,87**

usted podrá apreciar uno de esos puntos en el centro de la pantalla.

Los puntos empleados para generar figuras gráficas de alta nitidez o resolución se denominan "pixels", una abreviación de las palabras inglesas "picture cell" o celdas de colores. Como en el caso de los caracteres de baja resolución, la posición de cada pixel es especificada mediante dos números.

### La red de resolución o nitidez

La red de resolución consiste en 256 pixels distribuidos en hileras horizontales a lo largo de la pantalla y 176 distribuidos en columnas verticales. Sin embargo, y a diferencia de lo que sucede en el caso de las imágenes de baja resolución, el primer número o coordenada es utilizado para indicar la posición sobre una hilera horizontal en la pantalla. Estas posiciones varían entre 0 sobre el borde de la izquierda de la pantalla, hasta 255 sobre el borde de la derecha. El segundo número es la posición sobre la columna vertical. En este último caso los números varían entre 0 (parte inferior de la pantalla) y 175 (parte superior de la pantalla). Por ejemplo, las coordenadas 0,0 corresponden al rincón inferior de la izquierda de la pantalla de su televisor (en el caso de imágenes de baja resolución esas mismas coordenadas corresponderían a la esquina superior de la izquierda). La página 88 contiene un plano de la red de alta resolución.

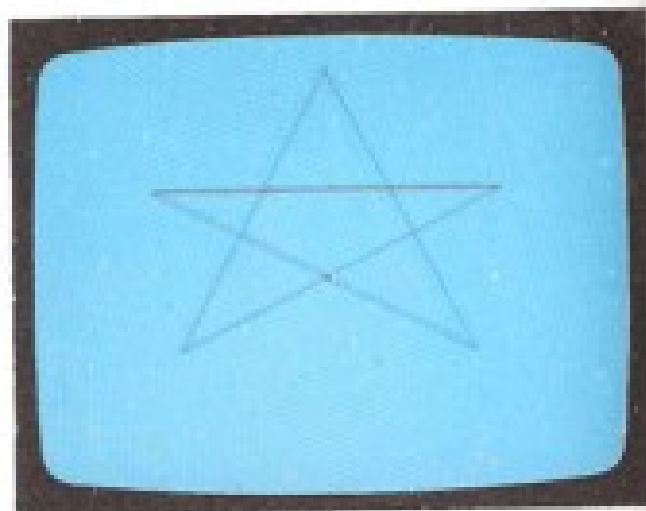
### Ubicando puntos y dibujando sobre la pantalla

Solamente se requieren tres palabras claves para producir gráficas de alta resolución: PLOT, DRAW y CIRCLE. La palabra clave PLOT es seguida por las coordenadas horizontal y vertical del punto requerido separadas por una coma. Esta instrucción ubica un pixel precisamente en la posición definida por las coordenadas. La palabra clave DRAW también es seguida por dos números, separados por una coma, pero estos números no son coordenadas de una posición. Estos números son distancias medidas en pixels entre dos posiciones, ya sea en dirección vertical u horizontal. El propósito de la instrucción DRAW es trazar una línea entre esas dos posiciones.

Si esta es la primera vez que se utilizan las palabras clave PLOT o DRAW en el programa la primera posición es 0,0. Si estas palabras clave han sido utilizadas anteriormente en el mismo programa, entonces la primera posición (el punto de partida para la nueva línea), es la última posición PLOT o DRAW anterior, cualquiera sea la más reciente. En este caso, la computadora ejecutará la instrucción DRAW, dibujando una línea desde aquel punto hasta la nueva posición. Veamos un ejemplo práctico: La palabra clave PLOT le indica a la computadora que debe mover la posición del punto donde comienza la línea al tope de la pantalla. Luego las cinco sentencias DRAW dibujan las cinco líneas rojas.

### ESTRELLA

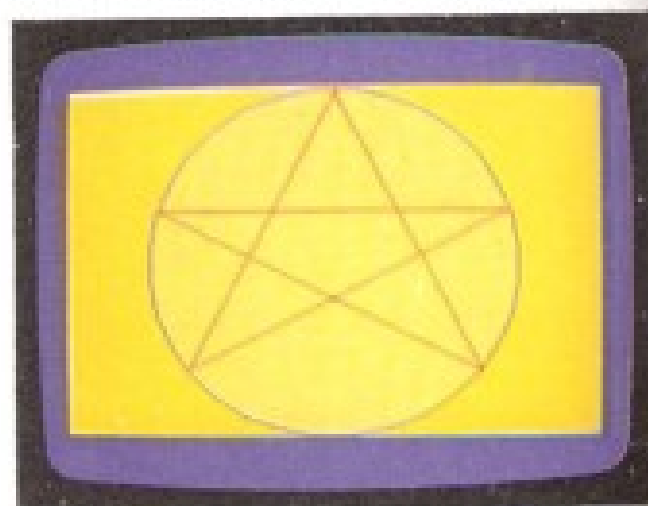
```
10 INK 2
20 PLOT 175,0
30 DRAW 70,-175
40 DRAW -175,0
50 DRAW 0,-175
60 DRAW 175,0
70 DRAW 70,-175
```



A continuación agregue las siguientes líneas de programa a su programa "Estrella"

```
4 BORDER 1: PAPER 8: INK 1: CLS
5 CIRCLE 128, 87, 87
```

Ejecute este programa, y verá como la computadora dibuja una estrella roja en un círculo azul.



La instrucción CIRCLE requiere tres números. Los primeros dos números le indican a la computadora cual es la posición del centro del círculo, y el tercer número es el radio de éste. También es posible

agregar un tercer número a las sentencias incluyendo la palabra clave DRAW.

Pruebe en su programa valores entre 2 y -2, para ver qué sucede.

### Como rellenar figuras

Crear figuras sólidas en alta resolución es una operación sencilla, el procedimiento consiste en dibujar una cantidad de líneas paralelas una junta a la otra. Esto puede hacerse mediante un ciclo (loop) conteniendo las palabras clave FOR NEXT, el cual cambiará la posición de DRAW, incrementándola en pasos de una unidad.

### TRIANGULO SOLIDO

```
10 BORDER 1: PAPER 5: INK 2: C
L5
20 FOR x=-100 TO 100
30 PLOT 120,150
40 DRAW x,-120
50 NEXT x
```



Es posible obtener un resultado interesante dibujando líneas con una pequeña separación. Es posible hacerlo agregando la palabra clave STEP y un número, a la sentencia FOR. Ingrese en la computadora la nueva línea de programa 20 y ejecute nuevamente el programa:

```
20 FOR x = -100 TO 100 STEP 4
```

Ahora aparece una imagen en forma de abanico. Esto se debe a que cuando la computadora dibuja las líneas, la palabra clave STEP incrementa en pasos de 4 unidades, en lugar de pasos de 1 unidad como anteriormente.

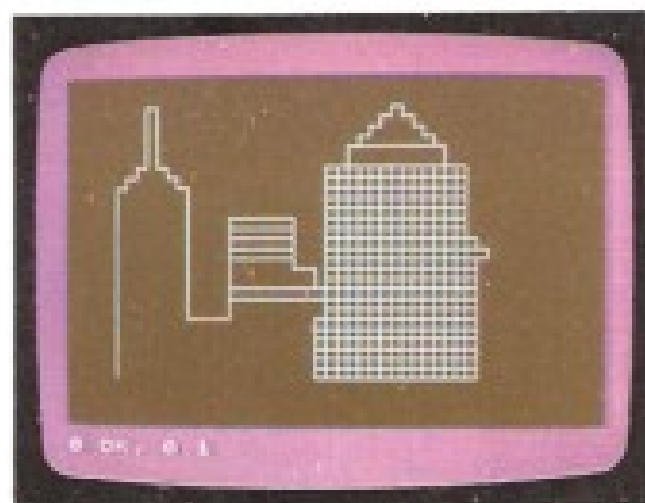


Utilizando la pantalla de su televisor como una pizarra. No es necesario escribir un programa cada vez que desee producir un dibujo o un patrón. También es posible utilizar un programa que le permitirá dibujar directamente en la pantalla.

Este programa comienza utilizando la palabra clave INPUT, cuyo propósito es preguntarle cual es el número control para la tinta (ink number). A continuación, y utilizando nuevamente la palabra clave INPUT (esta vez con un signo \$, para nombrar o "bautizar" la cadena de comandos), le indica a la computadora que dibuje líneas cortas cada vez que usted oprima una de cada cuatro teclas específicas. Estas son U,D,L, o R, seguidas en cada caso por la palabra clave ENTER. El programa utiliza las palabras claves IF y THEN para tomar decisiones.

### PIZARRA

```
10 INPUT "INK " I
20 BORDER 3: PAPER 0: INK I: C
L5
30 PLOT 25,25
40 LET X=5
50 INPUT $B
60 IF $B="U" THEN DRAW 0,X
70 IF $B="D" THEN DRAW 0,-X
80 IF $B="L" THEN DRAW X,0
90 IF $B="R" THEN DRAW -X,0
100 GO TO 50
```



### Tomando decisiones con IF y THEN

En este programa "Pizarra", las líneas 60 a 90 contienen sentencias con las claves IF y THEN. Estas permiten a su Spectrum tomar una decisión. En este caso la computadora examinará si la tecla oprimida es alguna de las indicadas en el programa (u,d,l,r). Si cualquiera de estas teclas es oprimida, entonces la computadora dibujará una línea. No dibujará la línea respectiva si la letra mayúscula es oprimida.

La clave IF es siempre seguida de una condición que la Spectrum comprueba si es verdadera o si se está cumpliendo. Si es verdadera ejecutará la acción indicada después de la tecla clave THEN. Si no es verdadera el programa pasa a la línea siguiente.

Todas las instrucciones en la línea después de THEN dependen de la decisión de la computadora. Por ejemplo en la línea siguiente:

```
110 IF b = 5 THEN PRINT "": GOTO 200
```

la computadora solo irá a la línea 200 si b = 5.

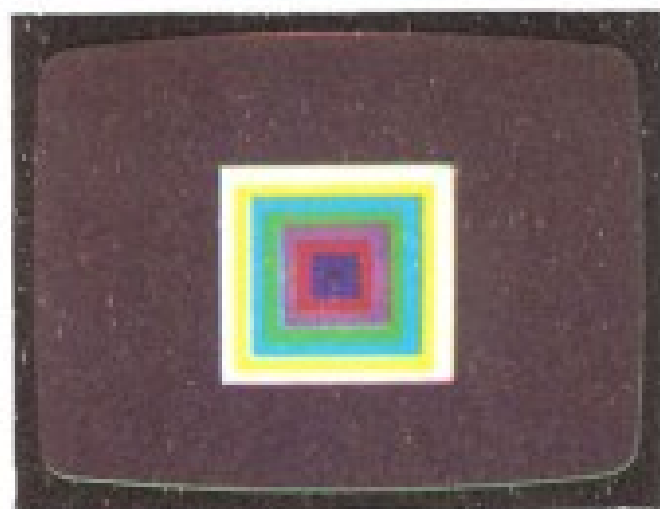
## COMO DISEÑAR SUS PROPIAS TRAMAS Y DIBUJOS

Es posible producir toda clase de tramas y dibujos con su Spectrum mediante gráficas de baja resolución, gráficas de alta resolución o ambos métodos. El mejor procedimiento para dibujar con su computadora es hacer previamente un borrador empleando copias de la cuadrícula en la última página.

Es posible utilizar ciclos FOR NEXT para dibujar. Estos ciclos repiten una parte del programa total, la cantidad de veces indicada. El programa puede ser redactado de tal forma que cada uno de los ciclos cambia las posiciones y colores de los caracteres o la línea, usualmente de acuerdo a un plan regular preestablecido. El siguiente programa hace uso de esta técnica.

### CUADRADOS

```
10 BORDER 0: PAPER 0: CLS
20 FOR x=7 TO 0 STEP -1
30 INK x
40 FOR l=11-x TO 11+x
50 FOR c=10-x TO 10+x
60 PRINT AT l,c: "■"
70 NEXT c
80 NEXT l
90 NEXT x
```



El programa contiene tres ciclos (o en inglés "loops") FOR... NEXT. El ciclo x cambia el color y permite el tamaño de los cuadrados grandes producidos por el programa, mientras que el ciclo l y el ciclo c cambian la posición de la línea y la columna de los cuadrados pequeños cada vez que son visualizados en la pantalla.

### Efectos al azar o aleatorios y subrutinas

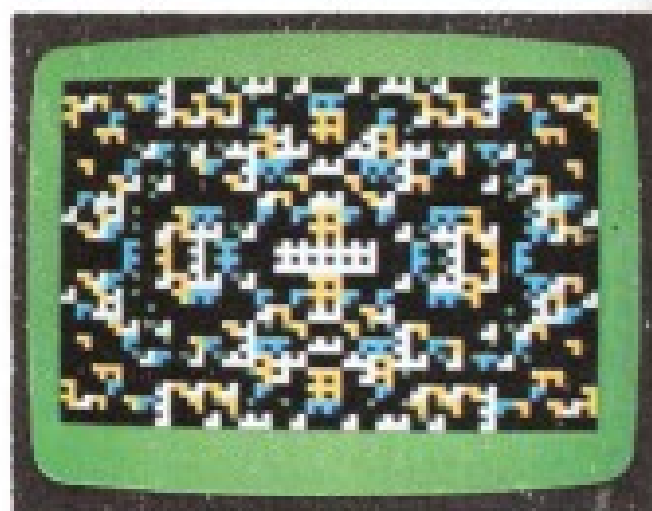
Los ciclos no le limitan a que las figuras sean idénticas cada vez que el programa es ejecutado. Al incluir la palabra clave RND (una abreviación de RaNDom) en el ciclo es posible cambiar los colores, posiciones de los elementos del dibujo o demás características, cada vez que el programa es ejecutado. Examine el programa para el mosaico en la página 18. El programa contiene la instrucción RND\*7 que controla el color de la tinta seleccionando cualquier número con un punto

ese número al número entero más próximo. El resultado es que, cada vez que la computadora visualice un cuadrado, su color variará entre INK 0 e INK 7.

El siguiente programa dibuja figuras simétricas formadas por caracteres gráficos en la pantalla. El programa emplea la palabra clave RND para cambiar los caracteres y sus posiciones. Las variables i y p determinan el valor de la tinta y el papel. La variable a determina la cantidad de figuras que serán dibujadas por la computadora (en este caso cuatro figuras). La variable n establece la cantidad de caracteres en cada figura, mientras que la variable x es un número seleccionado al azar por la computadora entre 129 y 142. La sentencia GOSUB 1000 en la línea 50 dirige la computadora a una subrutina, que él indica que al llegar a esta línea del programa debe dirigirse a una subrutina, definida en la línea 1000 y siguientes del programa.

### FIGURAS SIMÉTRICAS

```
10 BORDER 4: PAPER 0: CLS
20 LET i=4: LET p=0
30 FOR a=1 TO 4
40 LET x=RND*13+129
50 FOR n=1 TO 40: GO SUB 1000:
NEXT n
60 LET i=i+1: LET p=p+1
70 PAUSE 100
80 NEXT a
90 STOP
1000 LET l=INT (RND*11)
1010 LET c=INT (RND*10)
1020 INK i: PAPER p
1030 PRINT AT l,c:CHR$ x
1040 PRINT AT l,31-c:CHR$ x
1050 PRINT AT 21-l,c:CHR$ x
1060 PRINT AT 21-l,31-c:CHR$ x
1070 DEEP 0.01:l<c>0
1080 RETURN
```



Una subrutina es un grupo de líneas de programa que actúan como un pequeño programa incluido dentro del programa principal. En nuestro ejemplo, la subrutina se encuentra en la línea 1000. La subrutina visualiza un carácter gráfico en cuatro puntos de la pantalla, de forma tal que cada uno de ellos se encuentra a la misma distancia del centro (posición 11,10). La distancia desde el centro es determinada por las líneas de programa 100 y 1010, la variable i indica la variable en hileras, y la variable c la distancia en columnas. La palabra clave INT modifica el número seleccionado al azar, transformándolo en cualquier número entre 0 y 10 para la variable i, y en cualquier número entero entre 0 y 15 para la variable c. A continuación, las líneas de programa 1030 - 1060 visualizan el carácter gráfico cuyo código es x (consulte la tabla de caracteres en la página 51). La palabra clave DEEP

genera un sonido cuya intensidad está relacionada con la posición. Finalmente, la palabra clave RETURN, en la línea de programa 1000 cierra la subrutina y envía nuevamente a la computadora la instrucción inmediatamente después de la palabra GOSUB, en la línea 50.

La página 6 varía lo colores de la tinta y papel. La palabra clave PAUSE 100 (pausa) en la línea 70 demora la ejecución del programa por un período de dos segundos, antes de que la computadora reinicie el ciclo. La palabra clave STOP (alto) en la línea 90 es una advertencia para la computadora que le advierte que no debe continuar directamente a la subrutina después del cuarto ciclo.

Es posible modificar el programa, alternando el número 4 en la línea 30, y el número 50 en la línea 50, si aumenta la amplitud de x (línea del programa 40) se cambian los caracteres que aparecen en la pantalla.

### Empleando los ciclos de FOR NEXT para efectos gráficos

Los ciclos FOR NEXT pueden ser utilizados con gran eficacia para producir efectos gráficos de alta resolución. Ingrese mediante el teclado el programa siguiente y ejecútelo. Utilizando solamente las palabras clave PLOT y DRAW, los dos ciclos FOR NEXT, primero dibujan líneas y luego cinco triángulos o pirámides sólidas.

#### PIRAMIDES

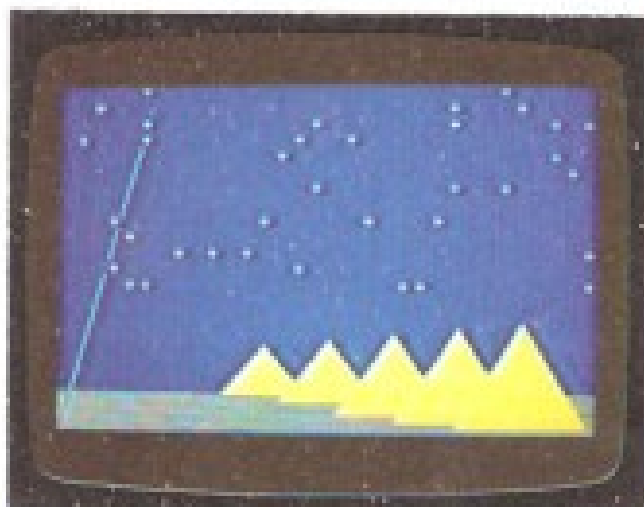
```
10 BORDER 0: PAPER 1: INK 0
20 CLS
30 FOR V=0 TO 20 STEP 2
40 PLOT 0,V
50 DRAW 255,0
60 NEXT V
70 FOR N=100 TO 200 STEP 30
80 FOR X=-10-N/10 TO 10+N/10
90 PLOT N,35+N/10
100 DRAW X,-N/4
110 NEXT X: NEXT N
```



Ahora agregue las líneas en la columna siguiente. Cuando ejecute nuevamente el programa encontrará que un rayo laser dispara en el cielo nocturno, creando una multitud de estrellas. El rayo es dibujado a partir de la esquina de la pantalla, a la posición x, y. Estas dos últimas variables (x, y) son números elegidos al azar por su computadora.

Tome nota del empleo de la palabra clave OVER 1; (incluyendo el ) en las líneas 150 y 180, que son idénticas. OVER 1 permite a la primera línea dibujar el rayo laser, y la segunda línea borra, "sin" modificar el resto de la figura.

```
120 LET X=RND*255
130 LET Y=RND*100+71
140 LET I=INT 175-Y/10
150 LET J=INT 10/5
160 PLOT 0,0: DRAW OVER 1,X,Y
170 REPEAT 0,0: X/4
180 PLOT 0,0: DRAW OVER 1,X,Y
190 PRINT AT 1,0:"+"
200 GO TO 120
```



### Las palabras clave FLASH, BRIGHT e INVERSE

Estas palabras clave son muy útiles para controlar los colores de su Spectrum. Cada una de ellas es seguida por los números 0 o 1. Es posible incluirlas en las sentencias conteniendo la palabra clave PRINT, siempre y cuando agregue un punto y coma (;) después del 0 o el 1. FLASH 1 hace parpadear las posiciones de los caracteres entre los colores de la tinta y el papel. BRIGHT 1 hace que los colores sean más brillantes. INVERSE 1 cambia el color de la tinta al del papel y viceversa. Cuando se ingresa una de estas tres palabras clave seguida del número 0, se causa el efecto opuesto, la pantalla vuelve a la situación normal. (El número 1 después de la palabra clave inicia el efecto especial y el número 0 lo termina).

Pruebe ingresar los cambios siguientes en los programas en estas dos páginas, para aprender a utilizar las palabras clave. En el programa de los cuadrados, modifique el cuadrado en la línea 60 transformándolo en una estrella y a continuación agregue el comando

```
15 INVERSE 1
```

Como resultado las estrellas aparecerán en color negro (el color del panel), resaltando contra las franjas de colores (los colores cambian de la tinta). Ingrese el comando INVERSE 0 antes de continuar.

En el programa de las Figuras Simétricas, agregue las líneas a continuación, para apreciar el efecto de la palabra clave BRIGHT y FLASH.

```
15 BRIGHT 1
16 FLASH 1
```

Observe cómo FLASH hace "parpadear" las figuras, lo que produce el efecto de que estas parezcan moverse en la pantalla. Ingrese la sentencia FLASH 0:CLS para interrumpir el efecto de "parpadeo" de las figuras en la pantalla.



las palabras clave READ y DATA. El primer paso es transformar los números binarios en números decimales. Puede hacerlo ingresando por el teclado de su computadora el comando PRINT BIN seguido por el número binario, por ejemplo:

```
PRINT BIN 00111100
```

En respuesta su Spectrum visualizará el número decimal 60, que es el equivalente del binario 00111100. En el ejemplo de nuestra araña, los ocho números decimales son: 60, 126, 219, 255, 189, 165, 165 y 36.

Ahora puede utilizar las palabras READ y DATA. Ambas le suministran un procedimiento sencillo para alimentar entidades de valores, como números, en las variables del programa. READ es seguido de una variable — una o más letras si está trabajando con números, o una sola letra seguida del símbolo \$ cuando se trata de cadenas de caracteres (strings). Cuando su Spectrum encuentra una palabra clave READ, inmediatamente se dirige a la primera sentencia con la palabra clave DATA en el programa. Estas sentencias contienen una lista de valores separados por comas. El procedimiento seguido en este caso es el siguiente: —la computadora toma el primer valor en la lista— retorna a la línea del programa donde había encontrado la palabra clave READ —y pone ese valor en la primera variable que encuentra después de la palabra clave READ.

La computadora continúa ejecutando el programa, y la próxima vez que encuentre la palabra clave READ, repetirá el procedimiento, pero esta vez tomará el segundo valor en la lista, y así sucesivamente.

El siguiente es el nuevo programa para producir la araña.

```
100 FOR A=0 TO 7
110 READ V
120 POKE USA, "A"+A,V
130 NEXT A
140 DATA 60,126,219,255,189,165,165,36
```

Este programa puede almacenar cualquier grupo de ocho números decimales en la memoria de la computadora para crear un carácter. Bastará para ello, con cambiar la letra s en la línea 30 por la letra que usted prefiera, e ingresar ocho números en la línea 50, después de la palabra clave DATA, separados entre sí por comas. Acto seguido oprima la tecla GRAPH y la letra elegida para obtener el carácter gráfico deseado al ejecutar el programa.

## Dibujando un tablero de ajedrez

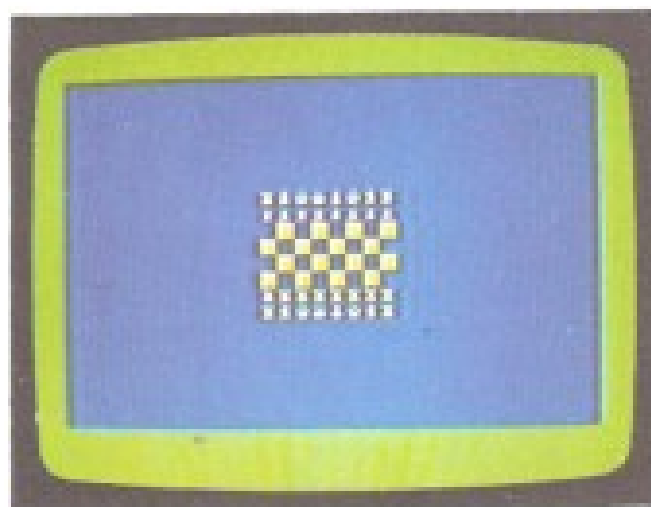
El programa visualiza un tablero de ajedrez en la pantalla, y a continuación dibuja las piezas. Para seleccionar los colores consulte la tabla de códigos de control en esta página.

### TABLERO DE AJEDREZ

```
10 FOR A=1 TO 8
20 READ AA
30 GO SUB 500
40 NEXT A
50 BORDER 4, PAPER 1, CLR
60 FOR L=7 TO 14 STEP 2: FOR C
70 TO 19 STEP 2
80 PRINT AT L,C: " " AT L+1,C:
90 NEXT C NEXT L
100 PRINT AT 7,10: "RNBCKBNR"
110 PRINT AT 8,10: "PBPBPBP"
120 PRINT AT 13,10: "PBPBPBP"
130 PRINT AT 14,10: "RNBCKBNR"
140 GO TO 100
150 FOR A=0 TO 7
160 READ V
170 POKE USA, "A"+A,V
180 NEXT A
190 RETURN
```

Las piezas son definidas por una subrutina que comienza en la línea 500.

```
500 DATA "P",0,0,16,55,55,16,17
600 DATA "R",0,64,124,55,55,124,0
700 DATA "N",0,16,55,120,24,55,124,0
800 DATA "B",0,16,40,55,100,55,124,0
900 DATA "K",0,16,55,16,55,55,0,0
1000 DATA "Q",0,64,40,16,100,124,124,0
```



### Utilizando los códigos de control para colores

En lugar de usar palabras clave como INK y PAPER, se pueden poner códigos de control después de las primeras comillas. Los caracteres a ser impresos cambiarán de color en el listado mismo y también aparecerán de este color en la pantalla. Para obtener estos códigos oprima

simultáneamente CAPS SHIFT y SYMBOL SHIFT para seleccionar el modo extendido y luego la tecla numérica correspondiente (con o sin CAPS SHIFT). Este diagrama le muestra cómo seleccionar las combinaciones de color:



# DIBUJOS ANIMADOS

Los elementos gráficos de la computadora son ideales para producir caracteres o líneas con movimientos en la pantalla. Producir estos dibujos animados con su Spectrum no es difícil. Todo lo que se requiere es cambiar continuamente la posición de la pantalla donde la computadora visualiza los caracteres o dibuja la línea. El mejor procedimiento es utilizar uno o más ciclos FOR NEXT.

## Movimientos verticales y horizontales

Ingrese en la computadora el programa siguiente. Si su Spectrum todavía conserva almacenado en su memoria el programa para producir la araña, descrito en la sección anterior (página 32), entonces no será necesario volver a ingresar las líneas 10-50. El carácter gráfico generado por el usuario en aquella oportunidad todavía se encontrará disponible y podrá ser llamado mediante la letra "s".

## LA ARAÑA SALTADORA

En este programa la araña desciende a lo largo de la pantalla colgada de su hilo, cada vez que se ejecuta el programa.

```
5 BORDER 3, PAPER 5: CLS
10 FOR x=0 TO 7
20 READ v
30 POKE USR "-s"+x,v
40 NEXT x
50 DATA 60,120,210,255,100,100
100 30
60 FOR x=0 TO 7
70 READ v
80 POKE USR "-1"+x,v
90 NEXT x
100 DATA 15,15,15,15,15,15,15,1
110 FOR i=0 TO 20
120 PRINT AT 1,3: INK 0; "-1"
130 PRINT AT 1+1,3: INK 2; "s"
140 NEXT i
```

Las líneas 60-100 del programa producen otro carácter gráfico ("1"), para el hilo. La animación se produce entre las líneas 110 y 140. Estas construyen un ciclo FOR NEXT, en el cual la variable i (el número de la línea) cambia de 0 a 20. Cada vez que el ciclo se repite, la computadora dibuja una parte del hilo de la araña en la pantalla, en una posición determinada, y la araña es dibujada precisamente en la posición inferior siguiente. En el próximo ciclo la araña anterior es sustituida por un sector de hilo y la computadora dibuja una nueva araña un poco más abajo. Mediante estos ciclos que se repiten rápidamente, se produce la ilusión de que la araña baja colgada de su hilo, hasta que llega a la línea 21, que es la parte inferior del área de visualización.

Para disminuir la velocidad de la acción, ingrese la línea siguiente:

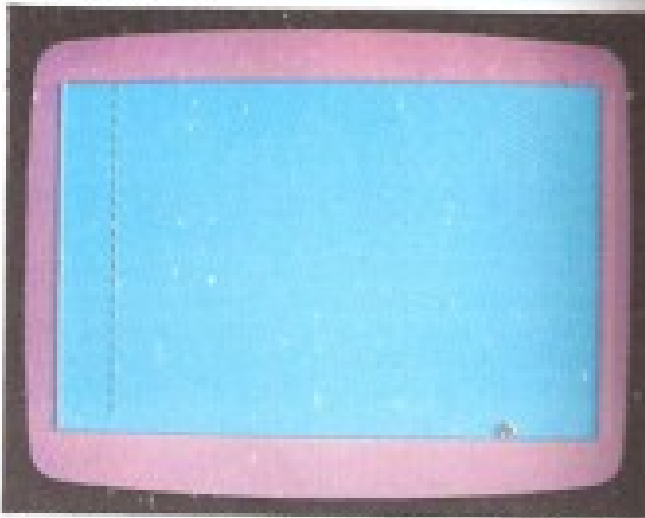
```
135 PAUSE 10
```

El efecto de esta sentencia es causar que la computadora espere un quinto de un segundo cada vez, antes de dibujar la araña en la posición siguiente.

Ahora le sugerimos agregar las siguientes líneas extra al programa anterior, para producir una

## EL ARACNIDO

```
150 FOR c=3 TO 30
160 PRINT AT 21,c: "-"
170 PRINT AT 21,(+1): INK 2; "s"
180 NEXT c
```



Esta vez, nuestra amiga comienza a correr hacia uno de los lados de la pantalla tan pronto como toca el suelo. El efecto se logra agregando un nuevo ciclo FOR NEXT al programa. Este ciclo controla la posición horizontal c de la figura medida en columnas. Observe que primero se visualiza en la pantalla un espacio y que la araña es dibujada en la posición correspondiente a la columna siguiente. El resultado es que la araña desaparece de una posición y aparece en la siguiente, lo que produce la ilusión de movimiento de derecha a izquierda. Siempre es mejor borrar un carácter antes de volverlo a visualizar en la posición siguiente, para evitar que las figuras animadas parezcan oscilar en la pantalla.

## Tiro al blanco

En muchos juegos de computadora, uno de los elementos más importantes de la acción es el choque entre dos figuras, o cuando un objeto es tocado por un rayo, por ejemplo un rayo de laser.

Es fácil detectar el punto de colisión. Si dos caracteres son visualizados por la computadora en la posición i,c (por línea y columna) y en la posición v,h (por vertical y horizontal) respectivamente, entonces es posible asumir que si i=v y c=h entonces ambos caracteres se encuentran en la misma posición. Esta identidad puede ser redactada como una sentencia en su programa de la computadora utilizando las palabras clave IF THEN (en este caso unidas por otra palabra clave: AND)

```
160 IF i=v AND c=h THEN PRINT "CRASH!"
```

(es decir, si la variable c es igual a la variable v, y si la variable v es igual a la variable h, entonces visualice lo siguiente en la pantalla del televisor: "CRASH!")

Otro procedimiento para determinar si dos figuras entran en colisión es utilizar colores. Borne el programa de la araña mediante la palabra clave NEW. A continuación ingrese el programa completo detallado en la página 31, o cárguelo en la computadora si ha tomado la precaución de almacenarlo en un cassette. Ahora podemos perfeccionar este programa combinándolo con el programa de la araña (el programa de la araña continúa en la memoria de la computadora, excepto naturalmente si usted oprimió el botón de reajuste

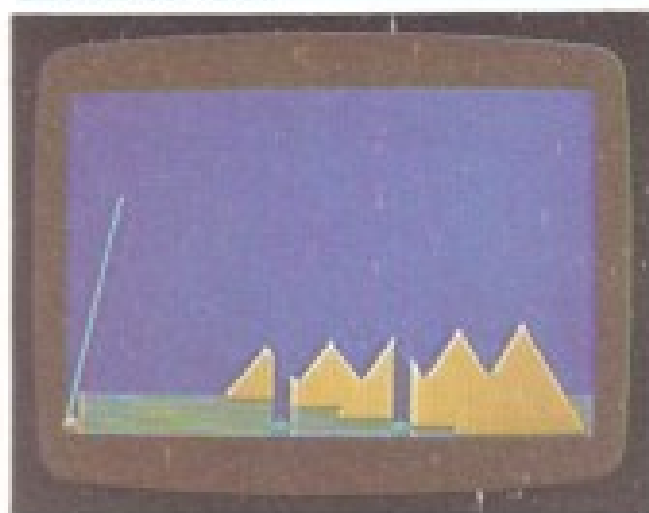
En primer lugar, agregue las líneas siguientes, las cuales crean una gráfica representando una explosión denominada "e".

```
5 FOR V=0 TO 7
6   ACOS V
7   POKS USR "E"=X,V
8 NEXT V
9 DATA 140,62,44,121,155,52,7
10,137
```

A continuación borre la línea 190 y agregue o cambie las líneas siguientes.

## ARAÑAS Y PIRAMIDES

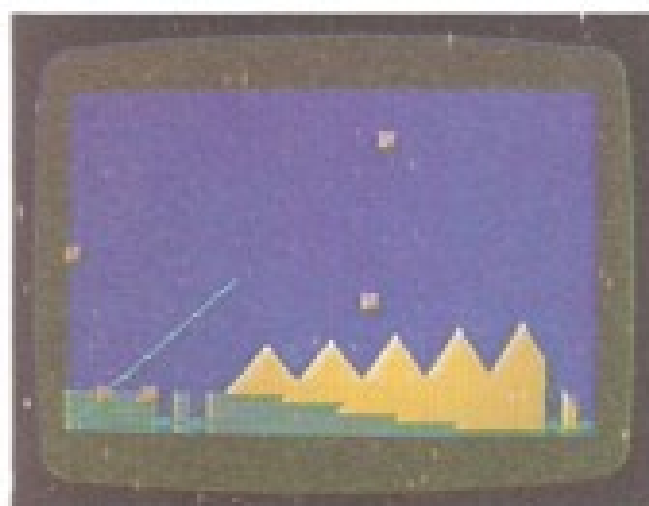
```
114 LET A=AND(21)
115 FOR V=0 TO 20
116   POINT AT V,A: AT V+1,A
117 NEXT V
118 GO TO 114
```



Debido a los cambios que hemos realizado, el programa ya no genera estrellas, en cambio ahora tenemos voraces arañas que caen sobre las pirámides y el suelo consumiéndolos rápidamente. Lo que ha sucedido es que hemos agregado un nuevo ciclo FOR NEXT en el cual las variables v y h suministran la posición de la araña. La variable h es aleatoria o random, lo que causa que las arañas aparezcan en diferentes posiciones de la pantalla. Ahora agregue las líneas siguiente:

## ARAÑAS EXPLOSIVAS

```
190 IF ATTR (V+1,A)=14 THEN GO
TO 500
200 PRINT AT V+1,A: FLASH 1: PA
PER 2,"E"
210 PAUSE 100
220 GO TO 114
```



Cuando el rayo laser hace impacto en una de las arañas, esta se torna amarilla. El procedimiento para

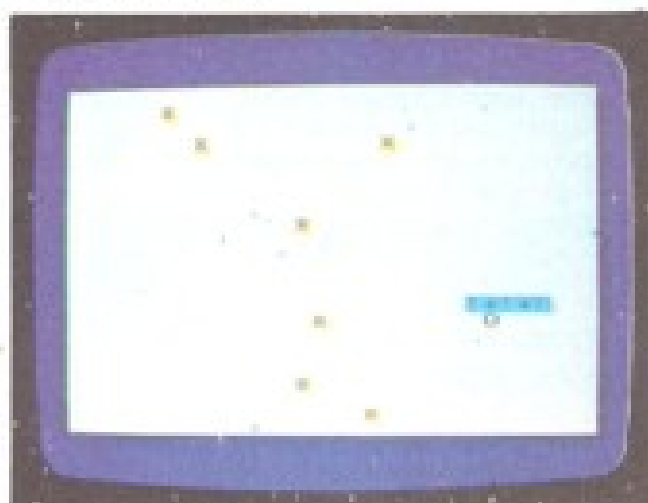
lograr este efecto es el siguiente: cuando la línea producida por DRAW entra en la posición del caracter representando la araña, se produce un cambio en el color de la tinta de este último, y pasa a ser el mismo color que la línea (amarillo en este caso). En la línea 190 del programa, ATTR detecta si la araña se torna amarilla. Cuando esto sucede la instrucción le indica a la computadora que debe dirigirse a la subrutina de la explosión, en la línea 500.

## SIMULANDO LOS REBOTES DE UNA PELOTA

Muchos de los programas con efectos gráficos muestran figuras que rebotan contra los bordes de la pantalla. Este programa le explicaría como lograr tal efecto. Las variables v y h operan de la misma manera que en el programa de las arañas explosivas, pero se agregan los valores +1 o -1 a las variables v y h para hacer que la pelota se mueva hacia arriba o abajo o hacia la derecha o izquierda. La sentencia SCREEN\$ comprueba si existe un X en la posición v,h.

## PELOTA REBOTANDO

```
10 BORDER 1
20 FOR I=1 TO 10
30 LET A=INT (RND*25): LET V=I
40 PRINT INK 2: PAPER 6: FLASH
1: AT V,A: "X"
50 NEXT I
60 LET V=-1: LET H=1
70 PRINT AT V,H: "X"
80 LET V=V+V: LET H=H+H
90 IF A=0 OR H=21 THEN LET V=-
V: IF V=0 OR V=21 THEN LET H=-
H: IF SCREEN$ (V,H)="X" THEN P
RINT INK 1: PAPER 6: AT V,H: "X"
100 STOP
110 PRINT AT V,H: "O"
120 PAUSE 2
130 GO TO 70
```



## Utilizando atributos

La palabra clave ATTR detecta los atributos o características de una posición determinada en la pantalla. Los atributos son los colores de la tinta o el papel, y si la posición se encuentra parpadeando (FLASH) o brillante (BRIGHT). En el programa de las arañas explosivas, la palabra clave ATTR asegura que cuando la araña cambia de color, tornándose amarilla, la computadora la haga desaparecer. Su color de tinta ahora es amarillo (número 6). El color de papel es azul (número 1). Esto significa que el total de atributos de esa posición es en total 14 (6 por la tinta amarilla más 8 por el papel azul). La sección correspondiente en la "Guía de Referencia del Programador" le explicará como se llega a estos números.

# UN INSTRUMENTO MUSICAL Y EFECTOS SONOROS

La Spectrum posee un sintetizador de sonidos capaz de producir una gran variedad de sonidos musicales y efectos sonoros, que pueden ser utilizados para dar vida a sus programas. El sintetizador es fácil de usar, y no requiere conocimientos musicales. El sintetizador produce una señal de sonido al altavoz interno de su Spectrum.

### Programando sonidos

Para producir sonidos en su Spectrum, solamente se emplea una palabra clave: BEEP. Esta palabra clave es seguida por dos números o variables representando números. El primer número o variable le indica a la computadora cuál es la duración (en segundos) del sonido, y el segundo le hace saber cuán agudo o grave debe ser el sonido. El tono es medido en semitonos. Los valores del tono son 0 para do central; 1 para do sostenido; - 1 para si (do bemol), etc.

Ingrese ese programa en su computadora y ejecútelo.

La Spectrum producirá toda la gama de sonidos de que es capaz, desde las notas más altas o agudas (66) hasta las más bajas o graves (- 66). Como puede apreciar las notas más altas son casi inaudibles, las más bajas suenan como "clicks".

```
10 FOR I=66 TO -66 STEP -1
20 BEEP .02,I
30 PRINT AT 8,2, "  " AT 8,14
40 NEXT I
```

La tabla de abajo le indicará cuales son los valores de tono correspondientes a una amplia gama de notas. Con esta información es posible transformar música escrita en parte de un programa para su computadora Spectrum.

### Efectos sonoros

La computadora Spectrum tiene la capacidad de producir una amplia variedad de efectos sonoros, usualmente mediante un ciclo incluyendo la palabra clave BEEP que cambia rápidamente el valor de tono. Pruebe los programas siguientes. Observe que los valores determinando la duración son muy breves, como de la centésima parte de un segundo. Oprima la tecla BREAK para terminar los programas conteniendo los ciclos.

#### BURBUJE0

```
10 LET P=INT (RND*40) -30
20 BEEP .005,P: BEEP .005,P+7:
BEEP .005,P+1
30 GO TO 10
```

Este programa interpreta un grupo de tres notas, repitiéndolas varias veces mientras varía los tonos al azar. La amplitud de los tonos es considerable, y se la puede modificar, simplemente alterando los valores en la línea 10 del programa.

#### MAQUINAS

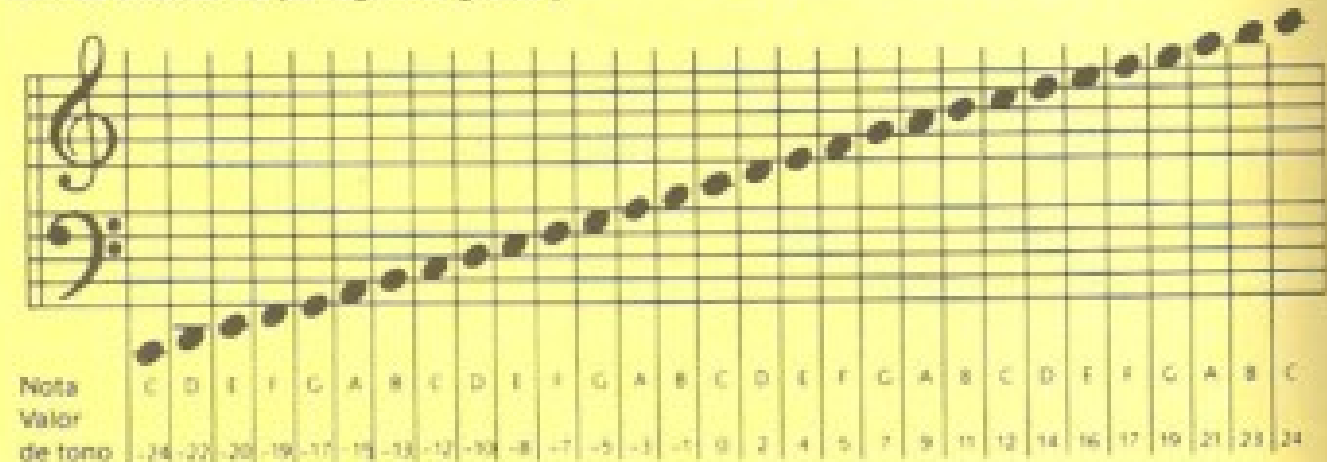
```
10 FOR I=10 TO 30
20 BEEP .01,I
30 BEEP .01,34-I
40 NEXT I
50 GO TO 10
```

El programa produce dos sonidos, el tono de uno aumenta mientras que el del otro baja. Este efecto es el resultado de la combinación de dos sentencias en el programa con la palabra clave BEEP. Las sentencias correspondientes repiten dos notas de tonos diferentes con una separación de apenas un centésimo de segundo.

### Valores de tono para efectos musicales

Los siguientes son los valores de los tonos para su Spectrum, desde los tonos más bajos hasta los más altos de los pentagramas graves y

agudos. Agregar 1 al valor del tono para una nota aguda y restar 1 para una nota grave.



## ESCALA

```
10 FOR P=1 TO 48 STEP 0.2
20 BEEP .01:P=DEP*.01:P=.5
30 NEXT P
```

Este programa es similar al programa "Máquinas" anterior, pero ahora las dos notas aumentan conjuntamente, con una separación de seis semitonos. Además los valores de tono cambian en pasos de 0.2 —un quinto de un semitono— cada vez, lo que determina que el tono del sonido se incremente más lentamente. Pruebe cambiar la magnitud de los cambios de tono alterando el valor de la instrucción STEP (en inglés STEP significa "paso").

## TECLADO MUSICAL

```
10 LET P=CODE INKEY$
20 IF P=0 THEN GO TO 10
30 BEEP .04:P=P*20.75
35 GO TO 10
```

Este programa adjudica un sonido diferente a cada una de las teclas en el teclado de su computadora. Si se oprime la tecla CAPS SHIFT mientras se mantiene oprimida otra tecla, el sonido baja. El programa hace uso de la sentencia CODE INKEY\$ que atribuye un valor diferente a la variable p cada vez que se oprime una nueva tecla.

## Sonido y visión

Los efectos de sonido producidos por su Spectrum pueden ser combinados con la acción que se desarrolla en la pantalla del televisor. Para demostrar como agregar sonido a un programa, examinaremos un ejemplo. Volvamos al programa del "Arácnido" en la página 34.

Recuerde que insertamos una sentencia de PAUSE en la línea 135, para disminuir la velocidad de la acción. En lugar de reducir la velocidad del programa de esta forma, es posible programar una pausa que produce sonidos. Cambie la línea 135 del programa como se indica a continuación.

135 GOSUB 500

A continuación agregue las siguientes líneas al programa

```
200 STOP
300 FOR P=40-1 TO 30-1 STEP -1
510 BEEP 0.02:P
520 NEXT P
530 RETURN
```

Ejecute el programa y verá como la araña hace un sonido especial cuando desciende. La nueva subrutina controlando el sonido le indica a la computadora que emita tres notas muy rápidas, cuyo tono baja a medida que la araña desciende a la próxima posición en la pantalla. Pruebe agregar más notas, cambiando el contenido de la línea 500. También es posible aumentar o disminuir la velocidad de las notas, alterando el valor 0.02 en la 510.

## Como amplificar los sonidos producidos por su Spectrum

Es posible conectar el enchufe de EAR o MIC a un amplificador o un altavoz para incrementar el volumen de los sonidos producidos por su Spectrum.

El procedimiento más sencillo es utilizar el cable del cassette de la Spectrum para conectar el punto de EAR o MIC al enchufe MIC de un grabador en cassettes. Si fuera necesario extraiga el cassette del grabador. Ponga en marcha el grabador de cassettes y oprima los controles PLAY, REWIND (REVERSE), o FAST FORWARD (CUE) (AVANCE RAPIDO). Ajuste el control de volumen del grabador. Ahora debe oírse el sonido producido por la computadora, emitido por el

altavoz del grabador. También es posible utilizar audifonos con el grabador.

Otra posibilidad es conectar su Spectrum a un equipo de música de alta fidelidad, si desea obtener sonidos de alta calidad. En este caso, necesitará un cable especial con un enchufe macho de 3,5 mm, para conectar su Spectrum y un enchufe para la conexión al equipo de música. La Spectrum produce una señal de línea similar a la señal de salida del grabador de cassettes, por lo tanto debe ser posible conectar su computadora a los puntos de conexión REPLAY o LINE IN del equipo de música.



## COMO GRABAR PROGRAMAS

Tarde o temprano encontrará que es conveniente almacenar sus programas en cintas de cassette. El primer paso es conectar un grabador a su Spectrum, para luego proceder a grabar (SAVE) el programa que se encuentra en la memoria de la computadora. La ventaja de grabar y almacenar los programas en cassettes es obvia: en el futuro cuando los vuelva a necesitar, utilizando el procedimiento descrito en las páginas 14 y 15. El propósito de estas páginas es enseñarle a grabar programas y también a verificar que el programa ha sido grabado correctamente.

### Grabando sus propios programas

**1** Primero conecte su Spectrum con un grabador de cassettes de tipo adecuado, utilizando con tal propósito el cable especial, tal como se describe en la página 14. Compruebe que solamente el enchufe MIC de la Spectrum se encuentra conectado al grabador de cassette.

**2** Si el grabador cuenta con controles de volumen o de nivel de la grabación, ajústelos a dos tercios del máximo. Si no cuenta con estos controles, no se preocupe, porque los niveles de la grabación se ajustarán automáticamente.

**3** Ingrese por el teclado de su computadora la palabra clave SAVE seguida por el nombre del programa que desea grabar entre comillas, por ejemplo

SAVE "prog2"

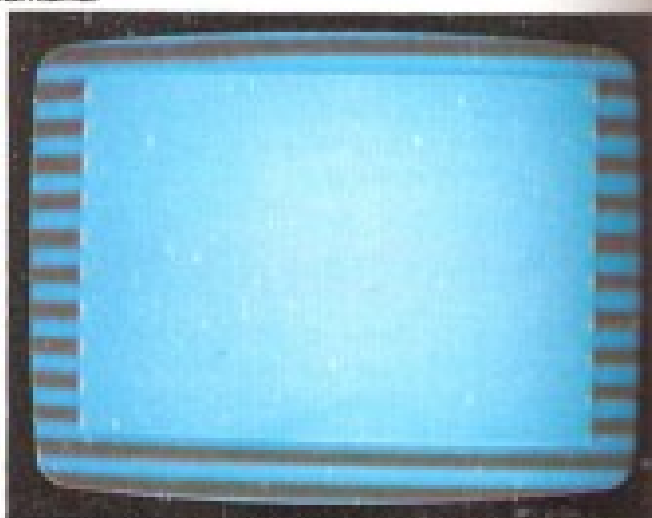


Es posible utilizar cualquier combinación de hasta diez letras o números para designar un programa. A continuación ingrese la palabra clave ENTER. La línea conteniendo la palabra SAVE desaparecerá de la pantalla y en su lugar podrá ver la instrucción de operación del grabador de su



**4** Ajuste los controles del grabador para grabar el programa. El procedimiento usual es presionar simultáneamente las teclas RECORD y PLAY del grabador. A continuación oprima cualquier tecla en su Spectrum.

**5** Espere mientras la Spectrum graba el programa. Al principio verá bandas de color rojo y azul ascendiendo lentamente en la pantalla.



Entonces la computadora proyecta una serie de franjas azules y amarillas. Esto indica que la Spectrum se encuentra enviando el nombre del programa a la cinta de cassette.

**6** Se produce un breve intervalo seguido otra vez por las franjas rojas y azules. Nuevamente aparecen las franjas amarillas y azules, cuando la Spectrum envía el programa a la cinta. La operación de grabar un programa extenso puede tomar varios segundos.



**7** Cuando el programa ha sido grabado en la cinta, su computadora se lo hace saber mediante el reporte @ OK,@:1. Ahora detenga la cinta. El programa ha sido grabado, y si lo desea puede controlarlo o "verificarlo".

#### "Verificando" el programa

Aunque ha transmitido el programa en su memoria a la cinta en el grabador de cassettes, usted no tiene la seguridad de que aquel fuera grabado correctamente. Afortunadamente, nuestra Spectrum puede verificar la calidad de la grabación.

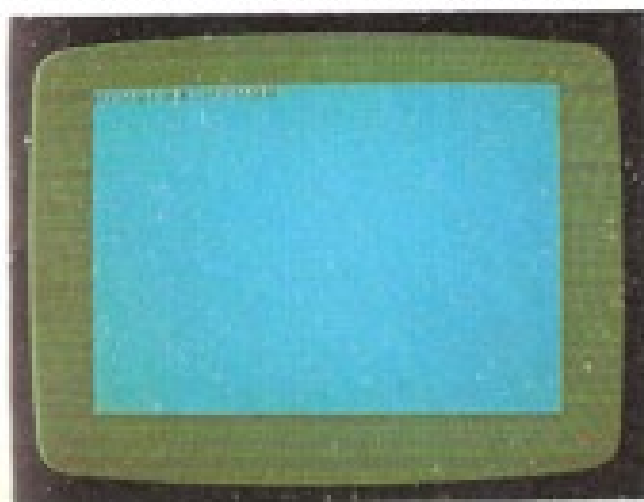
Este procedimiento se denomina "verificación". En primer lugar vuelva a rebobinar la cinta hasta el punto donde comenzó la grabación (utilice el contador de revoluciones de su grabador para este propósito). Luego se conecta el enchufe EAR de la Spectrum al enchufe EAR del grabador (no es necesario retirar la conexión entre los enchufes MIC). Una vez completada esta conexión, ingrese por el teclado la palabra clave VERIFY seguida por el nombre del programa entre comillas. Oprima la tecla ENTER y ponga en funcionamiento el grabador. A continuación se repetirá la secuencia de las bandas de colores en la pantalla del televisor: rojas y azules. Y luego azules y amarillas. Aparecerá el nombre del programa en la pantalla y permanecerá allí hasta que se complete el proceso de verificación.



Al final de la segunda serie de franjas azules y amarillas el siguiente informe aparecerá en la pantalla:

@ OK,@:1 Este informe significa que la Spectrum ha comparado el programa grabado en la cinta con el programa almacenado en su memoria, y verificado que son idénticos.

Ahora puede tener la seguridad de que su programa ha sido grabado correctamente en el cassette.



Si la computadora no visualiza el reporte @ OK,@:1, esto significa que la grabación no ha sido correcta. En este caso, deben tomarse las medidas siguientes: En primer lugar consulte la sección de Preguntas y respuestas sobre programas pre-registrados en la página 16. Recuerde que siempre es posible que el programa fuera grabado correctamente, y que la dificultad se deba a que no es posible volver a cargarlo en la computadora para la verificación. Examine el procedimiento de verificación y corrija cualquier error. A continuación vuelva a rebobinar la cinta y repita el procedimiento de verificación. Si la computadora todavía no verifica el programa, entonces le sugerimos que consulte la sección sobre Soluciones para los problemas que puedan presentarse al grabar programas en la página siguiente.

—Nota: No oprima la tecla NEW, ni apague la computadora. Si lo hace, perderá el programa almacenado en la memoria de la computadora sin tener la seguridad de que cuenta con una copia correcta en la cinta de cassette.

#### Indicando un programa automáticamente

Es posible agregar el nombre del programa (siempre entre comillas) a continuación de la palabra clave SAVE, seguido por la sentencia LINE 1, por ejemplo:

SAVE "ARACHIDO" LINE 1

En este caso el procedimiento para grabar el programa es similar al explicado anteriormente, pero al verificar la grabación, no debe incluir la sentencia LINE 1 después de VERIFY y el nombre del programa. La ventaja de agregar esta sentencia es que, en el futuro, al cargar el programa de la computadora, esta lo ejecutará automáticamente, tan pronto como lo cargue. En este caso no es necesario ingresar la instrucción RUN para indicarle a la computadora que ejecute el programa. (Recuerde detener el grabador una vez que el programa ha sido cargado en la computadora).

Lo que sucede en este caso es lo siguiente, el programa comienza en la línea 1, si la computadora encuentra una línea 1, entonces avanza la primera línea del programa. Cuando se cambia el número 1 a otro número, la computadora indicará el programa automáticamente en la línea con ese número.

#### Como almacenar sus programas en cassette

1. Escriba el nombre del programa en el rótulo del cassette o en su tarjeta en el momento de grabado. Utilice el mismo tipo de letras que aparecen en la pantalla del televisor (mayúsculas o minúsculas). Si su grabador tiene un contador de revoluciones, tome nota de las revoluciones antes de iniciar la grabación y después, lo que facilitará la tarea de ubicar el programa futuro.

2. Antes de grabar el programa, anote el nombre del programa en una línea de programa al principio de este, precedido por la palabra clave REM, por ejemplo

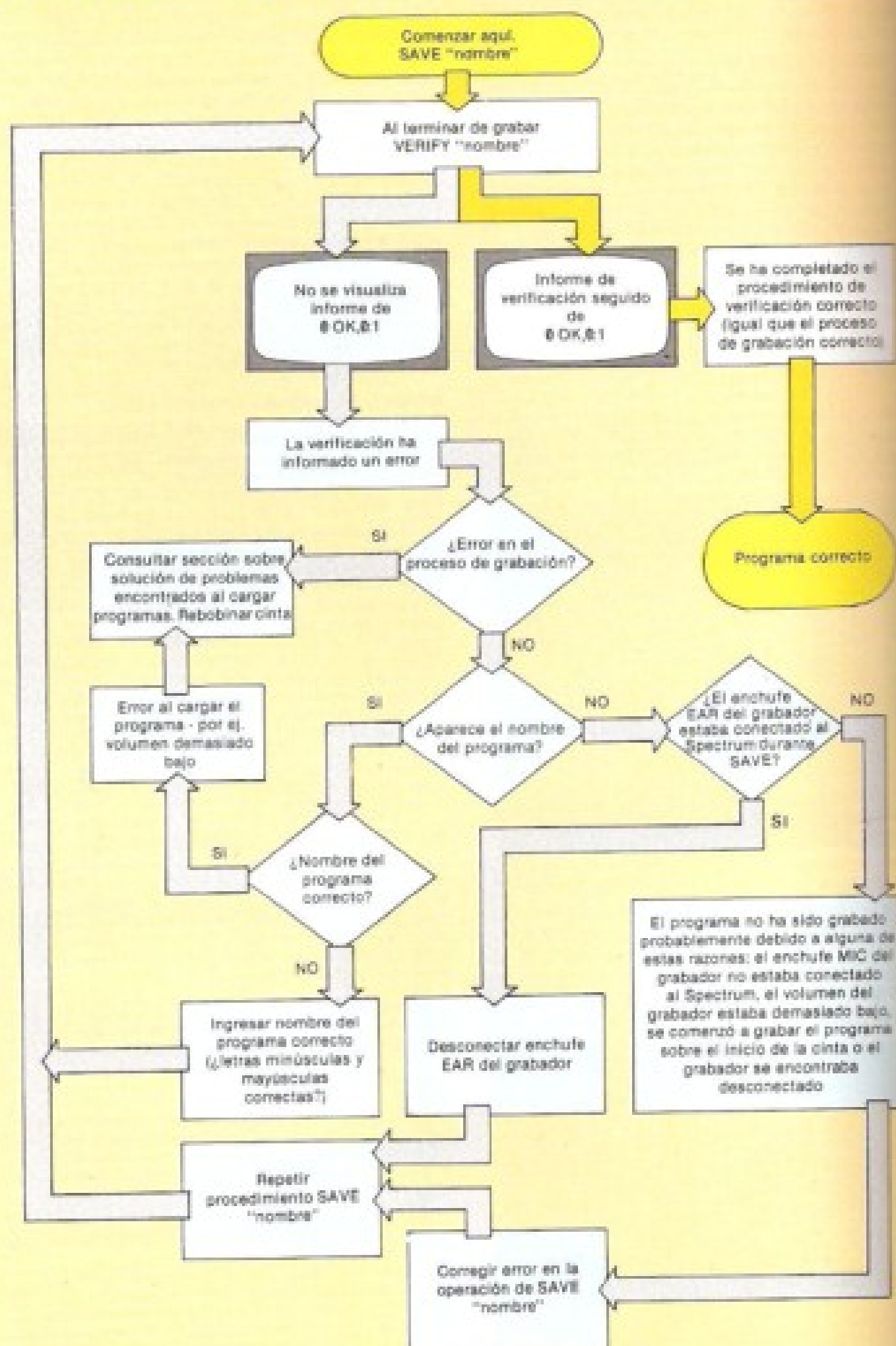
5 REM programa EL ARACHIDO. Versión 3

Al ejecutar el programa, la computadora ignorará todas las sentencias precedidas por la palabra clave REM. Esta palabra clave es muy útil porque le permitirá incluir comentarios y notas en su programa que le servirán como referencias en el futuro.

#### Grabando programas con las palabras clave CODE, SCREEN\$ o DATA

La palabra clave SAVE también puede ser utilizada en programas conteniendo las palabras clave CODE o SCREEN\$, para grabar una sección determinada de la memoria de la computadora, y conjuntamente con DATA para grabar una tabla (array). Consulte al respecto las secciones sobre SAVE CODE, SAVE SCREEN\$ y SAVE DATA, en la Guía de referencia del programador.

## Soluciones para problemas al grabar programas en cinta



## 3. APRENDIENDO ACERCA DE SU SPECTRUM

---

Este capítulo describirá los diversos componentes de su Spectrum y explicará su funcionamiento. También le explicaremos como utilizar equipos o accesorios "periféricos": estos son conectados a la computadora para incrementar su rendimiento y transformarla en un completo sistema de procesamiento de datos. Finalmente, tendrá la oportunidad de aprender más acerca de los aspectos técnicos de su computadora incluyendo la forma en que se organiza su memoria, y también le suministraremos información sobre las especificaciones técnicas de su Spectrum.

## COMPONENTES

Comenzaremos esta sección con un buen consejo: si desea aprender más de cerca de su Spectrum lea este capítulo, pero no abra su computadora. Recuerde que si la abre invalidará su garantía, y, lo que es peor, corre el riesgo de dañar alguno de los componentes.

En el interior de caja se encuentran dos conectores de cinta que interconectan el teclado de la computadora Spectrum con los demás componentes. Estos últimos vienen instalados en un único tablero de circuitos impresos. El tablero contiene elementos eléctricos de tipo standard, tales como resistencias y condensadores, pero sus elementos más importantes son los circuitos integrados rectangulares de color negro, dispuestos independientemente o en bloques.

### El interior de un circuito integrado

La parte funcional de un circuito es, en realidad, bastante más pequeña que la envoltura de plástico que la contiene. La envoltura del circuito es diseñada principalmente para albergar las conexiones requeridas por aquel. Estas conexiones permiten conectarlo a los puntos de conexión ubicados sobre el panel de circuitos. El circuito integrado en sí mismo es una pequeña placa de bifurcaciones eléctricas. Cada una de las bifurcaciones eléctricas actúa como un conmutador que permite detener, pasar, o almacenar las señales que llegan a él. El procedimiento es muy sencillo, pero existen tantas bifurcaciones operando conjuntamente en el mismo circuito integrado, que el sistema como un todo, puede producir señales complejas que almacenan o procesan información a una velocidad y precisión enormes.

### ¿Cómo se conectan los circuitos integrados con el resto del sistema?

Por lo tanto, en general, podemos decir que la Spectrum es un circuito eléctrico de gran complejidad. Cuando la computadora se encuentra en funcionamiento, un flujo continuo de pulsos eléctricos es transmitido a través de los senderos en el interior de cada circuito y entre estos y los demás componentes del sistema, para hacer funcionar la computadora.

Pero, ¿cómo es que se controlan estos procesos, para asegurar que la señal correcta llega al punto indicado, cuando se la necesita? Escondido en el interior de uno de los circuitos se encuentra el reloj de la computadora. El tic tac de este reloj es el resultado de una serie de pulsos de electricidad constantes.

Cuando la Spectrum se encuentra operando, al oprimirse una tecla se ponen en contacto un par de cables situados debajo del teclado. Esto genera una señal código que es enviada a la Unidad Procesadora Central (en inglés, Central Processing Unit o CPU).

### Unconmited Logic Array (ULA)

Este circuito genera las imágenes en la pantalla del televisor, partiendo de información almacenada en RAM y también opera como control del sistema.

### Punto de conexión del teclado

Uno de los conectores de cinta al tablero es conectado en este punto.

### Codificador de TV

Este componente cambia o traduce las señales producidas por los circuitos de la computadora en señales de color para su televisor.

### RAM (Random Access Memory)

Uno de los dos componentes principales de la memoria total de la computadora (el otro es el ROM). Es posible almacenar en ella programas u otro tipo de información. Este circuito contiene el programa que usted ingresa en la computadora y cualquier otra información requerida por ese programa, por

ejemplo valores para sus variables. La capacidad de memoria RAM de su Spectrum es 48K, y todo su contenido puede ser modificado desde el teclado de la computadora. También es posible borrar el contenido de la memoria RAM mediante el control de reajuste (reset) o apagando la computadora.

### Enchufes para el grabador de cassettes

Son empleados para transmitir información y programas desde el grabador a la memoria RAM de la computadora, o para transmitir desde la memoria a la cinta en el cassette para almacenarlos.

### Circuitos lógicos

Estos circuitos actúan como un punto de interconexión en el proceso de intercambio de información entre la CPU y la memoria RAM.

### Central Processing Unit o CPU (Unidad Procesadora Central)

Este es el cerebro de su computadora. El CPU de la Spectrum es un microprocesador Z80. La unidad CPU lleva a cabo todas las tareas de computación, cálculos y controla la operación total de la Spectrum.

**Enchufe de corriente eléctrica continua de 9 volt**  
Es la conexión a la fuente de alimentación

### Conector de borde

Este conector vincula su Spectrum con equipos externos, por ejemplo una impresora.

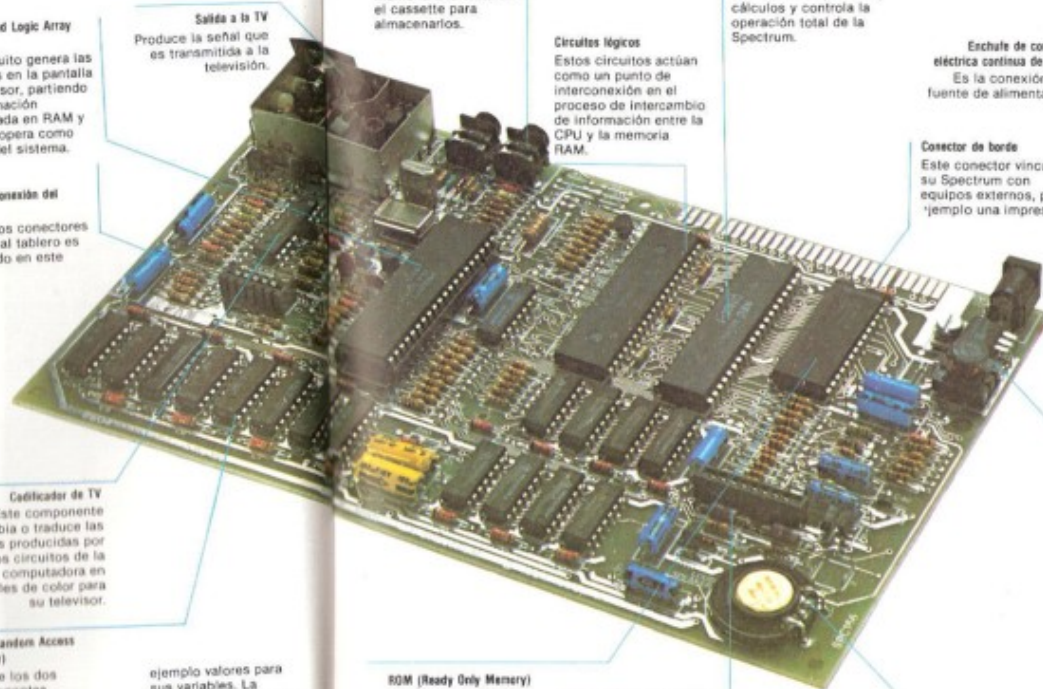
### Punto de conexión del teclado

Aquí se conecta uno de los conectores de cinta del tablero.

### Parlante

Este componente produce los sonidos.

**Regulador de tensión**  
Controla el voltaje para evitar cualquier cambio en la electricidad que pueda afectar a la computadora.



## ¿CÓMO FUNCIONA SU SPECTRUM?

Desde el punto de vista de su operación, la computadora Spectrum se compone de cuatro partes principales: unidades de ingreso de datos, como el teclado, mediante los cuales se ingresa la información necesaria, memorias temporarias y permanentes que almacenan la información, incluyendo programas, e instrucciones de operación; la Unidad Procesadora Central (CPU) que ejecuta las instrucciones en el programa, y las unidades de salida o output, al usuario.

### Ingresando y ejecutando un programa

¿Qué sucede en el interior de su Spectrum cuando usted ingresa y ejecuta un programa de una sola línea como el siguiente:

10 PRINT 6 + 2

Primero, usted ha operado el teclado. Debajo del teclado se encuentra una red de cables. Cada vez que una tecla es oprimida, un par de estos cables entra en contacto, lo que produce una señal que es enviada a la CPU. La CPU transmite el código a la memoria RAM, donde la señal es almacenada.

Al ejecutar el programa, la CPU recibe los códigos almacenados en la memoria RAM uno a uno, en el mismo orden que en el programa.

En nuestro ejemplo, la CPU recibirá en primer lugar el código para la palabra clave PRINT. Es un código particular que le indica a la CPU que debe tomar un código de operación especial guardado en su memoria ROM. Este código de operación es transmitido a la CPU, la cual se prepara para ejecutar las instrucciones. En este ejemplo la instrucción es sencilla: escribir algo sobre la pantalla del televisor. A continuación la CPU recibe el valor 6 contenido en la memoria RAM. Este valor también ha sido traducido al código. La CPU lo recibe y lo guarda en una pequeña memoria interna denominada "registro". La señal siguiente es el código para la suma, la CPU toma el valor 2 de la memoria RAM como en el caso del número 6. El valor 2 también ha sido traducido al código comprendido por la CPU y es sumado al valor 6 que se encuentra almacenado en el registro. El resultado es 8. La CPU traduce este resultado en otro código y finalmente, el número 8 es visualizado en la pantalla.

### Almacenando un programa

Cuando se le indica a su computadora que almacene el programa en una cinta de cassette, la CPU nuevamente toma los códigos de la memoria RAM. Pero en lugar de ejecutar las instrucciones, las envía a la unidad convertidora que los transforma en señales de sonido. Estas son las señales que serán transmitidas al grabador de cassettes y grabadas en la cinta.

Al cargarse el programa en la computadora, se produce el proceso inverso: las señales de sonido son transformadas en códigos de computadora por el convertidor.

### Códigos binarios

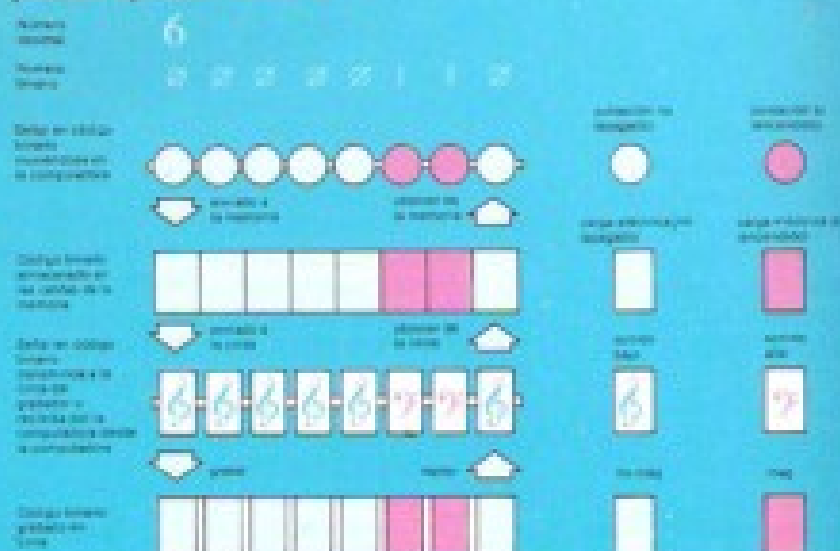
Todos los códigos que hacen funcionar su Spectrum son binarios. Se denominan binarios porque están compuestos por solamente dos tipos de señales. Estas señales pueden ser representadas por números binarios, esto es números que solamente contienen dos numerales: 0 y 1. Por ejemplo, el número binario para 6 es 0000110.

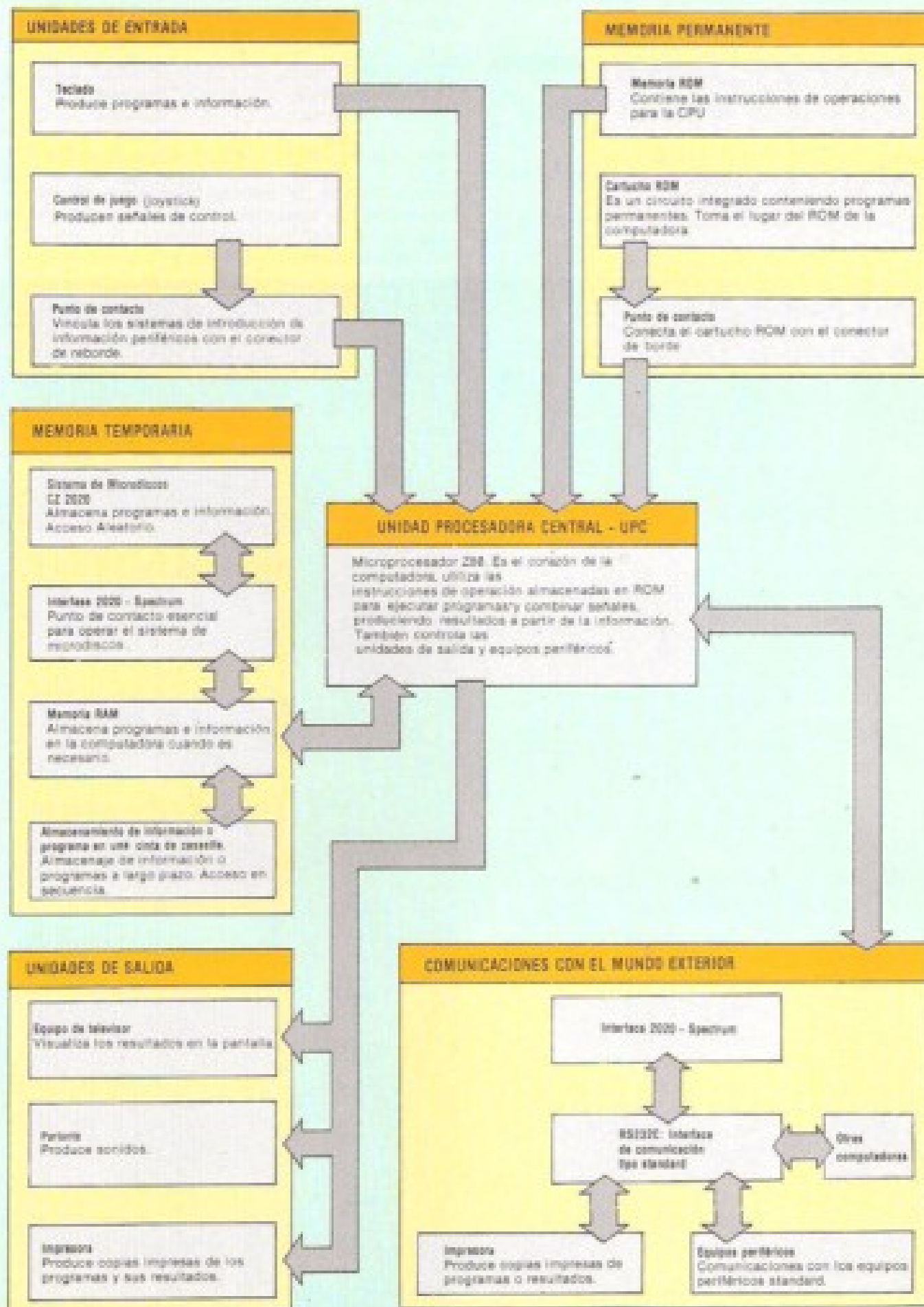
En el interior de su Spectrum, los códigos consisten en rápidas secuencias de pulsaciones de electricidad. Cuando una de estas pulsaciones llega a cualquier punto en el sistema, esta breve ráfaga de electricidad equivale a 1 en binario. Si no se produce una pulsación en un punto específico, en un momento determinado, esto equivale a 0 en binario. Por lo

tanto, en el código empleado por la computadora el valor 6 es equivalente a: no-no-no-no-no-sí-sí-no.

En el diagrama adjunto es posible apreciar como la

computadora utiliza diversos tipos de información traducida al código binario para mover información de un punto del sistema al otro.





## EL PROCEDIMIENTO PARA CONECTAR EQUIPOS ACCESORIOS:

Es posible completar su Spectrum Plus con una serie de equipos accesorios que incrementarán su rendimiento, transformándolo en un poderoso sistema de procesamiento de datos.

Existe disponible una amplia variedad de equipos periféricos CZ y de otras marcas compatibles con su Spectrum Plus.

Sin necesidad de ninguna interface adicional se pueden conectar:

- **2 Controles de Juego CZ 800 (Joysticks):** Estos se enchufan en los conectores DIN ubicados en el borde izquierdo de su CZ Spectrum Plus. En caso de emplear un solo conector, usar el conector 1 (trasero). Los controles de juego se pueden conectar y desconectar con la máquina encendida, a diferencia de los otros periféricos.
- **Impresoras de papel común de bajo costo o bien impresoras de papel térmico.**
- **Monitor de Video:** La CZ Spectrum Plus admite tanto la conexión a monitores monocromáticos como a Monitores color sistema Pal N, con entrada video-compuesta. La información sale de la computadora, desde su borde posterior, conectando el cable coaxial en el enchufe "Monitor".

Usando las interfaces correspondientes se pueden conectar:

- **Impresoras:** alimentadas por papel común o a tracción con formularios continuos.
- **Modem (modulador-demodulador de comunicaciones):** Para transmitir o recibir información a distancia y permitir el acceso a Bancos de Datos.
- **El Conjunto de Microdiscos CZ 2020,** con el que es posible utilizar discos magnéticos (Microdiscos) para el manejo ágil de programas e información. Este Conjunto también permite conectar una amplia gama de equipos periféricos, o a otra computadora Spectrum.

El conjunto de Microdiscos CZ 2020 se enchufa por su Interface Spectrum al conector de borde en la parte posterior de su Spectrum Plus. Los microdiscos pueden reemplazar a los cassettes de cinta para almacenar información.

### Programación con Controles de Juegos:

Usted puede escribir sus propios programas para usar los controles de juego en BASIC, o usando código de máquina. Los controles de juego simulan las teclas de la parte superior del teclado de la siguiente manera:

Tecla	Control 1	Tecla	Control 2
1	Izquierda	6	Izquierda
2	Derecha	7	Derecha
3	Abajo	8	Abajo
4	Arriba	9	Arriba
5	Disparo	0	Disparo

Se pueden leer desde BASIC usando INKEY\$, pero esta función leerá solo una tecla a la vez; no reconocerá izquierda y disparo simultáneos, por

ejemplo. La manera alternativa y más efectiva de leer los controles de juego es usar la función IN:

IN 83486 lee el control de juegos 1	IN 81438 lee el control de juegos 2
bit 0 - disparo	bit 4 - disparo
bit 1 - arriba	bit 3 - arriba
bit 2 - abajo	bit 2 - abajo
bit 3 - derecha	bit 1 - derecha
bit 4 - izquierda	bit 0 - izquierda

El siguiente programa BASIC muestra como usar el valor dado por la función IN.

Este programa dibuja sobre la pantalla de acuerdo al movimiento del Control de juegos. Al oprimir el pulsador de disparo se deja de dibujar, pero se sigue moviendo la posición del punto. La subrutina

en la línea 1000 lee la posición en que está el control de juegos; Ud. la puede adoptar para el uso de sus programas.

### PIZARRA MAGICA:

```

10 LET X=0 LET Y=0
20 GOTO 1000
30 IF IN(0) THEN GOTO 1000
40 IF IN(1) THEN GOTO 1000
50 IF IN(2) THEN GOTO 1000
60 IF IN(3) THEN GOTO 1000
70 IF IN(4) THEN GOTO 1000
80 IF IN(5) THEN GOTO 1000
90 IF IN(6) THEN GOTO 1000
1000 LET X=X+1
1100 LET Y=Y+1
1200 GOTO 30

```



Las impresoras standard RS232C se conectan a la Spectrum Plus mediante el Controlador CZ 2020. También puede conectarse aquí un Modem de comunicaciones.

**¡CUIDADO!** Antes de conectar cualquier accesorio debe desconectar la alimentación de energía a la computadora.

## MAPA DE LA MEMORIA DE LA SPECTRUM

La fotografía del interior de la computadora Spectrum, en las páginas 42-43, muestran entre otros componentes un circuito integrado para la memoria ROM y circuitos más pequeños para la memoria RAM. En conjunto estos circuitos constituyen la totalidad de la memoria de la Spectrum. La memoria de la Spectrum consiste en 85535 unidades de almacenamiento (storage units), cada una de las cuales contiene un byte (un número entre 0 y 255). Cada unidad es identificada por un número denominado su dirección.

En inglés ROM significa *Read Only Memory*, es decir la memoria que solamente puede ser leída por la computadora pero que no puede ser utilizada por el usuario para almacenar información ni borrarla. Esta memoria permanente contiene las instrucciones de operación de la Unidad Procesadora Central (en inglés, Central Processing Unit o CPU). La capacidad del circuito integrado ROM es de 16K. Esto significa que el circuito puede contener 16 x 1024 bytes o direcciones (en total 16384 bytes). Estos bytes solamente pueden ser leídos, no es posible modificarlos. (Si fuera posible cambiarlos la consecuencia sería que la computadora no podría continuar funcionando). Para examinar el contenido de una posición o dirección en la memoria de la computadora se utiliza la palabra clave PEEK (en inglés "peek" literalmente significa "mirar").

RAM significa *Random Access Memory*, es la memoria a la cual el usuario tiene acceso y contiene todos los programas e información ingresada en la computadora desde el teclado o por otros medios. La capacidad de la memoria RAM de la Spectrum es 48K, es decir, este sector de la memoria total contiene 48 x 1024 bytes o direcciones (en total 49152 bytes). Random Access significa en castellano "Acceso aleatorio" o "Acceso al azar", lo que indica que cualquiera de las direcciones de este sector de la memoria puede ser modificada por el programador. Para cambiar el contenido de una dirección en la memoria RAM se utilizan sentencias conteniendo la palabra clave POKE.

Las direcciones de la memoria se extienden desde 0 hasta 85535, la primera cuarta parte siendo memoria ROM y el resto memoria RAM.

### Variables de sistema

La columna a la derecha muestra la organización de la memoria de la Spectrum. El diagrama indica la posición de las diversas secciones que controlan la computadora. Es posible cambiar la posición de varias de estas secciones. Los límites de estas secciones son determinados por variables de sistema.

Las variables de sistema utilizadas por la computadora Spectrum no son variables como las que existen en el idioma BASIC. Las variables de sistema son nombres para ciertos valores fijos ubicados en direcciones o posiciones particulares de la memoria de la computadora. El propósito de estos nombres es ayudarte a recordar esas posiciones. Por ejemplo, la variable de sistema RAMTOP es la dirección en el tope o punto más alto del espacio destinado a la memoria RAM. El valor de RAMTOP establece la cantidad de espacio

### Mapa de la memoria de la Spectrum

Gráficas definidas por el usuario	
Pila GOSUB	RAMTOP
Libre	
Pila del calculador	STKEND
Espacio de trabajo temporario	STKBOT
INPUT de datos o información	WORKSP
Comando o línea del programa siendo editada	E-LINE
Variables	VARS
Programa en BASIC	PROG
Información del canal	CHANS
Mapas de la Microdrive	23734
Variables de sistema	23552
Memoria intermedia de la impresora	23296
Atributos	22528
Archivo de la presentación visual	16384
16 K de ROM	

## 4. APRENDIENDO ACERCA DEL BASIC DE SPECTRUM

---

Este capítulo describe los detalles del lenguaje BASIC suministrando un sumario sobre el empleo de cada una de las palabras clave y otros detalles adicionales acerca de aquel. La información cubre los aspectos requeridos para crear desde los programas en BASIC más sencillos hasta los más complejos y avanzados. El capítulo no ha sido organizado para ser leído desde el principio al final. Su estructura es la de un diccionario para la programación, y le permitirá sacar el máximo provecho de su Spectrum

# GUÍA DE REFERENCIA PARA EL PROGRAMADOR, UTILIZANDO LAS PALABRAS CLAVE DEL BASIC DE SPECTRUM

Esta guía contiene una descripción completa de todas las palabras clave en BASIC disponibles en la Spectrum Plus. Cada entrada incluye información sobre los siguientes aspectos:

- Posición de la palabra clave.
- Clase de palabra clave.
- Propósito de la palabra clave.
- Empleo de la palabra clave.
- Formato utilizado al redactar un programa conteniendo la palabra clave.

Los detalles referentes a la posición, propósito y empleo no requieren más explicación. La información sobre clase y formato es algo más compleja. Por lo tanto, le sugerimos que, primero estudie cuidadosamente la información de esta guía.

## Clases de Palabras Clave

Las palabras clave pueden clasificarse en una o más de cuatro clases.

### Comandos

Un comando es una palabra clave que le indica a la computadora que ejecute determinada acción al ser ingresada. Por ejemplo, RUN, LOAD.

### Sentencias

Como los comandos, las sentencias son palabras claves que le indican a la computadora que ejecute determinada acción. Las sentencias se usan en líneas de programa.

Ejemplos: DRAW, INPUT.

### Función

Es una palabra clave que produce un valor. La función forma parte de un comando o una sentencia.

Ejemplos: RND, INT.

### Operador lógico

Es una palabra clave empleada para realizar procesos lógicos, y puede ser incluida en un comando o una sentencia. Puede determinar si una determinada condición es verdadera o falsa, o cambiar tal condición. La Spectrum tiene tres operadores lógicos AND, OR y NOT.

## Números y variables

### Números

Almacenados con una precisión de 9 a 18 dígitos. La amplitud en la manipulación de números es de entre aproximadamente  $10^{-38}$  y  $4 \times 10^{-38}$ .

### Variables aceptadas

Números. Cualquier longitud, comenzando con una letra. Se ignoran los espacios y todas las letras son convertidas en minúsculas. No se distingue entre mayúsculas y minúsculas.

Cadena. Cualquier letra sola seguida de un signo de \$. No se distingue entre mayúsculas y minúsculas.

Tabla. Para variables de tabla ver la sección sobre DIM.

## Formato de las palabras clave

La sección sobre formato de las palabras claves describe la sintaxis correcta de cada palabra clave, explicando cual es la combinación correcta de palabras claves, y demás factores incluidos en una instrucción introducida en la computadora. Los elementos

frecuentemente combinados con palabras claves incluyen variables y valores. En el texto se utilizarán las siguientes abreviaciones al describir el formato correcto para una palabra clave determinada:

Abreviatura	Explicación	Ejemplo
const-num	Una constante numérica (número)	24.5
var-num	Una variable numérica (una variable que puede contener un número)	suma
expr-num	Expresión numérica (cualquier combinación de constantes y variables numéricas que producen un número como resultado)	suma*24.5 RND*7
const-num-ent	Una constante, variable o expresión numérica cuyo valor es redondeado al número entero más próximo	
var-num-ent		
expr-num-ent		
const de cadena	Una constante de cadena o cadena (cualquier combinación de caracteres entre comillas)	"Spectrum +"
var de cadena	Una variable de cadena (una variable que puede contener un cadena)	¡\$
expr de cadena	Una expresión de cadena (cualquier combinación válida de constantes de cadena y variables de cadena que suministre una cadena)	¡\$+ "Spectrum +" ¡\$ 05 TO 8)
letra	Cualquier letra mayúscula o minúscula	T x
letra\$	Cualquier letra mayúscula o minúscula seguida del signo \$	RS-¡\$
cond	Una condición, o una sub-condición insertada dentro de una condición	x = 12 AND y < 10
sentencia	Cualquier sentencia en BASIC que es válida cuando utilizada con otra sentencia.	PRINT INK 2; x
[ ]	Un elemento opcional que puede ser repetido	

NOTA: En el texto los términos valor numérico y valor de cadena denotan cualquier elemento numérico o de cadena respectivamente.

## Los signos en el BASIC de su Spectrum

Signo	Ubicación	Acción/Uso	2	Separa sentencias en líneas de programa
\$	4	Variable de cadena	/	División
:	7	Inicia nueva línea	*	Multiplicación
(	8	Abre paréntesis	.	Punto decimal
)	9	Cierra paréntesis	:	Controla visualización de información en la pantalla: Separa sentencias dentro de sentencia de programa
<=	Q	Menor o igual que	"	Abre y cierra una cadena
<>	W	No es igual a	'	Controla visualización de información en la pantalla. Ubica información en las columnas 0 o 16 de la pantalla. Separa los valores insertados a continuación de una palabra clave
>=	E	Mayor o igual que		
<	R	Menor que		
>	T	Mayor que		
↑	H	Elevar a la potencia		
-	J	Resta. Número negativo		
+	K	Suma. Número positivo.		
=	L	Concatenación de cadenas igual		

## Juego de caracteres del Spectrum

	0	1	2	3	4	5	6	7	8	9
0							PRINT	EDIT	cursor izquierda	cursor derecha
10	cursor abajo	cursor arriba	DELETE	ENTER	numero		INC control	PAPER control	FLASH control	BRIGHT control
20	INVERSE control	OVER control	AT control	TAB control						
30			espacio	!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	/	]	↑	-	£	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z				-	©	□	■
130	▣	▤	▥	▦	▧	▨	▩	▪	▫	▬
140	▭	▮	▯	▰	GRAPHICS A	GRAPHICS B	GRAPHICS C	GRAPHICS D	GRAPHICS E	GRAPHICS F
150	GRAPHICS G	GRAPHICS H	GRAPHICS I	GRAPHICS J	GRAPHICS K	GRAPHICS L	GRAPHICS M	GRAPHICS N	GRAPHICS O	GRAPHICS P
160	GRAPHICS Q	GRAPHICS R	GRAPHICS S	GRAPHICS T	GRAPHICS U	RND	INKEYS	IN	FN	POINT
170	SCREENS	ATTR	AT	TAB	VALS	CODE	VAL	LEN	SIN	COS
180	TAN	ASN	ACS	ATN	LN	EXP	INT	SQR	SGN	ABS
190	PEEK	IN	USR	STR\$	CHR\$	NOT	BIN	OR	AND	<=
200	>=	<>	LINE	THEN	TO	STEP	DEF FN	CAT	FORMAT	MOVE
210	ERASE	OPEN +	CLOSE +	MERGE	VERIFY	BEEP	CIRCLE	INK	PAPER	FLASH
220	BRIGHT	INVERSE	OVER	OUT	UPRINT	LLIST	STOP	READ	DATA	RESTORE
230	NEW	BORDER	CONTINUE	DIM	REM	FOR	GO TO	GO SUB	INPUT	LOAD
240	LIST	LET	PAUSE	NEXT	POKE	PRINT	PLOT	RUN	SAVE	RANDOMIZE
250	IF	CLS	DRAW	CLEAR	RETURN	COPY				

**ABS** ABSolute value Valor Absoluto

- **Posición en el teclado**  
MODULO EXTENDIDO  
G

**Función**

La palabra clave ABS produce la magnitud absoluta de un valor numérico. Esto es, el valor sin un signo de positivo o negativo.

- **Cómo emplear ABS**

ABS es seguido por un valor numérico. Una expresión debe estar encerrada entre paréntesis, por ejemplo:

```
50 LET x = ABS(y - z)
```

ABS retorna el valor absoluto del valor numérico.

- **Ejemplo**

El comando

```
PRINT ABS -34.2
```

produce como resultado que en la computadora se visualice el número 34.2

- **Formato**

ABS const-num

ABS var-num

ABS (expr-num)

**ACS** Arc CoSine Arco CoSeno

- **Posición en el teclado**  
MODULO EXTENDIDO  
SYMBOL SHIFT W

**Función**

ACS calcula el valor de un ángulo a partir de su coseno.

**Cómo emplear ACS**

ACS va seguido por un valor numérico. Una expresión debe estar encerrada entre paréntesis, por ejemplo:

```
60 LET x = ACS(y * z)
```

El valor a continuación de la palabra clave ACS ( $y * z$  en el ejemplo) es el coseno del ángulo requerido y puede variar entre  $+1$  y  $-1$ . ACS retorna el valor del ángulo en radianes (no en grados). Para convertir los radianes en grados se multiplica el valor retornado por ACS por  $180/\pi$ .

- **Ejemplo**

```
PRINT 180/PI * ACS 0.5
```

El resultado será 60, el ángulo en grados correspondiente a un coseno de 0.5

- **Formato**

ACS const-num

ACS var-num

ACS (expr-num)

**AND** y

- **Posición en el teclado**  
SYMBOL SHIFT Y

**Operador lógico/función**

AND opera como un operador lógico, para determinar la veracidad de una combinación de condiciones. Solamente produce el efecto deseado si todas las condiciones enumeradas en la sentencia son verdaderas. AND también actúa como una función para ejecutar operaciones binarias en dos valores de cadena o dos valores numéricos.

- **Cómo emplear AND**

Como un operador lógico, AND vincula dos condiciones contenidas en una sentencia, donde es necesario establecer la veracidad del conjunto, por ejemplo:

```
90 IF x = y + z AND time < 10  
THEN PRINT "Correcto"
```

Únicamente si ambas condiciones ( $x = y + z$ ) y ( $time < 10$ ) son verdaderas, la computadora visualizará la palabra Correcto en la pantalla. Si cualquiera de las dos condiciones es falsa, entonces la totalidad de la sentencia será considerada como falsa por la computadora que continuará a la próxima línea del programa.

- **AND como una función**

Como una función AND puede operar en relación a dos valores numéricos, por ejemplo:

```
50 LET x = y AND z
```

AND retorna el primer valor ( $y$ ) si el segundo valor ( $z$ ) no es igual a 0 y retorna 0 si el segundo valor ( $z$ ) es 0. AND también puede operar

con un valor de cadena (string value) siempre y cuando éste preceda dicha palabra clave. Un valor numérico debe siempre seguir AND, por ejemplo:

```
50 LET a$ = b$ AND z
```

AND retorna el primer valor ( $b$ ) si el segundo valor ( $z$ ) no es cero. AND retornará una cadena nula o vacía si el segundo valor ( $z$ ) es cero.

Observe que el Spectrum asigna un valor de 1 a una condición verdadera y un valor de 0 a una condición falsa. La computadora reconoce cualquier valor que no sea cero como verdadero y el valor cero como falso. El Spectrum no evalúa combinaciones de valores numéricos de acuerdo con las tablas de verdad estándar.

- **Formato**

cond. AND cond.

expr-num AND expr-num,

expr. de cadena de

caracteres AND expr-num.

**ASN** Arc Sine

Arco Seno

- **Posición en el teclado**  
MODULO EXTENDIDO  
SYMBOL SHIFT Q

**Función**

ASN calcula el valor de un ángulo a partir de su seno.

- **Cómo emplear ASN**

ASN es seguido por un valor numérico. Una expresión debe estar encerrada entre paréntesis, por ejemplo:

```
60 LET x = ASN(y * z)
```

El valor siguiendo la palabra clave ASN (en el ejemplo:  $y * z$ ) es el seno del ángulo requerido y puede variar entre  $-1$  y  $+1$ .

Obedeciendo la palabra clave ASN la computadora retornará el valor del ángulo en radianes (no en grados). Para convertir los radianes en grados se multiplica el valor retornado por ASN por  $180/\pi$ .

- **Ejemplo**

```
PRINT 180/PI * ASN 0.5
```

el resultado será 30, el



**BIN** BiNary number Número Binario

- Posición en el teclado  
MODULO EXTENDIDO  
B

BIN transforma un número binario en un número decimal.

- Cómo emplear BIN

BIN es seguido por el número binario. El número binario puede consistir de hasta dieciséis números 1 o 0, por ejemplo:

50 POKE USA "a", BIN 101010

La palabra clave producirá el número decimal correspondiente al número binario. BIN es utilizado comúnmente en conjunción con las palabras clave POKE y USA, como en el ejemplo anterior, para crear caracteres gráficos definidos por el usuario de la computadora. En este caso 1 significa un pixel de color de tinta y 0 un pixel de color de papel.

- Ejemplo  
El comando

PRINT BIN 11111110

visualizará el número decimal 254, el equivalente del valor binario indicado en la sentencia.

- Formato  
BIN [1] [0]

**BORDER** Borde

- Posición en el teclado  
B

Sentencia/comando

BORDER especifica el color del borde en torno del área de visualización en la pantalla.

- Cómo emplear BORDER

La palabra clave BORDER puede ser utilizada como un comando directo o como una sentencia contenida en un programa. BORDER es seguida por un valor numérico, por ejemplo:

30 BORDER RND\*7

El valor a continuación de la

palabra clave BORDER es redondeado al número entero más próximo. Su función es especificar el color del borde como sigue:

- 0 Negro
- 1 Azul
- 2 Rojo
- 3 Magenta (morado)
- 4 Verde
- 5 Celeste
- 6 Amarillo
- 7 Blanco

Observe que BORDER también define el color del papel en la parte inferior de la pantalla. A diferencia de lo que sucede en el caso de las palabras clave INK y PAPER, una sentencia conteniendo la palabra clave BORDER no puede ser incorporada (insertada) en una sentencia conteniendo la palabra clave PRINT

- Ejemplo

BORDER 2

Produce un borde de color rojo en torno del área de visualización en la pantalla.

- Formato  
BORDER expr num ent

**BRIGHT**

- Posición en el teclado  
MODULO EXTENDIDO  
SYMBOL SHIFT B

Sentencia/comando

El resultado de BRIGHT es incrementar el brillo de los colores de los caracteres visualizados en la pantalla por la computadora.

- Cómo emplear BRIGHT

La palabra BRIGHT puede ser utilizada como un comando directo, pero normalmente es empleada para formar una sentencia incluida en un programa. BRIGHT es seguida de un valor numérico. Por ejemplo:

80 BRIGHT 1

El valor a continuación de BRIGHT es redondeado al número entero más próximo si fuera necesario, entre 1 y 8. Un valor de 1 determina que todos los caracteres

visualizados a continuación en la pantalla, en ejecución de sentencias PRINT o INPUT, aparezcan en colores de tinta y papel más brillantes. Un valor de 8 causa que al escribirse nuevos caracteres en las posiciones brillantes aquéllos se mantengan brillantes y que las posiciones de caracteres normales se mantengan normales. La palabra clave BRIGHT seguida de 0 cancela las instrucciones BRIGHT 1 y BRIGHT 8, y como resultado, todos los caracteres visualizados a continuación serán de brillo normal.

BRIGHT también puede ser insertada dentro de sentencias afectando la visualización de información en la pantalla y conteniendo las palabras clave PRINT, INPUT, PLOT, DRAW y CIRCLE. La palabra clave BRIGHT es ubicada a continuación de una de aquellas palabras clave, pero precede los parámetros de datos o visualización. BRIGHT es seguida por los mismos valores y un punto y coma, por ejemplo:

50 PRINT BRIGHT 1:  
"ADVERTENCIA"

El efecto de BRIGHT entonces será local y se aplicará exclusivamente a los caracteres visualizados, punto dibujado (plotted), o línea trazada por la computadora en la pantalla, en ejecución de ese comando. Recuerde que BRIGHT 1 aumenta el brillo del color del papel de la totalidad de la posición del carácter, de 8x8 pixels, si cualquiera de los pixels en dicha posición es dibujado en color de tinta.

- Formato  
BRIGHT expr núm ent [:]

**CAT** CATalogue

Comando para el manejo de información almacenada. Ver Manual de Sistema de discos 2020 o de Microdrive de interface 1.

## CHR\$ CHaRacter string

Cadena de caracteres

- Posición en el teclado  
MODULO EXTENDIDO  
U

### Función

El juego de caracteres del Spectrum se compone de los caracteres y palabras clave en el teclado de la computadora más cualquier carácter gráfico definido por el usuario. Mediante el empleo de la palabra clave CHR\$ y un código numérico cada uno de los componentes del juego de caracteres puede ser obtenido como una cadena. El juego de caracteres también incluye varios códigos de control que afectan su visualización en la pantalla. Estos códigos pueden ser puestos en operación, y los caracteres visualizados, incorporando la palabra clave PRINT antes de CHR\$. El juego de caracteres completo y sus números-códigos se detallan en la página 51.

- Cómo emplear CHR\$

CHR\$ es seguido por un valor numérico, por ejemplo:

```
80 PRINT CHR$ x
```

Una expresión debe ser encerrada entre paréntesis. El valor a continuación de CHR\$ (x en el ejemplo) es redondeado al número entero más próximo. Los números enteros entre 32 y 255 producen como una cadena alguno de los caracteres en el teclado de la computadora elementos gráficos definidos por el usuario, o una palabra clave. El Spectrum emplea los valores en el código ASCII para valores entre 32 y 95, y entre 97 y 126. En el ejemplo, si se asigna un valor de 65 a la variable x, la computadora visualizará la letra A.

- Códigos de control CHR\$

Los valores entre 1 y 31 solamente se utilizan para producir códigos de control. Las sentencias CHR\$6 (PRINT coma), 8 (retroceder

un espacio) y 13 (nueva línea o ENTER) afectan las imágenes visualizadas en la pantalla si son incluidos en una sentencia con la palabra clave PRINT. CHR\$ puede ser seguido por el valor-código y un punto y coma, por ejemplo:

```
60 PRINT "A"; CHR$6; "B"
```

Esta sentencia le indica a la computadora que visualice lo siguiente:

```
A      B
```

Los códigos de control CHR\$ también pueden ser utilizados para formar una cadena de caracteres compuesta, que los contiene. La sentencia

```
60 PRINT "A" + CHR$6 + "B"
```

tiene exactamente el mismo efecto que la anterior.

Los códigos 16 a 23 afectan el color y la posición, y pueden ser incluidos en cadenas de caracteres compuestas conjuntamente con CHR\$. En este último caso, CHR\$ 16 (control de la tinta o INK) y CHR\$ 17 (control del papel o PAPER) serán seguidos por un valor código para los colores entre 0 y 7; y en el caso de las palabras claves para los efectos especiales (FLASH, BRIGHT, INVERSE y OVER) los CHR\$ 18 a CHR\$ 21 serán seguidos por el número 1 (pone en operación el efecto especial) o 0 (lo termina). Por ejemplo el comando

```
PRINT CHR$16 + CHR$3  
+ CHR$17 + CHR$6  
+ CHR$18 + CHR$1 +  
"Spectrum"
```

tendrá el resultado de visualizar las palabras SPECTRUM en letras a destellos, rojas y amarillas. Como en los ejemplos precedentes, es posible sustituir los símbolos de .+ por puntos y comas.

CHR\$22 (control AT) es seguido por dos valores CHR\$ para indicar los números de la línea y columna. El comando

```
PRINT CHR$22 + CHR$11  
+ CHR$16 + CHR$42
```

le indica a la computadora que visualice una estrella en el centro de la pantalla.

CHR\$23 (control TAB) también es seguido por dos valores de la misma manera. El segundo valor normalmente es 0 y el primero indica la posición TAB

```
PRINT CHR$23 + CHR$16 +  
CHR$0 + CHR$42
```

produce la estrella en otra posición.

Tenga en cuenta que estos son los únicos controles disponibles. Si se utiliza PRINT CHR\$ con un valor de palabra clave mayor que 164 solamente se visualizará el nombre de la palabra clave sin ponerla en ejecución.

- Formato

```
CHR$ const num ent [ ] +  
CHR$ var num ent [ ] +  
CHR$ ( expr num ent) [ ] +
```

## CIRCLE Círculo

- Posición en el teclado  
MODULO EXTENDIDO  
SYMBOL SHIFT H

### Sentencia/comando

CIRCLE dibuja un círculo en la pantalla.

- Cómo emplear CIRCLE

La palabra clave CIRCLE es seguida de tres valores numéricos, cada uno de ellos separado por una coma. Ejemplo:

```
80 CIRCLE x,y,z
```

Si fuera necesario cada uno de los tres valores será redondeado al número entero más cercano. La computadora traza un círculo en la grilla de gráficos de alta resolución, utilizando el color de tinta vigente en ese momento. Los primeros dos valores (x,y) definen coordenadas horizontal y vertical del centro del círculo. El tercer valor (z) define el largo del radio. Las dimensiones deben ser tales que el círculo no se extienda fuera del área de visualización de la pantalla.

CIRCLE es afectado por sentencias o comandos controlando el color de las figuras en la pantalla, y puede incluir instrucciones referentes al color, con los mismos efectos que en las sentencias conteniendo las palabras clave PLOT y DRAW.

#### CIRCLE 128,88,87

dibujará un círculo ocupando la mayor parte del área de visualización.

##### • Formato

CIRCLE [sentencia:] número entero, número entero, número entero

#### CLEAR Borrar

##### • Posición en el teclado X

##### Sentencia/comando

CLEAR borra los valores corrientes de todas las variables, liberando el espacio de la memoria ocupado por aquellos valores y el espacio hasta el límite RAMTOP, la dirección superior del área regida por el sistema BASIC. La palabra clave CLEAR también puede ser utilizada para reajustar el límite RAMTOP (literalmente RAMTOP significa "el tope del área RAM").

##### • Cómo emplear CLEAR

CLEAR puede ser utilizado como un comando directo o puede ser incluido como parte de una sentencia incorporada en un programa. No requiere parámetros. Por ejemplo:

#### 50 CLEAR

Obedeciendo la instrucción CLEAR la computadora borra los valores asignados corrientemente a todas las variables, incluyendo las tablas (arrays). También ejecuta las palabras clave CLS y RESTORE para limpiar la pantalla y restaurar el indicador de información o datos (data pointer) al primer elemento de datos o información. Además, la posición de PLOT (dibujar) es reajustada, poniéndola en

el punto inferior a la izquierda del área de visualización. La pila de GOSUB también es borrada.

Observe que no se requiere incluir CLEAR antes de proceder a dimensionar las tablas (arrays), ya que la palabra clave DIM utilizada en ese caso borra cualquier otra tabla preexistente de su mismo nombre. Tenga presente, además, que RUN ejecuta la sentencia o comando CLEAR.

##### • CLEAR Y RAMTOP

CLEAR puede ser seguida por un valor numérico, por ejemplo:

#### CLEAR 65267

A continuación, CLEAR opera como se explicó más arriba, y acto seguido ajusta el RAMTOP —la dirección más alta del área de BASIC de la computadora— al valor indicado. En la Spectrum el valor de RAMTOP es ajustado a 65367, y se encuentra ubicado debajo del área reservada para los gráficos definidos por el usuario. La palabra clave NEW borra el contenido de la memoria hasta el límite establecido en el RAMTOP. Como resultado, es posible recurrir al empleo de CLEAR para definir un sector de la memoria que es inmune a la palabra clave NEW. En el ejemplo, esta área se encuentra entre el límite RAMTOP de la computadora Spectrum (65367) y el límite establecido mediante la palabra clave CLEAR (65267), y su capacidad es de 100 bytes. El elevar el límite RAMTOP incrementa la memoria disponible para BASIC, sacrificando parte del espacio reservado para los gráficos definidos por el usuario. Observe que la pila GOSUB será situada en el RAMTOP, siempre y cuando el RAMTOP permanezca por encima de la pila de la calculadora.

Para determinar cuál es el RAMTOP, ingresar en la computadora las

instrucciones siguientes:

```
PRINT PEEK 23730 +
256*PEEK23731
```

##### Formato

CLEAR [exp-num]

#### CLOSE #

Comando para el manejo de información almacenada. Ver Manual de Sistema de discos 2020 o de Microdrive de interface 1.

#### CLS Clear Screen Limpiar la pantalla

##### • Posición en el teclado Y

##### Sentencia/comando

CLS borra todos los textos y gráficos en el área de visualización de la pantalla, dejándola en blanco y con el color de papel (fondo) actual.

##### • Cómo emplear CLS

CLS puede ser utilizado como un comando directo o como un elemento de una sentencia contenida en un programa. No requiere parámetros. Por ejemplo:

#### 250 IF a\$ = "NO" THEN CLS

Si se cumple la condición estipulada, la computadora borrará el área de visualización (pero no el borde de la pantalla), adoptando el color de papel seleccionado por el comando o sentencia PAPER inmediatamente anterior. Si el programa no incluye tal sentencia o comando, la computadora adoptará el color de papel (PAPER) blanco.

Note que CLS debe ser utilizada después de la palabra clave PAPER y antes de PRINT, o cualquier otra sentencia afectando la visualización, para producir un fondo de colores abarcando la totalidad del área de visualización.

##### • Formato

CLS

#### CODE Código

## • Posición en el teclado MODULO EXTENDIDO

### Función

La función CODE suministra el número código de un carácter determinado en el juego de caracteres del Spectrum (ver página 51).

### • Cómo emplear CODE

CODE es seguido por una cadena de caracteres (string). Por ejemplo:

```
90 IF CODE a$ < 65 OR CODE a$ > 90 THEN GOTO 80
```

Las expresiones parte de la cadena deben ser encerradas entre paréntesis. CODE suministra el número de código del primer carácter en la cadena de caracteres. Si la cadena es una "cadena nula" o "vacía" entonces CODE suministrará el valor 0.

CODE suministra los valores de código ASCII desde 32 a 95, y desde 97 a 126.

### Ejemplo

```
PRINT CODE "Spectrum"
```

visualizará en la pantalla el número código 83, correspondiente a la letra S.

### • SAVE CODE y LOAD CODE

CODE es utilizado en forma diferente cuando se la combina con SAVE y LOAD. Ver: SAVE CODE y LOAD CODE.

### • Formato

CODE const de cadena  
CODE var de cadena  
CODE (expr de cadena)

## CONTINUE Continuar

### • Posición en el teclado C

### Comando

Si el programa se detiene, la palabra clave CONTINUE puede ser utilizada para volverlo a iniciar a partir del punto donde se detuvo. Si la detención en la ejecución del programa se debe a un error, éste deberá ser rectificado antes de que CONTINUE permita que el

programa se reinicie.

### • Cómo emplear CONTINUE

Esta palabra clave es utilizada como un comando directo cuando un programa se ha detenido. No requiere parámetros. Normalmente, después de CONTINUE el programa se reinicia en la sentencia donde se produjo la interrupción. Si la causa de ésta fue un error, es posible ingresar un comando para rectificarlo y CONTINUE permitirá que el programa continúe a partir de esa sentencia. Cuando el programa se ha detenido en una sentencia conteniendo la palabra clave STOP, causando un informe 9, o si se detuvo debido a que la tecla BREAK fue oprimida, causando un informe L, entonces CONTINUE reinicia el programa a partir de la sentencia siguiente. Antes de ingresar CONTINUE es posible ingresar un comando para rectificar una situación si fuera necesario.

Cuando se emplea CONTINUE para reasumir un comando directo, esta palabra clave entrará en un ciclo (loop) si el comando se detuvo en la primera sentencia del comando. En este caso la imagen desaparece, pero es posible recuperar control oprimiendo la tecla BREAK.

CONTINUE causará un informe 0 si el comando se detuvo en la segunda sentencia, y un informe N si se detuvo en la tercera sentencia o alguna de las subsecuentes.

### Formato

CONTINUE

## COPY

### • Posición en el teclado Z

### Comando

COPY causa que la impresora produzca una copia de la información visualizada en la pantalla de la televisión.

### • Cómo emplear COPY

La palabra clave COPY es utilizada como un comando directo cuando un programa ha sido ejecutado completamente o detenido. No requiere parámetros. Si la impresora se encuentra conectada y funcionando, después del comando COPY producirá una copia de las primeras 22 líneas de la imagen en la pantalla. Los colores de tinta (primer plano) serán impresos en negro; los colores de papel (fondo) no son impresos. Es posible detener la impresora oprimiendo la tecla BREAK.

Cuando se ha visualizado el listado de un programa en la pantalla, es posible copiarlo mediante el comando COPY, si el listado fue producido mediante el comando o una sentencia LIST. Recuerde que un listado también aparecerá en la pantalla al presionarse la tecla ENTER, después que el programa ha sido completado o detenido. Sin embargo, en este último caso, de "listado automático", el listado no podrá ser copiado mediante el comando COPY.

### • Formato COPY

## COS Códice

Coseno

### • Posición en el teclado MODULO EXTENDIDO W

### Función

COS suministra el coseno de un ángulo.

### • Cómo emplear COS

COS es seguido por un valor numérico, por ejemplo

```
140 LET x = COS y
```

Una expresión debe ser encerrada entre paréntesis. El valor a continuación de COS es el ángulo en radianes. COS suministrará el coseno del ángulo. Es posible convertir grados a radianes multiplicándolos por  $\pi/180$ .

Observe que COS suministra un valor negativo para ángulos de entre 90 a 270 grados, y un valor positivo para ángulos entre 0 y 270-360 grados.

### • Ejemplo

El comando

```
PRINT COS (60* PI/180)
```

produce como resultado el valor 0.5, el coseno correspondiente a un ángulo de 60 grados.

### • Formato

**COS** const-num

**COS** var-num

**COS** (expr-num)

## DATA Datos

### • Posición en el teclado

MODULO EXTENDIDO

D

Sentencia

DATA suministra una lista de elementos de información o datos dentro de un programa. Por ejemplo, estos elementos de información pueden ser los valores adjudicados a variables o cadenas de caracteres a ser visualizadas por la computadora. Cada elemento de información es asignado a una variable mediante una sentencia de READ.

La asignación de datos o información a las variables en el programa se efectúa siguiendo el orden en el cual dichos datos aparecen en el programa. Sin embargo, es posible utilizar la palabra clave **RESTORE** para comenzar la asignación de datos a variables en el primer elemento de información contenido en una sentencia de DATA.

### • Como emplear DATA

DATA solamente puede ser empleada para formar una sentencia parte de un programa. Normalmente esta palabra clave es seguida por una lista de constantes numéricas o de cadena *separadas entre sí por comas*. Por ejemplo

```
50 DATA 31, "ENERO", 28, "FEBRERO"
```

Cada una de las constantes es asignada a una variable determinada mediante una sentencia de READ, que le indica a la computadora que

lea un determinado elemento de información o dato. La sentencia de DATA puede ser ubicada en cualquier parte del programa. El número, tipo (numérica o de cadena) y orden de las constantes debe corresponder a la cantidad de veces que la computadora debe ejecutar la sentencia de READ. La naturaleza y orden de las variables en la sentencia de READ también debe estar de acuerdo con los elementos en la sentencia de DATA. Si existen demasiados elementos de información para ser ubicados en la misma sentencia la lista de DATA puede ser separada en varias sentencias de DATA sucesivas.

### • Ejemplo

el programa siguiente

```
10 FOR n = 1 TO 2
20 READ x,a$
30 PRINT a$,x; "días"
40 NEXT n
50 DATA 31, "ENERO", 28, "FEBRERO"
```

visualizará lo siguiente en la pantalla:

```
ENERO          31 días
FEBRERO        28 días
```

• **Utilizando DATA con variables**  
Cada uno de los elementos de información o datos contenidos en una sentencia de DATA puede consistir de variables numéricas o de cadena, o de expresiones, siempre y cuando previamente se les haya asignado valores. En el ejemplo anterior, la sentencia de DATA puede ser modificada como sigue.

```
50 DATA d,m$,d-3 "FEBRERO"
```

Si previamente se atribuyó un valor de 31 a la variable d, y un valor "ENERO" a la variable m\$, entonces al ejecutarse se obtendrá precisamente el mismo resultado que antes.

### • LOAD DATA, SAVE DATA Y VERIFY DATA

La palabra clave DATA también puede ser utilizada con **LOAD**, **SAVE**, y **VERIFY**

para almacenar tablas (arrays) en cinta de cassette. Ver: **LOAD DATA**, **SAVE DATA** y **VERIFY DATA**.

### • Formato

**DATA** expr-num [,expr-num]  
(expr de cadena)

**DATA** expr de cadena [,expr-num] [,expr-num]

## DEF FN Define Function

Define Function

### • Posición en el teclado

MODULO EXTENDIDO

SYMBOL SHIFT 1

Sentencia

DEF FN permite al usuario definir una función que no se encuentra disponible como palabra clave. Es posible pasar una variedad de parámetros a la función en una sentencia de FN. La sentencia de FN llama aquella función y puede producir como resultado un valor numérico de cadena.

### • Como emplear DEF FN

DEF FN solamente puede ser utilizada como una sentencia contenida en un programa. Cuando se desea definir una función numérica, DEF FN es seguida por cualquier letra individual y entonces por una o más variables numéricas, *separadas entre sí por una coma y encerradas entre paréntesis*, por ejemplo: **DEF FN r(x,y)**. Esto es seguido por signos de igual y una expresión numérica conteniendo las variables, por ejemplo

```
1000 DEF FN r(x,y) =
      SQR (x ^ 2 + y ^ 2)
```

La letra a continuación de DEF FN (r en el ejemplo) es un nombre que identifica la función. Las variables también pueden consistir en letras individuales. Observe que en ambos casos, el Spectrum no distingue entre letras mayúsculas y minúsculas.

La expresión que sigue al signo de igual utiliza las variables (x,y en el ejemplo)

para definir la función.

Una sentencia de DEF FN puede ser ubicada en cualquier parte del programa.

Para invocar la función definida por DEF FN, se utiliza la palabra clave FN seguida por la letra que designa la función y una lista de valores numéricos, *cada uno de ellos separado por una coma y encerrado entre paréntesis*, por ejemplo:

```
50 PRINT FN r(3,4)
```

Los valores numéricos entre paréntesis son transferidos a la función en el mismo orden en que se encuentran las variables en la sentencia DEF FN.

Así, en el ejemplo anterior, x recibirá un valor de 3, mientras que y recibirá un valor de 4.

FN evalúa la expresión y produce como resultado un valor.

DEF FN también puede ser seguida por una letra y un par de paréntesis solamente, por ejemplo

```
1000 DEF FNr() = INT (x + 0.5)
```

El valor asignado en este momento a la variable (x en el ejemplo) es pasado a la función al ser llamado por la palabra clave FN.

En este caso, FNr() produce el valor asignado a x en este momento redondeado al número entero más próximo.

#### • DEF FN y cadenas de caracteres (string)

DEF FN también puede ser utilizado de la misma forma, con el propósito de definir e invocar una función de cadena (string function). En este caso, el nombre de la función es una sola letra seguida del signo \$; además una o más de las variables en la sentencia es una letra seguida por el signo \$.

La definición está formada por una expresión de cadena

```
1000 DEF FN a$(b$,x,y) =  
b$(x TO y)
```

En este ejemplo, la expresión de cadena a continuación del signo igual utiliza un rebanador de cadena y las variables x e y son el primer y último carácter de una sección de b\$.

FN debe ser seguida por el nombre de la función y un valor de cadena entre paréntesis, conjuntamente con cualquier otro parámetro a ser pasado a la función.

En este caso, el comando

```
PRINT FN a$(  
("FUNDAMENTAL", 1,3)
```

Produce como resultado que la computadora visualiza en la pantalla la palabra FUN. De la misma manera, el comando

```
PRINT FN a$(  
("FUNDAMENTAL", 5,8)
```

producirá en la pantalla:  
AMEN.

#### • Formato

```
DEF FN letra ([letra] [,letra]) =  
expr-num
```

```
DEF FN letra$ ([letra$] [,letra$] = expr de  
cadena)
```

```
FN letra ([expr-num] [,expr-  
num])
```

```
FN letra$ ([expr de cadena]  
[expr-num] [,expr-num] [,expr  
de cadena])
```

DIM	Dimensión	Dimensión
-----	-----------	-----------

#### • Posición en el teclado

D

#### Sentencia

La palabra clave DIM es utilizada para establecer las dimensiones (componer) una tabla (array) conteniendo una cantidad predeterminada de variables numéricas o de cadena de caracteres.

Una tabla contiene un juego de variables, todas con el mismo nombre, que se distinguen las unas de las otras por sus subíndices (en inglés: subscripts). Los subíndices son valores que identifican cada una de las variables o elementos en la tabla

#### • Como emplear DIM con tablas numéricas

DIM es utilizado para formar una sentencia en un programa, siendo seguida por una letra individual que designa la tabla, y uno o más valores numéricos. Cada valor numérico es *separado por una coma y encerrado entre paréntesis*. Por ejemplo

```
20 DIM x(10)  
30 DIM z(20,5)
```

La primera es una tabla unidimensional, conteniendo diez elementos con subíndices de 1 a 10. El nombre de la tabla es x, y sus variables son identificadas combinando el nombre de la tabla con sus respectivos subíndices, por ejemplo: x(1), x(3), x(10), etc. cuando la computadora llega a esta sentencia en el programa borra cualquier tabla con el mismo nombre existente en su memoria, y asigna un valor de 0 a cada una de las variables en la tabla.

Tenga presente que al establecer las dimensiones de una tabla el Spectrum no distingue entre letras mayúsculas - para la computadora la variable x (2) es idéntica a la variable X (2). Sin embargo las variables numéricas simples, teniendo la misma letra que el nombre de una tabla (en el ejemplo x o X) pueden coexistir con esta última y ser utilizadas separadamente si fuera necesario.

La cantidad de valores entre paréntesis es igual al número de dimensiones creadas en una tabla numérica.

El segundo ejemplo muestra una tabla bidimensional con un total de 100 elementos: 20 elementos en la primera dimensión y 5 en la segunda (el total es 20 x 5). En este caso los elementos se numeran z(1,1) hasta z(20,5)

Es posible crear tablas con cualquier número de dimensiones.

Los elementos de una

tabla numérica pueden ser identificados subsecuentemente por el nombre de la tabla seguida por un valor entre paréntesis, por ejemplo

```
70 PRINT x(a)
100 PRINT z(7,b)
```

#### • DIM y tabla de cadena de caracteres

La palabra clave DIM es utilizada de la misma forma en las tablas numéricas, con la diferencia de que se utiliza una letra individual seguida por el signo \$ para nombrar la tabla.

Además, debe agregarse un valor extra a los valores entre paréntesis para definir el largo de cada cadena de caracteres. Por ejemplo:

```
30 DIM a$(20,5)
90 DIM b$(20,5,10)
```

La primera sentencia crea una tabla de 20 elementos, cada uno de los cuales contiene una cadena de cinco caracteres, las variables identificadas por subíndices se designan desde a\$(1) hasta a\$(20) inclusive. Al principio, la computadora, asigna a cada variable una cadena de caracteres "vacía" o "nula" ("").

Como antes, la computadora borrará de su memoria cualquier otra tabla del mismo nombre que pueda encontrarse en ella. Además, a diferencia de lo que sucede en el caso de las tablas numéricas, una variable de cadena simple no puede coexistir con una tabla de cadena del mismo nombre.

El segundo ejemplo crea una tabla de cadena bi-dimensional, con un total de 100 elementos: 20 elementos en la primera dimensión y 5 en la segunda. Todos los elementos tienen una extensión de 10 caracteres.

Subsecuentemente, cuando se asignan valores a una tabla de cadena de caracteres, estos son complementados por valores

de relleno al final de la cadena, o son cortados, si fuera necesario para ajustarlo a la extensión especificada.

Los elementos de una tabla de cadena son identificados por el nombre de la tabla, seguido por uno o más valores numéricos entre paréntesis, proporcionando el número o números del subíndice correspondiente.

Por ejemplo, el elemento a\$(2) de la tabla puede ser "PEREZ", y el elemento b\$(12,4) puede ser "MADRID".

Sin embargo, es posible agregar un valor extra para definir un carácter particular en la cadena. En estos ejemplos a\$(12,4,5) sería 1 (el quinto carácter en MADRID).

#### • Tablas de cadena de caracteres de dimensión cero

Es posible crear una tabla de cadena con una dimensión de cero, utilizando solamente un valor entre paréntesis, por ejemplo

```
10 DIM c$(15)
```

Esta tabla contiene solamente un elemento, c\$, cuya extensión es establecida en el valor definido de 15 caracteres (15).

#### • Formatos

DIM letra (expr-num [,expr-num])

DIM letra\$ (expr-num [,expr-num])

### DRAW Trazar

#### • Posición en el teclado W

#### Sentencia/comando

La palabra clave DRAW es utilizada para trazar líneas rectas o curvas en la pantalla.

#### • Como emplear DRAW

Normalmente, DRAW es utilizada para formar una sentencia en un programa, cuando se requiere una línea recta, seguida de dos valores numéricos separados

por una coma. Por ejemplo

```
40 DRAW x,y
```

La computadora trazará una línea recta en la grilla de gráficos de alta resolución. La línea será trazada a partir de la posición definida por la sentencia de PLOT anterior o la posición donde terminó la línea trazada en ejecución de una sentencia DRAW anterior, cualquiera sea la última.

Los dos valores a continuación de la palabra clave DRAW serán redondeados al número entero más próximo si fuera necesario.

El primer valor (x en el ejemplo) define la distancia horizontal desde esta posición, y el segundo valor (y) define la distancia vertical. Estos valores son negativos cuando la línea se trazará hacia la izquierda o hacia abajo en la pantalla. La posición donde la línea termina debe encontrarse dentro del área de visualización de la pantalla.

Cuando el programa no contiene sentencias de PLOT o DRAW anteriores, la nueva sentencia DRAW tendrá el efecto de comenzar a trazar la línea a partir de la esquina inferior izquierda de la pantalla, identificada por las coordenadas 0,0.

DRAW será afectada por sentencias conteniendo instrucciones o comandos relativos al color. También puede incluir sentencias incorporadas, con los mismos resultados que en las sentencias PLOT y CIRCLE.

• DRAW, trazando curvas  
DRAW puede ser seguido por un tercer valor para producir una curva, sector de una circunferencia, por ejemplo

```
40 DRAW x,y
```

El tercer valor (z) define el ángulo en radianes de la línea trazada en la pantalla. Cuando el valor es positivo

la línea será trazada hacia la izquierda, y hacia la derecha cuando es negativo. Los valores PI o -PI producen un círculo.

#### • Ejemplo

El programa siguiente traza un triángulo en la pantalla

```
10 PLOT 127, 150
20 DRAW 70, -100
30 DRAW -140,0
40 DRAW 70,100
```

Cuando se agrega el valor 1 o -1 a la sentencia conteniendo DRAW, los lados se transformaran en curvas hacia adentro o hacia afuera.

#### • Formato

**DRAW** [sentencia:] expr  
num ent  
expr num ent [,expr num ent]

### ERASE Borrar

Comando para la operación de almacenamiento de información

Ver Manual de Sistema de discos de 3" 2020 o de Microdrive de interfase 1.

### EXP EXPonent Exponente

#### • Posición en el teclado

X

#### Función

EXP es una función aritmética que eleva la Base *e* a una potencia determinada.

#### • Cómo emplear EXP

La palabra clave EXP es seguida de un valor numérico, por ejemplo

60 LET y = EXPx

Una expresión debe estar encerrada entre paréntesis. Como resultado EXP suministra la Base *e* elevado a la potencia del argumento (x en nuestro ejemplo).

#### • Ejemplo

### PRINT EXP1

produce el valor de e, que es:  
2.7182818

#### • Formato

EXP const-num  
EXP var-num  
EXP (expr-num)

### FLASH Titilación

• Posición en el teclado  
MODO EXTENDIDO  
SYMBOL SHIFT  
V

#### Sentencia/comando

La palabra clave FLASH le indica a la computadora que determinadas posiciones de caracteres "titilen" o produzcan destellos. Este efecto óptico es obtenido alternando los colores de la tinta y el papel a una velocidad preestablecida.

#### • Cómo emplear FLASH

FLASH puede ser utilizado como un comando directo, pero, por lo general, forma parte de una sentencia en un programa. Es seguido por un valor numérico, por ejemplo

50 FLASH 1

El valor numérico a continuación de FLASH es redondeado al número entero más próximo si es necesario, y puede ser 0, 1 o 8.

El valor 1 causa que "titilen" todos los caracteres visualizados subsecuentemente en ejecución de sentencias conteniendo PRINT o INPUT.

FLASH es ubicado a continuación de la palabra clave, pero precede los parámetros de la información (datos) o de la imagen va a ser visualizada. FLASH es seguido por los mismos valores y un punto y coma.

Por ejemplo

120 PRINT FLASH 1; INK 2;  
PAPER 6; "ATENCION"

en este caso el efecto de FLASH será local y se aplicará únicamente a los caracteres visualizados, punto o línea trazada en ejecución de esa sentencia.

Recuerde que FLASH determina el estado de la totalidad de la posición de 8x8 pixels, aunque solamente se dibuje (PLOT) en un color de tinta la posición de uno de ellos.

#### • Formato

FLASH expr número entero [:]

### FN FUNción Función

• Posición en el teclado  
MODO EXTENDIDO  
SYMBOL SHIFT 2

#### Función

FN invoca una función definida por el usuario. Esta palabra clave siempre es utilizada en conjunción con DEF FN, la cual define la función a ser invocada.

#### • Cómo emplear FN

Si se trata de una función numérica, FN es seguida por una letra y entonces por un par de paréntesis. Cuando se desea pasar parámetros a la función, estos son separados por comas y encerrados entre paréntesis. Por ejemplo

170 LETx = FN r(3,4)

Los parámetros (3 y 4 en el ejemplo) son pasados a la función, denominada r en este caso. A continuación FN produce el resultado.

El par de paréntesis debe ser incluido aún cuando no se requiera pasar parámetros a la función, por ejemplo

70 PRINT FN r( )

En este caso la función empleará los valores asignados corrientemente a sus variables.

FN invoca una función de cadena de caracteres de acuerdo al mismo procedimiento, excepto que

en este caso será necesario agregar el símbolo \$ inmediatamente después de la letra que designa la función. Por más detalles consulte la sección sobre la palabra clave DEF FN.

- **Formato**

FN letra ([expr-num] [,expr-num])

FN letra\$ ([expr de cadena] [expr-num] [,expr-num] [,expr de cadena])

## FOR Para

- **Posición en el teclado**

F

### Sentencia/comando

FOR es utilizado siempre con las otras dos palabras clave TO y NEXT para construir un ciclo (loop) FOR NEXT. Esta estructura permite que una determinada sección del programa se repita una determinada cantidad de veces.

- **Como emplear FOR**

FOR siempre forma una sentencia con TO. La palabra clave FOR es seguida por una letra, un signo de igual, y dos variables numéricas separadas por la palabra clave TO. Por ejemplo

```
60 FOR a = 1 TO 9
```

La letra (a en el ejemplo) es una variable de control. A continuación se incluyen las sentencias que deberán ser repetidas; normalmente una o más de estas sentencias hace uso de la variable de control. El ciclo termina con una sentencia de NEXT. En esta última sentencia, NEXT es seguida por el nombre de la variable de control (a en el ejemplo). Por ejemplo

```
90 NEXT a
```

Al ejecutarse el programa, FOR cumple una serie de operaciones. Primero: FOR borra cualquier variable del mismo nombre que la variable de control. Segundo, FOR le asigna a la variable

de control un valor inicial idéntico al valor especificado antes de TO (1 en el ejemplo). A continuación las instrucciones en las sentencias son ejecutadas.

Tercero: cuando la computadora llega a la palabra clave NEXT incrementará el valor de la variable de control en una unidad (a 2 en el ejemplo). Cuarto: la computadora comparará el nuevo valor de la variable de control con el valor indicado después de TO (9). Este último valor establece el límite para el ciclo. Quinto: si este valor es inferior al del límite la computadora reiniciará el proceso, retornando a la sentencia FOR y repitiendo el ciclo FOR NEXT. Al finalizar cada ciclo el valor de la variable de control se incrementa en una unidad. Cuando la computadora comprueba que el valor de la variable de control es superior al valor límite (9) terminará el ciclo. A continuación el programa continuará con la sentencia después de NEXT.

En el ejemplo, el ciclo es repetido nueve veces, y el valor de la variable de control se incrementará de 1 a 9. Al terminarse el último ciclo la variable de control tendrá el valor de 10.

Recuerde que el Spectrum no distingue entre letras mayúsculas y minúsculas cuando estas se utilizan para designar una variable de control.

- **Utilizando STEP en un ciclo FOR NEXT**

La palabra clave STEP puede ser incorporada en una sentencia para incrementar o disminuir el valor de la variable de control en pasos de más de 1. STEP es ubicado después del valor límite y es seguido por un valor numérico. Por ejemplo

```
60 FOR a = 1 TO 9 STEP 2
```

En este caso el valor de la variable de control será incrementado al cumplirse un ciclo en la cantidad

indicada por STEP (2 en el ejemplo). El ciclo se repetirá hasta que el valor de la variable de control sea superior al valor límite establecido.

En el ejemplo los valores sucesivos de la variable de control serán 1,3,5,7 y 9. La computadora abandonará el ciclo cuando la variable de control alcance el valor de 11 (9 + 2).

Es posible estipular un valor negativo para STEP, lo que determinará que el valor de la variable de control disminuya, en lugar de incrementarse. Lógicamente, en este caso el valor inicial debe ser mayor que el valor límite. El ciclo será terminado cuando el valor de la variable de control sea menor que el valor límite. Por ejemplo

```
60 FOR a = 9 TO 1 STEP -1
```

El valor límite disminuye desde 9 a 1, y la computadora abandona el ciclo cuando el valor de la variable de control llega a 0.

- **Acomodando un ciclo dentro de otro**

en inglés esta operación es descrita muy gráficamente como "anidar ciclos". Es posible anidar uno o más ciclos FOR NEXT, uno dentro del otro. En este caso el orden de las variables de control en las sentencias NEXT deberá ser el inverso del orden de las variables de control en las sentencias FOR. Es posible anidar, o acomodar uno dentro de otro, en cantidad ilimitada.

- **Formato**

FOR letra = expr-num TO expr-num  
[STEP expr-num]  
NEXT letra

## FORMAT Formato

Comando utilizado para controlar el procedimiento de almacenaje de información

Ver Manual de Sistema

de discos de 3" 2020  
o de Microdrive de interface  
1.

---

## **GOSUB** Diríjase a la subrutina

---

- **Posición en el teclado**  
**H**

### **Sentencia/comando**

---

La palabra clave **GOSUB** causa que la computadora se desvíe a una subrutina. Las subrutinas son secciones separadas del programa, normalmente compuestas de varias líneas. Este procedimiento es muy útil cuando la misma subrutina es utilizada varias veces durante el mismo programa.

- **Cómo emplear GOSUB**  
**GOSUB** puede ser utilizado como una sentencia o un comando directo. **GOSUB** es seguida por un valor numérico. Por ejemplo

### **GOSUB 10**

---

Cuando la computadora ejecuta el programa, el valor a continuación de **GOSUB** es redondeado al número entero más próximo, y la computadora toma el desvío indicado, dirigiéndose a la línea del programa con ese número. El empleo de una variable o expresión permite que la computadora tome el desvío a la subrutina, encaminándose a una línea calculada. Tenga presente que la computadora se desviará al llegar a la sentencia **GOSUB**, aún si el número de línea mencionado no se encuentra en el programa. En este caso la computadora continuará con la primera sentencia que encuentre en esa posición.

La subrutina es terminada con la palabra clave **RETURN** (retorno). Al encontrarla la computadora retornará a la sentencia inmediatamente después de la sentencia conteniendo la palabra clave **GOSUB** que inició el proceso.

Es posible "anidar" subrutinas, unas dentro de otras. En este caso la computadora es enviada de una subrutina a la otra. En

todos los casos, la palabra clave **RETURN**, que termina la subrutina, siempre le indica a la computadora que debe retornar a la sentencia inmediatamente después de la sentencia conteniendo **GOSUB** ejecutada en último lugar.

- **La pila de GOSUB**  
Siempre que se ejecuta una instrucción **GOSUB**, su número de línea es almacenado en la pila **GOSUB** de la memoria de la computadora. Cuando se ejecutan dos o más sentencias **GOSUB** antes de la palabra clave **RETURN**, la computadora almacenará el número de cada una de las líneas del programa conteniendo la palabra clave **GOSUB** ejecutada. Estos números de línea serán almacenados de forma tal que el último número se encuentre en el tope de la pila. El efecto de la palabra clave **RETURN** es indicarle a la computadora que retorne al número línea **GOSUB** que se encuentra en el tope de la pila. Como resultado, la computadora retornará a la línea inmediatamente después de la última sentencia conteniendo la palabra clave **GOSUB**.

Tenga presente que cuando no existen suficientes palabras clave **RETURN** puede producirse el error 4 (Out of memory, o "sin memoria").

- **Formato**  
**GOSUB** expr num ent

---

## **GOTO** Diríjase a la línea

---

- **Posición en el teclado**  
**G**

### **Sentencia/comando**

---

La palabra **GOTO** le indica a la computadora que se desvíe y se dirija a una línea determinada del programa (a diferencia de **GOSUB** que le indica que avance a una subrutina).

- **Cómo emplear GOTO**  
La palabra clave **GOTO** puede ser utilizada como un comando directo para

ejecutar un programa a partir de una línea determinada, sin borrar previamente el contenido de la pantalla. También puede ser empleada para formar una sentencia incorporada en un programa. **GOTO** es seguida por un valor numérico. Por ejemplo

### **60 GOTO 350**

---

Cuando la instrucción conteniendo la palabra clave **GOTO** es ejecutada, el valor a continuación de **GOTO** es redondeado al número entero más próximo, y la computadora se desvíe encaminándose a la línea indicada. El empleo de una variable o expresión permite que el programa se desvíe a una línea calculada. Tenga presente que la computadora se desviará al llegar a la sentencia **GOTO**, aún si el número de línea mencionado en la sentencia no se encuentra en el programa. En este caso la computadora continuará con la primera sentencia que encuentre en esa posición.

- **Formato**  
**GOTO** expr num ent

---

## **IF** si

---

- **Posición en el teclado**  
**I**

### **Sentencia/comando**

---

La palabra clave **IF** siempre es utilizada conjuntamente con la palabra clave **THEN** (entonces) para producir una decisión que determina la acción subsecuente. Antes de tomar la decisión, la computadora examina ciertas condiciones, para determinar si son verdaderas o falsas. El efecto de estas dos palabras claves es indicarle a la computadora que "si se cumplen ciertas condiciones, entonces haga lo siguiente". Cuando llega a este punto del programa la computadora se encuentra frente a una encrucijada, y el programa le indica que si cierta condición es verdadera debe tomar un camino, y si no es verdadera,

debe tomar el otro camino.

• **Cómo emplear IF y THEN**  
IF forma parte de una sentencia con THEN. IF es seguido, en primer lugar, por un valor numérico o una condición; en segundo lugar por THEN; y a continuación por una o más sentencias válidas en BASIC. Por ejemplo

```
80 IF x THEN GOTO 250
240 IF a $ = "NO" THEN PRINT "FIN": STOP
```

Una constante, variable o expresión (por ejemplo x) será considerada como verdadera si tiene un valor diferente a 0. En este caso la sentencia a continuación de THEN y las demás sentencias en la misma línea de programa serán ejecutadas. Después el programa continúa a la línea que sigue la instrucción conteniendo la palabra clave THEN. La constante, variable o expresión serán consideradas como falsas cuando su valor es 0. En consecuencia, las sentencias a continuación no serán ejecutadas y el programa continuará a la línea inmediatamente después de la línea conteniendo la palabra clave IF. En el ejemplo la computadora no continuará a la línea 250 si el valor x es 0.

Si la condición especificada (a \$ = "NO") a continuación de la palabra clave IF es verdadera, entonces la computadora ejecutará las sentencias a continuación de THEN. Si la condición es falsa, la computadora continuará a la próxima línea. En el caso de nuestro ejemplo, si la variable a \$ tiene el valor "NO", la computadora visualizará el texto indicado entre comillas ("FIN") y ejecutará la instrucción "STOP" poniendo fin al programa. En cambio, si a \$ tiene cualquier otro valor, la computadora continuará el programa saltando a la próxima línea.

El Spectrum adjudica a una condición verdadera el

valor de 1 y a una condición falsa el valor de 0. Como resultado, la computadora reconocerá cualquier valor excepto cero como verdadero y considerará a 0 como falso. Es posible asignar el valor de una condición a una variable mediante una sentencia como la siguiente

```
70 LET x = a$ = "NO"
```

Recuerde que, a diferencia de lo que sucede en otros dialectos del BASIC, en el Spectrum la palabra clave THEN no puede ser omitida antes de GOTO.

• **Formato**  
IF expr-num THEN [:sentencia]  
[sentencia]  
IF condición THEN sentencia  
[sentencia]

#### IN Entrada

• **Posición en el teclado**  
MODO EXTENDIDO  
SYMBOL SHIFT I

#### Función

IN controla el estado del teclado y otros equipos que sirven para ingresar información a la computadora (input) y de salida (out put). Esta palabra clave lee un byte de una puerta de entrada/salida dada, que indica el estado del dispositivo conectado a esa puerta.

• **Cómo emplear IN**  
IN es seguido de un valor numérico, por ejemplo:

```
150 LET x = IN y
```

El valor a continuación de IN puede variar entre 0 y 65535. la función de este valor es indicar la dirección de la puerta de entrada/salida a ser leída. IN devuelve el valor del byte leído.

• **Direcciones del teclado**  
El teclado tiene ocho direcciones, cada una de las cuales contiene uno de cinco bytes diferentes, dependiendo de la tecla oprimida en cada caso. Las direcciones son las siguientes: 65278, 65022,

64510, 63486, 61438, 57342, 49150 y 32766. Los valores de los bytes en esas direcciones pueden ser: 175, 183, 187, 189 o 190.

#### Formato

IN num-const  
IN num-var  
IN (expr-num)

#### INK Tinta

• **Posición en el teclado**

MODO EXTENDIDO  
SYMBOL SHIFT X  
Sentencia/comando

La palabra clave INK especifica el color de los elementos en primer plano visualizados en la pantalla. Estos elementos incluyen caracteres, puntos, líneas rectas y curvas.

• **Cómo emplear INK**  
INK puede ser empleada como un comando directo, pero es utilizada normalmente como parte de una sentencia en un programa. INK es seguida por un valor numérico, por ejemplo:

```
70 INK x
```

El valor numérico a continuación de INK es redondeado al número entero más próximo, y puede variar entre 0 y 9. Los números clave de los colores para los elementos en primer plano son los siguientes:

0	Negro
1	Azul
2	Rojo
3	Magenta (morado)
4	Verde
5	Celeste
6	Amarillo
7	Blanco
8	Transparente
9	Negro o blanco de contraste

La sentencia INK 8 indica que se mantenga el color sin cambiar en cualquier lugar de la pantalla donde se use INK 8.

La sentencia INK 9 modifica el color de la tinta a blanco o negro para que contraste con

el color de fondo o papel.

- **Colores de tinta (INK)**  
globales o locales

Cuando la sentencia está formada solamente por la palabra INK, como en el ejemplo anterior, el efecto es global (el color es global), y como resultado todas las demás imágenes serán visualizadas en ese color de tinta. La palabra clave INK también puede ser insertada en sentencias formadas por las palabras claves: PRINT, INPUT, PLOT, DRAW y CIRCLE.

INK es ubicada a continuación de la palabra clave, pero precede la información o datos y los parámetros controlando la visualización de la información. INK es seguida por los mismos valores y un punto y coma. Por ejemplo:

```
60 CIRCLE INK 4; 128, 88, 87
```

En este caso el efecto de la palabra clave INK es local y se aplica únicamente a los caracteres, puntos, líneas rectas o curvas, ejecutadas por la computadora en cumplimiento de las instrucciones contenidas en esa línea de programa. La sentencia en el ejemplo le indica a la computadora que dibuje un círculo de color verde. Una vez completada esta instrucción local la computadora retornará al color de tinta global o al color normal (negro).

- **Formato**  
INK expr num ent [:]

### INKEY\$ Input KEY

- **Posición en el teclado**  
MODULO EXTENDIDO  
N

#### Función

La palabra clave INKEY\$ es utilizada para detectar si se ha oprimido alguna de las teclas en el teclado de la computadora.

- **Cómo emplear INKEY\$**  
La palabra clave INKEY\$ no requiere un argumento y por

lo general es utilizada para asignar un carácter determinado a una variable o controlar un carácter en particular. Por ejemplo:

```
70 LET a$ = INKEY$
130 IF INKEY$ = "N" THEN
STOP
```

Al ejecutarse la instrucción, INKEY\$ retorna el valor del carácter producido por la tecla oprimida en ese momento. Si no se oprime una tecla, entonces INKEY\$ retorna una cadena de caracteres vacía (" "). Es importante recordar que INKEY\$ distingue entre mayúsculas y minúsculas y otros caracteres cambiados (shift). Para detectar cualquier tecla sin tomar en cuenta el carácter específico producido por ella, se utilizará la palabra clave IN.

A diferencia de la palabra clave INPUT, INKEY\$ hace avanzar inmediatamente la computadora a la sentencia siguiente. Por lo tanto, normalmente INKEY\$ es ubicada dentro de un ciclo que se repite constantemente hasta que la tecla requerida es oprimida.

- **Ejemplo**

Esta línea suspende la operación hasta que se introduce la letra "y", sin cambio a mayúsculas (CAPS SHIFT) ni cierre de mayúsculas (CAPS LOCK).

```
60 IF INKEY$ <> "y" THEN
GOTO 60
```

- **Formato**  
INKEY\$

### INPUT

Entrada de datos

- **Posición en el teclado**  
I

#### Sentencia/comando

La palabra clave INPUT permite ingresar información o datos (data) en la computadora durante la ejecución de un programa.

- **Cómo emplear INPUT**  
La palabra clave INPUT forma parte de

una sentencia contenida en un programa y se utiliza de manera muy similar a PRINT. En su forma más sencilla, INPUT es seguida por una variable numérica o de cadena de caracteres, por ejemplo:

```
60 INPUT x
90 INPUT a$
```

Al llegar a esta instrucción la computadora espera hasta que algún número o cadena se ingrese en la computadora. A medida que la información es ingresada a través del teclado, el valor o dato es visualizado en la línea inferior de la pantalla. Al oprimirse la tecla ENTER el valor o dato ingresado es asignado a la variable indicada y el programa continúa.

Una sentencia INPUT puede contener más de una variable y puede visualizar mensajes advirtiendo que es necesario ingresar ciertos datos o números. Este efecto se logra exactamente de la misma forma que con la palabra clave PRINT, utilizando comillas para delimitar el mensaje pidiendo la información, incluyendo los puntos y comas, y comas, necesarios para separar los diferentes elementos. Es posible incluir sentencias tales como INK, FLASH y PAPER en la sentencia conteniendo INPUT. Por ejemplo:

```
60 INPUT INK2; "Cómo te llamas?"; n$; ("Cuántos años tienes?"; + n$ + "?"); edad
```

Tome nota de las siguientes diferencias entre sentencias conteniendo INPUT y PRINT. INPUT espera cuando llega a una variable, por lo tanto todas las variables y expresiones (por ejemplo la variable incluyendo el signo \$ en el ejemplo) que serán incluidas en la pregunta, deben ser encerradas entre paréntesis. El ordenador visualiza el mensaje al pie de la pantalla, y las líneas ascienden a medida que se van agregando nuevos elementos. Se puede incluir la palabra clave AT en una

sentencia INPUT, como en el caso de sentencias PRINT. La palabra clave AT 0,0 visualiza el mensaje al principio de la línea encima de la última línea en la pantalla. La imagen en la pantalla ascenderá en la pantalla si se visualizan más de dos líneas.

#### • Cómo interrumpir INPUT

Cuando INPUT es seguida por una variable numérica y se introduce la palabra clave STOP, el programa se detendrá. En el caso de las variables de cadena de caracteres para interrumpir la operación de la palabra clave INPUT se seguirá el procedimiento siguiente: borrar (delete) el primer par de comillas que aparezca en la pantalla, y a continuación introducir la palabra clave STOP.

#### • Empleando INPUT con LINE

La sentencia INPUT LINE solamente puede ser utilizada con variables de cadena de caracteres. Normalmente la combinación de INPUT con una variable de cadena causa que la computadora visualice dos pares de comillas en la pantalla. A medida que la cadena de caracteres es ingresada mediante el teclado en la computadora, su contenido aparecerá en la pantalla, entre las comillas. Para borrar las comillas, utilice la sentencia INPUT LINE seguida de la variable de cadena. Si se requiere un mensaje al usuario para que ingrese la información, ese mensaje es situado entre las palabras clave INPUT y LINE, por ejemplo:

```
70 INPUT "Cómo te llamas?";
LINE n$
```

#### • Formato

```
INPUT [pregunta] [:I:]
var-num
INPUT [pregunta] [:I:]
var de cadena
INPUT [pregunta] [:I:] LINE
var de cadena.
[pregunta] = [const de
```

cadena]  
[ (expr de cadena) ] [AT expr  
num ent, expr num ent]  
[sentencia] [:I:]

**INT** INTEger Número Entero

• Posición en el teclado  
MODULO EXTENDIDO  
R

#### Función

La palabra clave INT transforma números fraccionarios en números enteros suprimiendo la parte fraccionaria.

#### • Cómo emplear INT

La palabra clave INT es seguida por un valor numérico, por ejemplo:

```
70 LET x = INT y
```

Una expresión debe estar *encerrada entre paréntesis*. INT produce el valor del número fraccionario *redondeando en menos* al número entero más próximo.

#### • Ejemplo

La siguiente sentencia contiene dos números fraccionarios, uno de ellos negativo. Al introducirse el comando PRINT

```
PRINT INT 45.67,INT -77.6
```

La computadora visualizará lo siguiente en la pantalla.

```
45 -78
```

#### • Formato

```
INT const-num
INT var-num
INT (expr-num)
```

**INVERSE** Invertir

• Posición en el teclado  
MODULO EXTENDIDO  
M

#### Sentencia/comando

La palabra clave INVERSE produce el efecto de invertir los colores en las posiciones de los caracteres, con el resultado de que los colores de tinta se tornan colores del papel y viceversa.

#### • Cómo emplear INVERSE

Normalmente INVERSE es utilizado para formar una sentencia contenida en un

programa. INVERSE es seguido por un valor numérico. Por ejemplo:

```
70 INVERSE 1
```

El valor numérico a continuación de la palabra clave INVERSE es redondeando al número entero más próximo, y puede ser 0 o 1. El efecto de INVERSE 1 es causar que todas las imágenes visualizadas en ejecución de las palabras clave PRINT y INPUT sean producidas en colores invertidos. En cambio, INVERSE 0 restituye el color normal del papel y la tinta.

Observe que la palabra clave INVERSE puede ser incluida dentro de una sentencia controlando la visualización de información en la pantalla, como sucede con la palabra clave INK. Al usar INVERSE 1 en CIRCLE, PLOT o DRAW, el punto o línea se dibuja en color del papel, y en consecuencia desaparece.

#### • Formato

```
INVERSE expr número entero
```

**LEN** LENgth Largo de una cadena

• Posición en el teclado  
MODULO EXTENDIDO  
K

#### Función

La palabra clave indica el largo o extensión de una cadena de caracteres (string)

#### • Cómo emplear LEN

LEN es seguida por un valor de cadena, por ejemplo:

```
50 LET x = LEN a$
```

Una expresión debe estar *encerrada entre paréntesis*. LEN informa cuál es la cantidad de caracteres en la cadena.

#### • Ejemplo

La siguiente línea

```
120 INPUT a$:IF LEN a$>9
THEN GOTO 120
```

pasa solamente cadenas conteniendo menos de diez caracteres.

• **Formato**

LEN const de cadena

LEN var de cadena

LEN (expr de cadena)

## LET

• **Posición en el teclado**

L

Sentencia/comando

La palabra clave LET es utilizada para asignar un valor a una variable. En el dialecto de BASIC utilizado por el Spectrum siempre es necesario incluir LET en las sentencias que asignan valores a variables.

• **Cómo emplear LET**

Usualmente LET forma parte de una sentencia componente de un programa, aunque también puede ser utilizada como un comando directo. Esta palabra clave es seguida por una variable de cadena de caracteres o una variable numérica, un signo de igual, y a continuación un valor. El valor puede ser numérico o una cadena de caracteres dependiente de la naturaleza de la variable. Por ejemplo:

```
60 LET x = x + 1
```

```
80 LET a$ = "Correcto"
```

Este valor es asignado entonces a la variable indicada (x en el primer ejemplo y a\$ en el segundo). Las variables simples (son variables sin subíndices) se mantienen indefinidas hasta que se les asigna un valor mediante sentencias

LET, READ o INPUT.

Las variables contenidas en tablas (arrays) reciben inicialmente los valores 0 o cadena de caracteres vacía (" "). (Ver DIM).

• **Formato**

LET var-num = expr-num

LET var de cadena = expr de cadena

## LIST Listar

• **Posición en el teclado**

K

Comando/sentencia

Al ejecutar la instrucción LIST la computadora visualizará en la pantalla una lista del programa contenido en su memoria en ese momento.

• **Cómo emplear LIST**

Normalmente LIST es utilizado como un comando directo, aunque también puede formar parte de una sentencia en un programa. Cuando se trata de hacer una lista del contenido de un programa completo, bastará con ingresar la palabra clave LIST sin más información. Después del comando directo

LIST

La computadora visualizará en la pantalla la primera y subsecuentes páginas conteniendo las líneas del programa. Este proceso se cumplirá por páginas. Para desplazar las páginas hacia arriba (scroll) bastará con oprimir cualquier tecla excepto: N, la barra espaciadora, STOP O BREAK.

LIST también puede ser seguido del número de una línea determinada como un valor numérico, por ejemplo:

```
LIST 100
```

El valor numérico a continuación de LIST será redondeado al número entero más próximo si fuera necesario. A continuación la computadora visualizará una lista comenzando a partir de esta línea. Si no existe una línea con ese número, la computadora comenzará su lista en la próxima línea.

• **Formato**

LIST [expr num ent]

## LLIST

• **Posición en el teclado**

MODO EXTENDIDO

V

Sentencia/comando

Esta palabra clave le indica a la impresora que imprima una copia de la lista del programa contenido en la memoria de la computadora en ese momento.

• **Cómo emplear LLIST**

LLIST es utilizado exactamente de la misma forma que LIST (ver LIST por más detalles). Observe que la información visualizada en la pantalla no cambiará mientras se está imprimiendo el listado.

• **Formato**

LLIST [expr núm ent]

**LN** Logarithm(Natural)      Logaritmo (natural)

• **Posición en el teclado**

MODO EXTENDIDO

Z

Función

LN suministra el logaritmo natural (en base e) de un valor especificado. Actúa como el inverso de EXP.

• **Cómo emplear LN**

La palabra clave LN es seguida por un valor numérico. Por ejemplo:

```
60 LET x = LN y
```

Una expresión debe estar encerrada entre paréntesis. El valor numérico a continuación de LN debe ser mayor que 0. LN produce el logaritmo natural de este valor.

• **Formato**

LN const-num

LN var-num

LN (expr-num)

## LOAD Cargar

• **Posición en el teclado**

J

Comando/sentencia

La palabra clave LOAD le indica a la computadora que cargue un programa completo almacenado en cinta, depositándolo en su memoria.

• **Cómo emplear LOAD**

Normalmente LOAD es utilizado como un comando

directo, pero también puede formar parte de un programa para cargar otro programa.

La palabra clave **LOAD** es seguida del nombre del programa a ser cargado. Los nombres de los programas son un valor de cadena con una extensión máxima de diez caracteres. Por ejemplo:

**LOAD "nombre"**

Los efectos de **LOAD** son drásticos: al ejecutarse esta instrucción la computadora borra el programa actualmente en la memoria de la computadora, así como todos los valores de sus variables. A continuación el Spectrum busca el nombre del nuevo programa en la cinta y lo carga.

Recuerde que en el caso de los nombres de programas la computadora distingue entre letras mayúsculas y minúsculas.

Cuando **LOAD** es seguido por una cadena de caracteres vacía, como en el ejemplo:

**LOAD " "**

el Spectrum cargará el primer programa completo que encuentre.

Observe que **LOAD** es utilizado en forma diferente cuando la computadora ha sido conectada a una unidad Microdrive. Ver el Manual de Microdrive e Interface y el punto de contacto Interface 1, por detalles adicionales.

#### • Formato

**LOAD** expr de cadena

### LOAD CODE

#### • Posición en el teclado MODO EXTENDIDO I

#### Comando/sentencia

La palabra clave **LOAD CODE** es empleada para indicarle a la computadora que cargue en su memoria cierta información, que ha sido previamente almacenada en cinta.

La información consiste en una serie de bytes, las cuales serán enviadas a diversas direcciones en la

memoria de la computadora.

Es posible hacer uso de **LOAD CODE**, por ejemplo, para cargar una imagen gráfica, u otro tipo de información para gráficas definidas por el usuario.

#### • Cómo emplear LOAD CODE

**LOAD CODE** puede ser utilizado como un comando directo o, alternativamente, como parte de una sentencia en un programa.

La sintaxis de esta palabra clave es la siguiente: en primer lugar **LOAD**, luego el nombre del programa entre comillas, finalmente **CODE**. Por ejemplo:

**LOAD "datos" CODE**

El nombre de programa a continuación de **LOAD** es el nombre correspondiente a la información registrada en cinta que se desea cargar a la computadora. A este nombre se le aplican las mismas limitaciones que en el caso de la palabra clave **LOAD** (ver: **LOAD**).

Cuando la computadora llega a esta sentencia en el programa —o cuando recibe el comando directo— procede a buscar en la cinta la información, almacenada bajo ese nombre. Cuando la computadora encuentra la información, visualizará Bytes: seguidos por el nombre de ese grupo de datos.

A continuación el Spectrum carga los bytes en su memoria distribuyéndolos en las direcciones a partir de las cuales fueron registradas previamente. La nueva información cargada en ejecución de este comando es superpuesta sobre toda otra información anterior en la memoria de la computadora.

**CODE** también puede ser seguido por uno o dos valores numéricos separados por una coma.

**LOAD "dibujo" CODE  
16384,6912**

Los valores numéricos incluidos a continuación de **CODE** serán redondeados al número entero más cercano

si fuera necesario. Su propósito es definir la dirección inicial desde donde se cargará la información indicada en la sentencia (16384 en el ejemplo), y la cantidad de bytes que se enviarán a las ubicaciones comenzando a partir de esa dirección.

Si el número indicado no es el correcto, la computadora visualizará el mensaje de error de carga de la cinta (tape loading error).

Cuando **CODE** es seguido por solamente un valor numérico, la computadora interpreta que se trata del número definiendo la posición de la dirección inicial a partir de la cual deberán ubicarse todos los bytes.

El resultado del ejemplo de más arriba también puede ser obtenido mediante las palabras claves **LOAD SCREEN\$**.

Por detalles acerca del procedimiento empleado para almacenar bytes, ver la sección **SAVE CODE**.

#### • Formato

**LOAD** expr de cadena **CODE**  
[expr num ent] [,expr num ent]

### LOAD DATA

#### • Posición en el teclado MODO EXTENDIDO D

#### Sentencia/comando

Las palabras clave **LOAD DATA** son empleadas para cargar tablas (arrays) almacenadas en cinta. Las tablas son grabadas en cinta mediante una instrucción **SAVE DATA**.

#### • Cómo emplear LOAD DATA

Es posible emplear las palabras claves **LOAD DATA** como parte de una sentencia en un programa o como un comando directo. El orden seguido es el siguiente: a continuación de **LOAD** se incluye el nombre que identifica la información grabada en la cinta (un valor de cadena). A continuación se incluye la palabra clave **DATA** y una letra o una letra y un signo de \$. El último elemento de la sentencia es

un par de paréntesis vacíos. Por ejemplo:

```
270 LOAD "números" DATA n()
300 LOAD "nombres" DATA n$()
```

El nombre incluido a continuación de LOAD es el nombre de la tabla grabada en la cinta. Este nombre se encuentra sujeto a las mismas restricciones que los demás nombres de programas cuando se utiliza la palabra clave LOAD. La letra o letra\$

Luego de DATA es el nombre que se le da a la tabla en el programa cuando se la carga y usa.

Al ejecutarse esta instrucción, el Spectrum busca la tabla indicada en la cinta. Cuando la encuentra la computadora visualiza el mensaje "Number array:" o "Character array:" en la pantalla, seguido del nombre. A continuación la computadora carga la tabla en su memoria.

Cuando la computadora carga la tabla grabada en cinta, la deposita en su memoria: la computadora borrará cualquier tabla designada con la misma letra nombre preexistente en su memoria (en el ejemplo los nombres de las tablas son "n" y "n\$"); finalmente, la computadora formará una nueva tabla designada con la letra de nombre especificada.

#### • Formato

LOAD expr de cadena DATA letra [\$()]

### LOAD SCREEN\$

#### • Posición en el teclado

MODO EXTENDIDO  
SYMBOL SHIFT  
K

#### Sentencia/comando

LOAD SCREEN\$ le indica a la computadora que cargue la información referente a una imagen de la pantalla de la televisión almacenada en

cinta.

Como resultado de esta instrucción, la computadora tomará la información de la cinta y la enviará a las secciones de su memoria controlando la visualización de imágenes en la pantalla de la televisión, para generar la imagen.

#### • Cómo emplear LOAD SCREEN\$

LOAD SCREEN\$ puede ser integrado en una sentencia incluida en un programa o puede ser utilizado como un comando directo.

LOAD es seguido del nombre bajo el cual la información fue grabada en la cinta. Este nombre es un valor de cadena. Finalmente se agrega SCREEN\$. Por ejemplo:

```
LOAD "dibujo" SCREEN$
```

El nombre entre comillas a continuación de LOAD es el nombre que identifica la información almacenada en la cinta, y se encuentra sometido a las mismas restricciones que los nombres de programas cuando se utiliza la palabra clave LOAD.

La imagen se forma gradualmente en la pantalla con los colores de tinta y papel activos en ese momento y luego toma los colores definidos como atributos.

Por más detalles acerca del procedimiento para almacenar información contenida en la pantalla, ver SAVE SCREEN\$.

#### • Formato

LOAD expr de cadena SCREEN\$

### LPRINT Imprimir en papel

• Posición en el teclado  
MODO EXTENDIDO  
C

#### Sentencia/comando

La palabra clave LPRINT le indica a la impresora que imprima una determinada información, de la misma forma que la palabra clave PRINT le indica a la computadora que visualice una determinada información en la pantalla de la televisión.

#### • Cómo emplear LPRINT

LPRINT puede ser incluido como parte de una sentencia contenida en un programa, o puede ser utilizado como un comando directo. Es seguido por elementos de información o datos (data) que pueden ser separados por puntos y comas, comas o apóstrofes, por ejemplo:

```
60 LPRINT "Número";x;
"Nombre",n$, "Edad";a
```

Al ejecutar esta instrucción la computadora transfiere la información indicada a la impresora, la cual lo imprime. El formato será el mismo que hubiera sido utilizado para visualizar la información en la pantalla en ejecución de la instrucción PRINT.

Una instrucción conteniendo LPRINT también puede incluir sentencias TAB, algunos controles CHR\$, sentencias INVERSE y OVER, y códigos de control, con los mismos resultados que cuando se utiliza PRINT.

#### • Formato

```
LPRINT [TAB expr
num ent;]
[AT expr num ent, expr
num ent;] [CHR$(expr
num ent;)] [Sentencia;]
[expr num ent] [expr
de cadena;] [, ] [']
```

### MERGE Fundir o combinar

• Posición en el teclado  
MODO EXTENDIDO  
SYMBOL SHIFT  
T

#### Sentencia/comando

La palabra clave MERGE permite fusionar dos programas.

#### • Cómo emplear MERGE

MERGE puede ser incluida en una sentencia parte de un programa, o como un comando directo.

MERGE es seguida por el nombre de la información grabada en cinta como un valor de cadena de caracteres, por ejemplo:

```
500 MERGE "prog2"
```

El nombre a continuación de

**MERGE** ("prog2" en el ejemplo) es el nombre del programa almacenado en cinta que se desea combinar con el programa que ya se encuentra en la memoria de la computadora.

Este nombre se encuentra sometido a las mismas limitaciones que los nombres de programas utilizados en sentencias **LOAD**.

La computadora ejecuta la instrucción **MERGE**, cargando el programa almacenado en cinta sin borrar el otro programa en su memoria. Sin embargo, el nuevo programa sobrescribe cualquier línea con el mismo número en el programa existente, así como las variables con el mismo nombre.

- **Fermate**  
**MERGE** expr de cadena

## **MOVE**

Comando para la operación de almacenamiento de información. Ver Manual de Sistema de discos 2020 o de Microdrive de Interface 1.

## **NEW**

- **Posición en el teclado**  
**A**

Comando/sentencia

Cuando se introduce la palabra clave **NEW** la computadora borra el contenido del área de memoria **BASIC** (el área hasta el límite **RAMTOP**), removiendo todos los programas que se encuentran en ese sector de su memoria.

- **Cómo emplear NEW**  
Por lo general **NEW** es utilizado como un comando directo, pero también puede formar parte de una sentencia en un programa. Es utilizado independientemente.

Al ejecutar esta instrucción la computadora borra el programa y variables en su memoria hasta el límite **RAMTOP**. Como resultado, las gráficas definidas por el usuario

conservadas en un área por encima del límite **RAMTOP** no son afectadas por **NEW**.

- **Fermate**  
**NEW**

## **NEXT** *Siguiente*

- **Posición en el teclado**  
**N**

Sentencia/comando

La palabra clave **NEXT** siempre es utilizada en conjunto con **FOR** para construir un ciclo **FOR NEXT**.

- **Cómo emplear NEXT**  
**NEXT** se utiliza normalmente en una sentencia contenida en un programa y como parte de un ciclo **FOR NEXT**. Es seguida por una letra. Esta última es la variable de control del ciclo. Por ejemplo:

```
90 NEXT a
```

En este **BASIC** la variable de control debe ser especificada.

Por más detalles acerca de los ciclos (loops) **FOR NEXT** ver la sección sobre **FOR NEXT**.

- **Fermate**  
**NEXT** Letra

## **NOT** *No*

- **Posición en el teclado**  
**SYMBOL SHIFT S**

Operador lógico/función

La palabra clave **NOT** es empleada para invertir la veracidad de una condición, causando que una condición falsa se torne en verdadera y viceversa.

- **Cómo emplear NOT**  
**NOT** es seguida por una condición o un valor numérico. Dos ejemplos:

```
90 IF NOT x = y + z THEN PRINT "Error"
90 LET correcto = x = y + z:
IF NOT correcto THEN PRINT "Error"
```

Cuando **NOT** es seguida por una condición ( $x = y + z$  en el ejemplo), el **Spectrum** opera de la siguiente forma:

primero asigna un valor de 1 a la condición si es verdadera y 0 si es falsa. **NOT** opera entonces como una función, invirtiendo el valor producido, para poder examinar la validez contraria de la condición.

Observe que si la condición contiene las palabras clave **AND** u **OR** debe estar encerrada entre paréntesis.

Cuando **NOT** es seguida por un valor numérico, retornará un 0 si el valor numérico que le sigue no es 0, y un número 1 si el valor que le sigue es 0. Por lo tanto, en los ejemplos anteriores: el **Spectrum** visualizará "Error" si  $x < > y + z$ , o si correcto tiene un valor de 0.

- **Fermate**  
**NOT** cond  
**NOT** expr-num

## **OPEN #**

Comando utilizado para controlar el almacenamiento de información. Ver Manual de Sistema de discos 2020 o Manual de los Microdrives. Interface 1.

## **OR**

- **Posición en el teclado**  
**SYMBOL SHIFT U**

Operador lógico/función

**OR** actúa como un operador lógico para establecer la veracidad de una combinación de condiciones. Si una o más de las condiciones son verdaderas, entonces la condición total es verdadera. **OR** también opera como una función para ejecutar operaciones binarias sobre dos valores numéricos.

- **Cómo emplear OR**  
Como un operador lógico, **OR** enlaza dos condiciones en una sentencia donde se trata de establecer la veracidad o falsedad del

conjunto total. Por ejemplo:

```
70 IF INKEY$ = "N" OR
INKEY$ = "n" THEN STOP
```

Si cualquiera de las dos condiciones indicadas en la sentencia es verdadera, entonces la computadora llega a la conclusión de que la totalidad de la sentencia es verdadera y obedece la instrucción (THEN STOP).

En el ejemplo una de las condiciones (INKEY\$ = "N" o INKEY\$ = "n") se transforma en verdadera tan pronto como se oprime la tecla "N" en el teclado de la computadora indistintamente de que se trate de la "N" mayúscula o minúscula. En los dos casos (mayúscula o minúscula) la totalidad de la combinación en la sentencia se ha transformado en verdadera y el programa se detiene obedeciendo la instrucción THEN STOP.

#### • OR como una función

El Spectrum asigna un valor numérico de 1 a la condición verdadera y de 0 a la condición falsa. La palabra clave reconoce cualquier valor distinto de 0 como verdadero y 0 como falso. Por lo tanto, OR puede ser precedido o seguido por un valor numérico, por ejemplo:

```
40 LEY x = y OR z
```

En el ejemplo, la computadora asignará un valor de 1 a la variable x si el valor de z es cualquier número excepto 0. Alternativamente, si el valor de z es 0 o una condición falsa, la computadora asignará un valor y.

Recuerde que el Spectrum no evalúa combinaciones de valores numéricos de acuerdo con las tablas de veracidad estándar.

#### • Formato

cond OR cond  
expr-num OR expr-num

### OUT Salida

• Posición en el teclado  
MODO EXTENDIDO  
SYMBOL SHIFT O

### Sentencia/comando

La palabra clave OUT envía un byte a la dirección correspondiente a una "puerta" o punto de conexión de entrada/salida de la computadora.

#### • Cómo emplear OUT

OUT puede ser empleada para formar una sentencia contenida en un programa o como un comando directo independiente. OUT es seguida por dos valores numéricos, separados por una coma. Por ejemplo:

```
40 OUT 254,3
```

Ambos valores son redondeados al número entero más próximo. El primer valor (254 en el ejemplo) puede variar entre 0 y 65535 y es la dirección de la puerta o punto de conexión. El segundo valor (3 en el ejemplo) puede variar entre 0 y 255, siendo el byte enviado a la dirección especificada anteriormente.

Como se explicó, cada byte se compone de una serie de bits. En el ejemplo los bits 0-2 del byte enviado a la dirección de punto de contacto (port address) 254 definen el color de borde o border. En el ejemplo, esta instrucción cambiará el color del borde a magenta. El bit número 3 enviado a la misma dirección, controla el enchufe de MIC, y el bit 4 controla el altavoz. La dirección de "la puerta", o punto de contacto 251 controla la impresora, y las direcciones de punto de contacto 254, 247 y 239 son utilizadas para controlar otros equipos periféricos conectados a la computadora.

#### • Formato

OUT expr num ent,  
expr  
num ent

### OVER Sobreimpresión

• Posición en el teclado  
MODO EXTENDIDO  
SYMBOL SHIFT N

Sentencia/comando

La palabra clave OVER es utilizada para sobreimprimir un carácter encima de otro. También puede ser empleada para dibujar puntos o trazar líneas rectas o curvas de color de papel (paper) en lugar de color de tinta (ink).

#### • Cómo emplear OVER

Por lo general OVER es utilizada para formar una sentencia parte de un programa. Es seguida por un valor numérico. Por ejemplo:

```
80 OVER 1
```

El valor numérico a continuación de la palabra clave OVER es redondeado al número entero más próximo, y puede ser 0 o 1. El valor normal es OVER 0, y determina que un nuevo carácter impreso en una posición de carácter borre completamente lo que había antes en la misma posición tomando su lugar.

En cambio el valor OVER 1 pone en operación un efecto especial: el nuevo carácter se agregará al carácter anterior en la misma posición, combinándose con él.

OVER puede ser insertado en una sentencia PRINT o INPUT, como sucede con INK. En este caso su efecto será local, OVER afectará únicamente los caracteres visualizados en ejecución de esa sentencia. Por ejemplo, la sentencia siguiente produce el efecto de subrayar una palabra.

```
60 PRINT AT 11, 15; "SI";  
OVER 1; AT 11, 15; " _ "
```

Sin embargo, observe que los caracteres se combinan de tal forma que, en el sector donde los dos caracteres se sobreponen, la computadora visualiza el color de papel (paper).

#### • OVER en el modo de alta resolución

OVER puede ser utilizado conjuntamente con PLOT, DRAW, y CIRCLE. Cuando

no se incluye la palabra clave **OVER** las líneas rectas y curvas pueden superponerse las unas a las otras, pero deben ser del mismo color de tinta, para evitar que cambie de color de la totalidad de la posición donde ambas líneas se cruzan. En cambio, si se incluye la palabra clave **OVER 1**, cuando dos líneas rectas o curvas se cruzan, o cuando se encuentran con algún otro carácter, se visualizará el color del papel. Otro efecto interesante es que si se dibujan puntos, o se trazan dos líneas rectas o curvas, exactamente en las mismas posiciones, en presencia de la instrucción **OVER 1**, los elementos sobrepuestos desaparecerán.

- **Formato**  
**OVER** expr num ent

## **PAPER** Papel

- **Posición en el teclado**  
**MODO EXTENDIDO**  
**SYMBOL SHIFT C**

### **Sentencia/comando**

**PAPER** es utilizado para seleccionar el color del papel o fondo de la pantalla. Este color puede ser el de la totalidad del fondo, o solamente del fondo de ciertos caracteres, puntos o líneas visualizados en las posiciones de caracteres individuales.

- **Cómo emplear PAPER**  
La palabra clave **PAPER** puede ser utilizada para formar una sentencia incluida en un programa o como un comando directo. **PAPER** es seguido por un valor numérico, por ejemplo:

**80 PAPER x**

Este valor numérico es redondeado al número entero más próximo, y puede variar entre 0 y 9. Los números código para los colores de **PAPER** son similares a los utilizados con la palabra clave **INK** (ver la sección correspondiente).

Como en el caso de **INK**, los colores pueden ser globales o locales. Para hacerlos locales, la palabra clave **PAPER** es insertada en sentencias controlando la visualización de imágenes en la pantalla, exactamente de la misma manera que con los colores de tinta (ver la sección **INK**).

Cuando se visualizan caracteres en la pantalla a continuación de una sentencia incluyendo la palabra clave **PAPER**, ya sea que la instrucción tenga un efecto global o local, el color del fondo de la posición del carácter correspondiente cambiará al color seleccionado. Esto se aplica también cuando se dibujan puntos, o se trazan líneas rectas o curvas, mediante instrucciones conteniendo una sentencia de **PAPER** (efecto local), pero no a continuación de una sentencia o comando de efecto global.

Para producir un fondo de color que abarque la totalidad del área de visualización de la pantalla, es necesario utilizar la palabra clave **CLS** después de una sentencia conteniendo **PAPER**. Como resultado toda la imagen en la pantalla se torna del color indicado, el cual permanece como el color de fondo.

- **Formato**  
**PAPER** expr num ent [:]

## **PAUSE** Pausa

- **Posición en el teclado**  
**M**

### **Sentencia/comando**

El efecto de la palabra clave **PAUSE** es suspender la ejecución del programa por un periodo de tiempo definido o indefinido.

- **Cómo emplear PAUSE**  
La palabra clave **PAUSE** es utilizada normalmente como parte de una sentencia en un programa. Es seguida de un valor numérico, por ejemplo:

**130 PAUSE 100**

El valor numérico a continuación de **PAUSE** es redondeado al número entero más próximo. El número puede variar entre 0 y 65535 y define la duración de la pausa en términos de imágenes de televisión (50 imágenes por segundo en Europa o 60 en los Estados Unidos). Por ejemplo, un valor de 50 produce una pausa de un segundo en Europa y en otros países donde la frecuencia de las imágenes es de 50Hz.

Sin embargo, la pausa puede ser interrumpida presionando una tecla. La instrucción **PAUSE 0** crea una pausa de duración ilimitada que continúa hasta que es interrumpida oprimiendo una tecla.

- **Formato**  
**PAUSE** expr num ent

## **PEEK** Mirar

- **Posición en el teclado**  
**MODO EXTENDIDO**  
**O**

### **Función**

**PEEK** suministra el valor del byte almacenado en una dirección particular de la memoria de la computadora.

- **Cómo emplear PEEK**  
La palabra clave **PEEK** es seguida por un valor numérico, por ejemplo:

**80 LET x = PEEK (256\*y)**

Observe que la expresión debe ser encerrada entre paréntesis. El valor numérico a continuación de **PEEK** es redondeado al número entero más próximo si fuera necesario. El número puede variar entre 0 y 65535 e identifica una determinada dirección en la memoria. Como resultado de esta instrucción la computadora suministra el valor del byte (un número entre 0 y 255) en la dirección especificada.

- **Ejemplo**  
La cantidad de imágenes (frames) de televisión ocurridas desde el momento en que se encendió el Spectrum es almacenada en

las direcciones 23672 a 23674. Como esas imágenes son producidas a una velocidad constante, explorar estas ubicaciones de la memoria mediante una sentencia conteniendo la palabra clave PEEK, ofrece un método preciso para medir el tiempo. En el siguiente ejemplo, la línea de programa le indica a la computadora que indique cuanto tiempo ha transcurrido desde que el Spectrum fue encendido (medido en segundos) descontando el tiempo utilizado en producir sonidos y operar periféricos tales como el grabador de cassettes o la impresora.

```
10 PRINT (PEEK 23672 + 256 *  
PEEK 23673 + 65536 * PEEK  
23674)/50
```

*NOTA: Cuando la frecuencia de la electricidad es de 60 Hz, el valor 50 debe ser cambiado a 60.*

#### • Formato

PEEK const num ent  
PEEK var num ent  
PEEK (expres num ent)

### PI $\pi$

#### • Posición en el teclado MODO EXTENDIDO M

#### Función

PI suministra el valor de pi ( $\pi$ ) para ser utilizado en cálculos.

#### • Cómo emplear PI

Esta palabra clave no requiere valores o variables complementarias cuando es utilizada en una sentencia o comando, por ejemplo:

```
DRAW 25,0,—PI
```

PI suministra un valor de 3,1415927, y como resultado del comando en el ejemplo le indicará a la computadora que trace un semicírculo de gran tamaño en la pantalla.

#### • Formato

PI

### PLOT Dibujar

#### • Posición en el teclado

### Q

#### Sentencia/comando

PLOT es utilizado en la producción de gráficas de alta resolución, para dibujar un pixel o punto de color, en una posición específica de la pantalla.

#### • Cómo emplear PLOT

La palabra clave PLOT puede formar parte de una sentencia incluida en un programa o también puede ser utilizada como un comando. Es seguida por dos valores numéricos separados por una coma, por ejemplo:

```
50 PLOT 128,87
```

Los dos valores numéricos después de PLOT serán redondeados al número entero más próximo si fuera necesario. El primer valor puede variar en 0 y 255, su función es definir la coordenada horizontal de la posición en la pantalla. El segundo valor puede variar entre 0 y 175, su función es definir la coordenada vertical de la posición.

La computadora ejecuta las instrucciones dibujando un pixel en el punto indicado, utilizando el color de tinta efectivo en ese momento.

Tome nota de los siguientes efectos producidos sobre la instrucción PLOT por las sentencias o comandos referentes a los colores. Después de OVER 1, el color de un punto existente en la misma posición es cambiado al color del papel (paper). A continuación de INVERSE 1, el punto es dibujado en el mismo color del papel en ese momento. Después de BRIGHT 1 o FLASH 1, toda la posición del carácter en la pantalla de baja resolución donde se dibuja el pixel será de color brillante o "relampagueará" (flash).

Estas cuatro palabras claves e INK también pueden ser insertadas en una sentencia conteniendo la palabra clave PLOT. El procedimiento a seguir es el mismo que en el caso de la

palabra PRINT. Por ejemplo:

```
100 PLOT INK 2;x,y
```

El efecto de estas palabras claves es el mismo que antes, con una diferencia importante: será un efecto limitado al pixel dibujado en ejecución de la sentencia. Cuando se inserta la palabra clave PAPER en una sentencia conteniendo PLOT, el efecto de la instrucción será cambiar el color de papel de toda la posición del carácter donde se encuentra el pixel.

#### • Formato

PLOT [sentencia:] expr  
num ent, expr num ent.

### POINT Punto

#### • Posición en el teclado MODO EXTENDIDO SYMBOL SHIFT 8

#### Función

La palabra clave POINT es empleada para determinar si el color de una posición determinada en la pantalla de alta resolución es un color de tinta o de papel.

#### • Cómo emplear POINT

POINT es seguido por dos valores numéricos separados por una coma y encerrados entre paréntesis. Por ejemplo:

```
240 IF POINT (x,y) = 1 THEN  
GOSUB 600
```

Los dos valores después de POINT son redondeados al número entero más próximo si fuera necesario. La función del primer valor es definir la coordenada horizontal de un pixel en la pantalla y puede variar entre 0 y 255. El segundo valor define la coordenada vertical y puede variar entre 0 y 175. Al recibir la instrucción POINT, la computadora responderá con un número 1, si el pixel en la posición indicada es de color de tinta y con un 0 si es de color de papel.

#### • Formato

POINT (expr num ent,  
expr num ent)

**POKE** ingresar

- Posición en el teclado  
O

Sentencia/comando

La palabra clave **POKE** es utilizada para cambiar el valor de un byte situado en una dirección específica de la memoria de la computadora.

- **Cómo emplear POKE**  
**POKE** puede formar parte de una sentencia incluida en un programa o también puede operar como un comando. Es seguido por dos valores numéricos separados por una coma. Por ejemplo:

**POKE 23609,255**

Los dos valores a continuación de **POKE** son redondeados al número entero más próximo si fuera necesario. El primer número es una dirección en la memoria RAM de la computadora, pudiendo variar entre 16384 y 65535. El segundo valor puede variar entre 0 y 255, y es el nuevo byte a escribir en la dirección de la memoria dada.

- **Fermate**  
**POKE expr num ent,**  
**expr num ent.**

**PRINT** Imprimir

- Posición en el teclado  
P

Sentencia/comando

La sentencia **PRINT** muestra datos en la pantalla. Puede incorporar otras palabras clave para definir la posición y color de la información visualizada.

- **Cómo emplear PRINT**  
**PRINT** puede ser empleada independientemente o puede ser seguida por datos. Estos datos pueden ser cualquier tipo de expresión numérica o de cadena de caracteres, o una combinación de ambos tipos.

Cuando se emplea **PRINT** con datos, dos o más elementos deben ser separados por un punto y coma, una coma o un

apóstrofe.

También es posible insertar ciertas palabras claves entre **PRINT** y los datos, en cualquier orden, siempre y cuando la sentencia formada por aquella palabra clave termine en un punto y coma. Estas palabras claves son las siguientes: **CHR\$, TAB, AT, INK, PAPER, FLASH, BRIGHT, INVERSE** y **OVER**.

- **Empleando PRINT con cadenas**  
El efecto de la palabra clave **PRINT** sola, o seguida por una cadena de caracteres vacía (""), será visualizar una línea blanca en la pantalla y desplazar el cursor al principio de la línea siguiente.

**PRINT** seguida por una cadena tendrá el efecto de visualizar en la pantalla los caracteres entre las comillas. El comando

**PRINT "3/542/76/21"**

producirá el siguiente resultado en la pantalla:  
3/542/76/21

**PRINT** seguida por una variable o expresión de cadena visualizará o "imprimirá" en la pantalla la cadena de caracteres.

- **PRINT con números**  
**PRINT** seguido por cualquier expresión numérica visualizará en la pantalla el valor de la expresión. Los números son notación decimal con hasta ocho dígitos significativos de un número.

Los números muy grandes y muy pequeños son visualizados en notación científica, como dos figuras separadas por la letra **E**. Esta indica un número en el cual la primera parte (la mantisa) es elevado a la potencia de la segunda parte (el exponente). El comando

**PRINT 3/542/76/21**

por ejemplo, producirá el siguiente resultado en la pantalla:  
3.4688798E-6

- **Impresión con PRINT mediante los signos de puntuación**

**PRINT** seguido por elementos de información, datos, separados por un punto y coma, visualiza dichos elementos uno junto al otro. El comando

**PRINT 1;2;3**

producirá

123

**PRINT** es seguida por elementos de información separados por comas, la computadora visualizará cada uno de esos elementos en el centro o al principio de una línea en la pantalla, dependiendo de la posición del primer elemento de información. El comando

**PRINT 1,2,3**

producirá el siguiente resultado

1	2
3	

**PRINT** seguida por elementos de información separados por apóstrofes, visualizará cada uno de dichos elementos al principio de la línea siguiente. El apóstrofe opera como un "punto y aparte". El comando

**PRINT 1'2'3'**

producirá el siguiente resultado:

1
2
3

**PRINT** termina en un punto y coma, coma o apóstrofe, la información visualizada por la computadora ejecutando la próxima sentencia **PRINT** será afectada de la misma forma.

- **PRINT y otras palabras claves**  
**PRINT** puede ser seguida por **TAB**, un valor numérico, punto y coma y un elemento de información. Por ejemplo:

**60 PRINT TAB x;a\$**

El valor a continuación de **TAB** (x en el ejemplo) será redondeado al número entero más próximo si fuera necesario. A continuación ese valor será dividido por

32 y se lleva el resto para producir un valor entre 0 y 31.

PRINT puede ser seguido por la palabra clave AT y dos valores numéricos separados por una coma, un punto y coma y un elemento de información. Por ejemplo:

```
50 PRINT AT 1, 0;  
"Información"
```

El primer valor (la variable I en el ejemplo) define la línea horizontal en la pantalla donde se "imprimirá" o visualizará la información. Este valor puede variar entre 0 y 21. El segundo valor, la variable c en el ejemplo, define el número de la columna donde se imprimirá el primer carácter o dígito de la información. El número de columna puede variar entre 0 y 31. El comando PRINT AT 11,16;"\*" escribe una estrella en el centro de la pantalla.

PRINT también puede ser seguida por una o más funciones CHR\$. Por más detalles consulte la sección sobre CHR\$.

#### • PRINT y las palabras claves controlando los colores

Lo escrito por el comando PRINT en la pantalla del televisor es afectado por las sentencias y comandos referentes al color y conteniendo la palabra clave INK, PAPER, FLASH, BRIGHT, INVERSE y OVER que se encontraban incluidas en las líneas anteriores del programa. Estas seis instrucciones pueden ir a continuación de PRINT, separándolas entre sí con punto y coma. Por ejemplo:

```
50 PRINT AT 11,16; INK 2;  
FLASH 1; "*" "
```

Después de haber ejecutado la sentencia, la computadora retornará a los valores normales o a los valores globales preestablecidos. PRINT también obedecerá los códigos de control del color locales insertados conjuntamente con la información (ver la página 33).

#### • Formato

```
PRINT [TAB expr  
num ent;]  
[AT expr num ent. expr  
num ent;]  
[CHR$ (expr num ent;)  
[sentencia;] [expr numérica]  
[expr de cadena];;:]
```

### RANDOMIZE (RAND)

#### • Posición en el teclado T

#### Sentencia/comando

La palabra clave RANDOMIZE es abreviada en el teclado como RAND, y se utiliza en conjunción con RND (sin embargo no debe confundirse la una con la otra) para generar una secuencia de números aleatorios (al azar) y que, por lo tanto, no pueden ser predecidos, o una secuencia de números predecibles.

#### • Cómo emplear RANDOMIZE

La palabra clave RANDOMIZE es utilizada dentro de una sentencia incluida en un programa o como un comando. Esta palabra clave puede ser utilizada independientemente o seguida de un valor numérico. Por ejemplo:

```
RANDOMIZE 1  
10 RANDOMIZE
```

El valor numérico a continuación de RANDOMIZE será redondeado al número entero más próximo si fuera necesario. El valor numérico puede variar entre 0 y 65535. Un valor superior a 0 ajusta el valor de la variable de sistema SEED a este valor. Como resultado la palabra clave RND siempre generará la misma secuencia de números (por información referente a variables de sistema ver página 48). La secuencia de los números dependerá del valor de RANDOMIZE.

En cambio, cuando RANDOMIZE es seguida por 0 o ningún valor, entonces la variable de sistema SEED recibirá el valor de otra

variable de sistema denominado FRAMES. Esta variable FRAMES cuenta la cantidad de imágenes visualizadas por la televisión desde el momento cuando se encendió el Spectrum. SEED se incrementa cada 50 o 60 imágenes

—dependiendo del tipo de televisión— (ver sección sobre PAUSE). Debido a que SEED cambia 50 o 60 veces por segundo, la secuencia de números generados por una sentencia conteniendo RANDOMIZE y a continuación RND o solamente RANDOMIZE 0, ofrecerá un efecto sumamente aleatorio.

#### • Formato

```
RANDOMIZE [expr num ent]
```

### READ Leer

#### • Posición en el teclado MODO EXTENDIDO A

#### Sentencia/comando

La palabra clave READ es utilizada en combinación con la palabra clave DATA, y tiene el propósito de asignar valores a variables, haciendo uso de los valores especificados en sentencia de DATA.

#### • Cómo emplear READ

Normalmente READ es utilizada para formar una sentencia contenida en un programa.

Es seguida por una o más variables numéricas o variables de cadena de caracteres separadas entre sí por comas. Por ejemplo:

```
20 READ a$,x
```

Cuando la instrucción conteniendo la palabra clave READ se ejecuta por primera vez, la computadora se dirigirá a la línea conteniendo la palabra clave DATA y tomará la misma cantidad de valores que variables, comenzando a partir del primer elemento en la lista de valores a continuación de DATA. Los valores contenidos en la lista de DATA son asignados a las variables en la sentencia

READ sucesivamente y siguiendo su orden en la sentencia.

• **Formato**

READ var-num [,var-num]  
[var de cadena]  
READ var de cadena [,var-num]  
[var de cadena]

---

**REM** REMark Comentario

---

• **Posición en el teclado**

E

Sentencia

---

REM tiene el propósito de indicarle a la computadora que la información a continuación es solamente un comentario u observación. Esta puede consistir, por ejemplo, en el título y nombre del autor del programa, explicaciones acerca de las líneas siguientes en el programa, por ejemplo el propósito de una variable. La sentencia REM es incluida en el programa para la información del usuario, pero no interviene en la operación del programa. La sentencia REM no es visualizada en la pantalla excepto cuando se hace un listado del programa (LIST).

• **Cómo emplear REM**

REM forma parte de una línea independiente del programa o también puede ser incluida como el elemento final de otra línea. La palabra clave REM es seguida por los comentarios, y no requiere comillas. Por ejemplo:

```
10 REM El autor de este
programa es Pedro
80 INPUT n$: REM n$
es el nombre
```

Cuando la computadora llega a una línea encabezada por la palabra clave REM ignorará el resto de la información en la misma línea.

• **Formato**

REM [cualquier carácter]

---

**RESTORE** Restaurar

---

• **Posición en el teclado**

**MODO EXTENDIDO**  
S

Sentencia/comando

La palabra clave RESTORE es utilizada junto con READ y DATA para modificar el mecanismo normal de READ DATA. Su función es indicarle a la computadora que, al ejecutar la instrucción READ, tome los valores de una determinada sentencia conteniendo la palabra clave DATA, en lugar de seguir el procedimiento normal, de dirigirse a la primera sentencia de DATA en el programa (ver DATA y READ).

• **Cómo emplear RESTORE**

Por lo general RESTORE es incluida dentro de una sentencia parte de un programa. El programador tiene la opción de agregar a continuación de ella algún valor numérico. Por ejemplo:

```
160 RESTORE 800
```

El valor numérico a continuación de RESTORE es redondeado al número entero más próximo si fuera necesario. La computadora asume que el valor numérico es el número de la línea de programa conteniendo una sentencia de DATA. Después de haber encontrado la palabra clave RESTORE, la computadora procederá de la manera siguiente: al encontrar una línea de programa con la palabra clave READ, se encaminará a la línea 800 del programa y tomará los valores de DATA especificados en ella, para asignarlos a las variables en aquella línea. El procedimiento mediante el cual un valor es adjudicado a una variable es similar al utilizado con READ DATA (ver las secciones DATA y READ).

Si la línea de DATA (línea 800 en el ejemplo) no existe o no contiene una palabra clave DATA, entonces la computadora continuará hasta la próxima sentencia de DATA.

Cuando RESTORE es

seguida por el número 0 o si no es seguida por un valor numérico, entonces la computadora al encontrar una sentencia con la palabra clave READ avanzará hasta la próxima línea conteniendo la palabra clave DATA.

• **Formato**

RESTORE [expr num ent]

---

**RETURN** Regresar

---

• **Posición en el teclado**

Y

Sentencia/comando

La palabra clave RETURN es utilizada para terminar una subrutina e indicarle a la computadora que regrese al programa principal o a una subrutina previa.

• **Cómo emplear RETURN**

Normalmente RETURN es utilizada como parte de una sentencia en un programa. RETURN es utilizada independientemente, ubicada al final de la subrutina. Por ejemplo:

```
1000 RETURN
```

Al ejecutar esta instrucción la computadora regresa a la sentencia inmediatamente después de la última palabra clave GOSUB ejecutada.

Por más detalles consultar la sección sobre GOSUB.

• **Formato**

RETURN

---

**RND** RaNDom number Núm aleatorio

---

• **Posición en el teclado**

MODO EXTENDIDO

T

Función

Esta palabra clave es utilizada para generar un número aleatorio (al azar)

• **Cómo emplear RND**

La palabra clave es utilizada independientemente, como parte de una sentencia en un programa o un comando. Por ejemplo:

```
60 LET x = RND
```

Al ejecutar esta instrucción la computadora genera un número al azar o aleatorio, menor que 1 y mayor o igual

a 0.

Cuando la Spectrum es encendida, reajustada (control reset), o se introduce la instrucción NEW, los números invocados mediante RND son producidos en la misma secuencia. Esta secuencia es generada tomando las potencias de 75 (75, 75\*75, 75\*75\*75, y así sucesivamente) y dividiendo cada potencia por 65537, y utilizando solamente el residuo entonces se sustrae 1 del residuo y se divide este resultado por 65536, etcétera.

Cuando se requiere una sentencia más aleatoria u otra secuencia predefinida se incluirá la palabra clave RANDOMIZE antes de RND.

• **RANDOM y números enteros**  
Si se desea obtener números enteros entre 1 y x, se utilizará la palabra clave INT, en la fórmula  $\text{INT}(\text{RND} * x) + 1$ . Para generar un número entero entre 0 y x, se utilizará la fórmula  $\text{INT}(\text{RND} * x + 0,5)$

• **Formato**  
**RND**

## **RUN** Ejecutar

• **Posición en el teclado**  
**R**

Comando/sentencia

La palabra clave RUN le indica a la computadora que ejecute el programa, normalmente a partir de su primera línea.

• **Cómo emplear RUN**  
RUN puede ser empleado como un comando directo o puede formar parte de una sentencia de un programa. Es posible agregar un valor numérico a continuación de RUN. Por ejemplo:

**RUN 50**

Cuando RUN no es seguido por un valor numérico, la computadora ejecutará el programa a partir de la primera línea del programa. Cuando se incluye un valor

numérico después de RUN, éste será redondeado al número entero más próximo si fuera necesario y el programa será ejecutado a partir de esa línea. Si la computadora no encuentra la línea de ese número, entonces comenzará a ejecutar el programa a partir de la línea inmediatamente siguiente.

Cuando se carga en la memoria de la computadora un programa almacenado en cinta mediante la instrucción LINE, la computadora lo ejecutará inmediatamente y no será necesario utilizar la instrucción RUN.

• **Formato**  
**RUN [expr num ent]**

## **SAVE** Grabar

• **Posición en el teclado**  
**S**

Comando/sentencia

La palabra clave SAVE le indica a la computadora que grabe en cinta el programa en su memoria, para almacenarlo.

• **Cómo emplear SAVE**  
SAVE es utilizado por lo general como un comando directo, aunque también puede formar parte de una sentencia contenida en un programa.

**SAVE "ProgramaX"**

El nombre del programa puede contener hasta diez caracteres. Al ejecutarse estas instrucciones la computadora visualizará el siguiente mensaje en pantalla:

"start tape, then press any key", lo cual significa ponga en funcionamiento el grabador, luego oprima cualquier tecla

Cuando la cinta comienza a pasar frente a la cabeza de grabación de su grabador el usuario oprimirá cualquier tecla, y a continuación la computadora comenzará a transmitir el contenido de su memoria RAM a la cinta. Al terminar de grabar, la

computadora visualizará en la pantalla, el reporte 0 OK,0:1.

• **Ejecución automática**  
Para indicarle a la computadora que ejecute un programa pregrabado automáticamente, inmediatamente después de haberlo cargado en su memoria, se utilizarán las palabras claves SAVE y LINE al grabar en cinta el programa. La sintaxis es la siguiente: el nombre del programa, seguido por la palabra clave LINE y un valor numérico, por ejemplo.

**SAVE "programaX" LINE 1**

El valor numérico a continuación de LINE será redondeado al número entero más próximo si fuera necesario, y será un valor 1 o el número de una línea de programa.

Al ejecutar estas instrucciones, la computadora grabará el programa en cinta, como cuando ejecuta la instrucción SAVE.

Cuando la computadora carga el programa pregrabado, lo pondrá en ejecución automáticamente a partir de la línea indicada por el valor numérico. Si no encuentra una línea de ese número, la computadora comenzará a ejecutar el programa a partir de la línea siguiente en el programa.

• **Formato**  
**SAVE expr de cadena [LINE expr núm ent.]**

## **SAVE CODE**

• **Posición en el teclado**  
**S**  
**MOD0 EXTENDIDO**  
**I**

Comando/sentencia

Las palabras clave SAVE CODE le indican a la computadora que envíe una determinada sección de información en su memoria al grabador de cassettes para ser grabada en cinta y almacenada. Esta información podrá ser utilizada nuevamente en el futuro. El procedimiento para cargar la información

almacenada en cinta en la memoria de la computadora es mediante las palabras clave LOAD CODE.

• **Cómo emplear SAVE CODE**

Las palabras clave SAVE CODE pueden ser utilizadas como un comando directo o ser incluidas en una sentencia contenida en un programa. La primera parte de la palabra clave,

SAVE va seguido por el nombre de archivo, que es una cadena de caracteres, y luego CODE, que va seguido por dos valores numéricos separados por una coma, por ejemplo:

**SAVE "dibujo" CODE 16384,6912**

El nombre que identifica la información grabada en cinta, insertado entre SAVE y CODE puede tener un máximo de diez caracteres.

Los dos valores numéricos después de CODE son redondeados al número entero más próximo si fuera necesario.

El primer número indica la dirección inicial (16384) de la información almacenada en la memoria. El segundo valor le indica a la computadora la cantidad de bytes que deben ser grabadas (6912 bytes en el ejemplo).

La computadora ejecuta esta instrucción y envía la información al grabador de cassettes para ser grabada como cuando se ejecuta la instrucción SAVE. (Ver la sección sobre SAVE.)

• **Formato**  
SAVE expr de cadena CODE expr num ent, expr num ent.

**SAVE DATA** Grabar datos

• **Posición en el teclado**  
S  
MODO EXTENDIDO  
D

**Sentencia/comando**  
La palabra clave SAVE DATA le indica a la computadora que registre una tabla (array)

en cinta magnética. Más tarde la tabla puede ser vuelta a cargar en la computadora utilizando las palabras clave LOAD DATA.

• **Cómo emplear SAVE DATA**

Las palabras clave SAVE DATA puede ser utilizada formando una sentencia en un programa o como un comando directo. SAVE es seguida por el nombre bajo el cual se grabará la tabla, a continuación se incluye la palabra DATA y luego una letra o una letra con el signo \$, y finalmente un par de paréntesis vacíos. Por ejemplo,

450 SAVE "números" DATA n)  
750 SAVE "nombres" DATA n\$()

El nombre de la tabla puede incluir hasta un máximo de diez caracteres. La letra sola o con el signo de \$ ubicada después de DATA será el nombre identificando la tabla en el programa. La tabla es grabada en cinta siguiendo el mismo procedimiento que en el caso de los programas grabados con la palabra clave SAVE.

• **Formato**  
SAVE expr de cadena DATA letra (\$)()

**SAVE SCREENS**  
Grabar pantalla

• **Posición en el teclado**  
S  
MODO EXTENDIDO  
SYMBOL SHIFT  
K

**Comando/sentencia**  
SAVE SCREEN\$ le indica a la computadora que grabe la información visualizada en la pantalla en cinta magnética. En el futuro la información podrá ser cargada nuevamente en la computadora mediante la instrucción LOAD SCREEN\$.

• **Cómo emplear SAVE SCREENS**  
La palabra clave SAVE SCREEN\$ puede ser incluida en una sentencia

componente de un programa o como un comando directo. El primer término, SAVE, es seguido por el nombre bajo el cual se registrará la imagen en la cinta. A continuación se inserta SCREEN\$

**SAVE "imagen" SCREEN\$**

El nombre que identificará la imagen o información visualizada en la pantalla, registrada en el cassette, puede tener hasta diez caracteres.

Cuando la computadora recibe esta instrucción, enviará la imagen o información al grabador para ser grabado en cinta, como cuando se ejecuta la palabra clave SAVE. (Ver la sección sobre SAVE).

• **Formato**  
SAVE expr de cadena SCREEN\$

**SCREEN\$** Pantalla

• **Posición en el teclado**  
MODO EXTENDIDO  
SYMBOL SHIFT  
K

**Función**  
El propósito de la función SCREEN\$ es detectar cuál caracter aparece en una posición determinada de la pantalla.

• **Cómo emplear SCREEN\$**  
La palabra clave SCREEN\$ es seguida por dos valores numéricos separados por una coma y encerrados entre paréntesis, por ejemplo

100 IF SCREN\$(I, c) = "x"  
THEN PRINT  
"CRASH!"

Los valores numéricos después de SCREEN\$, redondeados al número entero más próximo si fuera necesario, son las coordenadas de la posición en la pantalla. El primero de los valores (indicado en el ejemplo por la variable I) define el número de la línea horizontal sobre la cual se encuentra el punto, pudiendo variar entre 0 y 21. El segundo valor numérico (c) define la columna sobre

la cual se encuentra la posición. Este valor puede oscilar entre 0 y 31.

Cuando la computadora recibe esta instrucción, examinará el contenido de la posición especificada en la pantalla, y devolverá el carácter en esa posición como una constante de cadena (el carácter entre comillas "x", por ejemplo).

Si la computadora no encuentra un carácter en la posición indicada entonces se limitará a devolver una cadena vacía (" ").

#### • Formato

**SCREEN\$** (expr num ent, expr num ent).

### **SGN** signo

#### • Posición en el teclado

MODO EXTENDIDO  
F

#### Función

La palabra clave **SGN** sirve para determinar si un número es positivo, negativo o cero.

#### • Cómo emplear **SGN**

**SGN** es seguido por un valor numérico. Por ejemplo:

**50 LET x = SGN y**

Una expresión debe ser *encerrada entre paréntesis*. Obediendo la palabra clave **SGN** la computadora devolverá un número 1 si el argumento (y en el ejemplo) es positivo, -1 si el argumento es negativo y 0 si el argumento es cero.

#### • Formato

**SGN** const-num

**SGN** var-num

**SGN** (expr-num)

### **SIN** sine

Seno

#### • Posición en el teclado

MODO EXTENDIDO  
U

#### Función

La función **SIN** suministra el valor del seno de un ángulo.

#### • Cómo emplear **SIN**

**SIN** es seguido por un valor numérico. Por ejemplo:

**80 LET x = SIN y**

Una expresión debe ser *encerrada entre paréntesis*. El valor numérico insertado a continuación de **SIN** es el ángulo en radianes, y la función **SIN** indicará cuál es el seno correspondiente a ese ángulo.

Los grados pueden ser convertidos en radianes multiplicándolos por  $\pi/180$ .

Observe que **SIN** suministra un valor positivo para ángulos de entre 0 y 180 grados, y un valor negativo para ángulos de entre 180 y 360 grados.

#### • Formato

**SIN** const-num

**SIN** var-num

**SIN** (expr-num)

### **SQR** Square Root

Raíz cuadrada

#### • Posición en el teclado

MODO EXTENDIDO  
H

#### Función

**SQR** calcula la raíz cuadrada de un número.

#### • Cómo emplear **SQR**

La palabra clave **SQR** es seguida por un valor numérico. Por ejemplo:

**70 LET x = SQR y**

Una expresión debe ser *encerrada entre paréntesis*. El valor a continuación de **SQR** (y en el ejemplo) siempre debe ser mayor que cero, y la función **SQR** produce su raíz cuadrada.

#### • Formato

**SQR** const-num

**SQR** var-num

**SQR** (expr-num)

### **STOP**

#### • Posición en el teclado

SYMBOLSHIFT  
A

#### Sentencia/comando

**STOP** detiene la ejecución del programa en un punto determinado. Por ejemplo, esta palabra clave puede ser utilizada para terminar la sección principal de un

programa, lo que permitirá incluir las subrutinas en una sección separada, a continuación de la principal. **STOP** también es un auxiliar útil cuando se aplican técnicas para detectar y corregir los errores en programas.

#### • Cómo emplear **STOP**

Normalmente **STOP** forma parte de una sentencia contenida en un programa. Es utilizado independientemente, por ejemplo:

**650 STOP**

Durante la ejecución del programa la computadora se detendrá al llegar a esta línea y producirá el siguiente mensaje de error en la pantalla:

#### 9 STOP Statement

agregando el número de la línea y de la sentencia donde se ha detenido el programa.

Si se ingresa la instrucción **CONTINUE** la computadora reanudará el programa a partir de la sentencia siguiente, con los nuevos valores.

#### • Formato

**STOP**

### **STR\$** STRings

Cadena

#### • Posición en el teclado

MODO EXTENDIDO  
Y

#### Función

La palabra clave **STR\$** convierte números en cadenas (string).

#### • Cómo emplear **STR\$**

**STR\$** es seguido por un valor numérico, por ejemplo:

**90 LET a\$ = STR\$ x**

Una expresión debe estar *encerrada entre paréntesis*. Al ejecutar la instrucción **STR\$**, la computadora devolverá el valor de su argumento (x en el ejemplo) como constante de cadena de caracteres.

En el ejemplo, si se asigna un valor de 65 a x, entonces

la sentencia en el ejemplo le asignará un valor de "65" a a\$.

• **Formato**  
STR\$ const-num  
STR\$ var-num  
STR\$ (expr-num)

---

**TAB** Var. LPRINT, PRINT

---

**TAN** TANgent Tangente

• **Posición en el teclado**  
MODULO EXTENDIDO  
E

**Función**

TAN suministra la tangente de un ángulo.

• **Cómo emplear TAN**  
La palabra clave TAN es seguida por un valor numérico. Por ejemplo:

130 LET x = TAN y

Una expresión debe estar *encerrada entre paréntesis*. El valor a continuación de TAN es el ángulo en radianes, y TAN suministra el ángulo de la tangente.

Es posible convertir grados en radianes multiplicándolos por  $\pi/180$ .

• **Formato**  
TAN const-num  
TAN var-num  
TAN (expr-num)

---

**TO** Hasta

• **Posición en el teclado**  
SYMBOL SHIFT  
F

**Función**

La palabra clave TO tiene dos usos diferentes en el dialecto de BASIC utilizado.

En primer lugar, TO puede ser utilizado conjuntamente con la palabra clave FOR para construir un ciclo FOR NEXT (ver la sección FOR por más detalles).

En segundo lugar TO también puede ser utilizado para rebanar ("slicing") cadenas de caracteres (strings). El rebanado es la operación mediante la cual una cadena es dividida en

subcadenas más pequeñas.

Como se emplea TO para rebanar cadenas

• En este caso TO es utilizado para designar el primer y último carácter de la subcadena contenida en la cadena principal. La palabra clave TO es precedida por una variable de cadena de caracteres, a continuación se abre un paréntesis, seguido de un valor numérico. Después de TO se agrega otro valor numérico y se cierra el paréntesis. Por ejemplo:

80 PRINT a\$ (4 TO 7)

Una expresión de cadena también debe *ser encerrada entre paréntesis*. El valor de cadena (en el ejemplo a\$) identifica la cadena a ser rebanada. Los dos valores numéricos (en el ejemplo 4 y 7) le indican a la computadora las posiciones del primer y último carácter en la subcadena dentro de la cadena de caracteres principal. Al ejecutar la instrucción TO la computadora tomará el fragmento definido en la sentencia de la cadena principal y visualizará la subcadena (caracteres 4 a 7 de a\$).

Si no se incluye un número identificando el comienzo de la subcadena, la computadora asumirá que el valor numérico es 1. De la misma manera, si no se incluye un segundo número, para indicar donde termina la subcadena, entonces la computadora asume que ésta termina en el último carácter de la cadena de caracteres principal.

• **Formato**  
const de cadena ([expr-num])  
TO [expr-num]  
var de cadena ([expr-num]) TO [expr-num]  
(expr de cadena) ([expr-num])  
TO [expr-num]

---

**USR** User SubRoutine  
Subrutina del usuario

• **Posición en el teclado**  
MODULO EXTENDIDO  
L

**Función**

La palabra clave USR es empleada para invocar una subrutina en código de máquina depositada en una dirección especificada de la memoria de la computadora. USR también puede ser utilizada para depositar los datos (data) para gráficas definidas por el usuario en las ubicaciones reservadas del espacio en el tope de la memoria de la computadora.

• **USR y el código de máquina**  
Para utilizar el código de máquina, USR es seguido por un valor numérico, por ejemplo:

80 PRINT USR 65000  
100 RANDOMIZE USR 65000

Una expresión debe *ser encerrada entre paréntesis*. El valor numérico a continuación de la palabra clave USR será redondeado al número entero más próximo. Este valor es la dirección inicial en la memoria donde se ha depositado la subrutina en código de máquina. Cualquier sentencia en el programa conteniendo USR invocará dicha subrutina, y devolverá el valor del par de registros BC. RANDOMIZE o RESTORE USR, por ejemplo, solo ejecutan la subrutina, mientras que PRINT USR le ejecutará e imprimirá el valor del par de registros BC.

• **USR y gráficas definidas por el usuario**  
Para crear gráficas definidas por el usuario se empleará la palabra clave USR junto con la palabra clave POKE. USR es seguida por una constante de cadena o una variable para invocar una dirección para la sentencia conteniendo POKE. Por ejemplo:

5 POKE USR "a",255

El valor de cadena a continuación de USR puede ser una sola letra, en minúsculas o mayúsculas,

entre A y U. La computadora no distinguirá entre mayúsculas y minúsculas.

USR también suministrará la dirección inicial (dirección del primer byte) de una de las 21 secciones de la memoria especialmente reservadas para gráficas definidas por el usuario. Cada sección contiene ocho bytes, y mediante la palabra clave POKE es posible almacenar un número directamente en la posición en la memoria, para crear un carácter gráfico.

#### • Formato

USR const num ent  
USR var num ent  
USR (expr num ent)  
USR const de cadena  
USR var de cadena

**VAL** VALue Valor

• Posición en el teclado  
MODO EXTENDIDO  
J

#### Función

VAL convierte una cadena con valor numérico en un número.

#### • Cómo emplear VAL

VAL va seguido por una constante o variable de cadena de caracteres, por ejemplo:

70 LET x = VAL a\$

Al valor de la constante o variable de cadena se le sacan las comillas, debiendo luego ser un valor numérico. VAL lo evalúa y lo devuelve como constante numérica.

#### • Ejemplos

Si a\$ tiene el valor "435", entonces la sentencia del ejemplo asignará 435 a x. VAL también puede evaluar expresiones, por ejemplo:

10 INPUT a\$, x  
20 PRINT VAL a\$

El valor de cadena asignado a a\$ debe ser una expresión utilizando x, por ejemplo "x\*x". Luego se le asignará un valor numérico a x, por ejemplo 5. VAL elimina las

comillas para obtener x\*x, y evalúa el resultado utilizando el valor asignado a x. Por ejemplo si el valor de x era 5 en este caso el resultado será  $5*5 = 25$ .

#### • Formato

VAL const de cadena  
VAL var de cadena

**VAL** \$ VALue (string) Valor de cadena

• Posición en el teclado  
MODO EXTENDIDO  
SYMBOL SHIFT J

#### Función

VAL\$ evalúa una cadena como una expresión de cadena de caracteres. Su resultado es una cadena, como en caso de VAL, pero en lugar de producir un valor numérico como este último, VAL\$ produce otra cadena de caracteres.

#### • Cómo emplear VAL\$

VAL\$ es seguido por una variable de cadena, por ejemplo:

130 PRINT VAL\$ a\$

La computadora elimina las comillas del valor de la variable de cadena y lo que queda se evalúa como una cadena. VAL\$ evalúa esta última expresión y produce el resultado como una constante de cadena de caracteres.

#### • Ejemplos

Pruebe este programa

10 INPUT a\$, x\$  
20 PRINT VAL\$ a\$

El valor de cadena asignado a a\$ debe ser una expresión empleando x\$, por ejemplo "x\$ + x\$". La computadora evaluará x\$ como una cadena, por ejemplo "DO". La palabra clave VAL\$ elimina las comillas del valor atribuido a a\$ para obtener x\$ + x\$, y lo evalúa utilizando el valor asignado previamente a x\$. En este caso la computadora visualizará el resultado DODO.

#### • Formato

VAL\$ var de cadena

## VERIFY Verificar

• Posición en el teclado  
MODO EXTENDIDO  
SYMBOL SHIFT R

Para CODE:  
MODO EXTENDIDO  
I

Para DATA:  
MODO EXTENDIDO  
D

#### Sentencia/Comando

VERIFY verifica que un programa ha sido correctamente grabado en la cinta a continuación de SAVE.

#### • Cómo usar VERIFY

VERIFY es usado como un comando de la misma manera que LOAD y es seguido por el nombre del programa. Por ejemplo:

VERIFY "nombre"

Cuando se arranca la cinta se exhibe el nombre de cada programa encontrado y cualquier programa en la cinta que tenga el mismo nombre es comparado con el programa en la memoria. Si los dos son iguales, aparece el informe:

0 OK,0:1

VERIFY se usa en forma distinta con el Microdrive (ver manual).

#### • VERIFY CODE y VERIFY DATA

VERIFY CODE puede ser usado igual que LOAD CODE para verificar que una parte de la memoria ha sido bien grabada en cinta. VERIFY DATA funciona igual que LOAD DATA verificando que una tabla ha sido bien grabada en cinta. Vea LOAD CODE y LOAD DATA para más detalles.

#### • Formato

VERIFY expr de cadena  
VERIFY expr de cadena CODE [expr núm ent], expr núm ent  
VERIFY expr de cadena DATA letra (\$)(

## INFORMES EN LA PANTALLA

Cuando la computadora Spectrum deja de escribir BASIC, un informe aparece al fondo de la pantalla del televisor, indicando que un comando o un programa ha sido completado o que un error ha ocurrido. Cada informe consiste en un número de código o una letra seguida por un mensaje breve, y a consecuencia los números de la línea y la sentencia en los cuales paró la computadora. Un comando se indica como línea 0 y dentro de una línea, sentencia 1 se ubica al principio de la línea, sentencia 2 se ubica después de los primeros dos puntos o THEN, y así sucesivamente. CONTINUE por lo general hace que el programa comience de nuevo en la sentencia especificada en el informe.

### 0 OK

Se ha completado con éxito una operación determinada. Si se inserta la palabra clave CONTINUE la computadora ignorará este último informe y reanudará la ejecución del programa en la sentencia especificada en el informe previo.

### 1 NEXT without FOR

Al ejecutar el programa, la computadora ha encontrado una palabra clave NEXT, pero no ha podido encontrar su palabra clave complementaria FOR, pero existe una variable ordinaria con el mismo nombre.

### 2 Variable not found

Se ha utilizado una variable de control con la palabra clave NEXT, pero sin incluir previamente su palabra clave complementaria FOR.

Se ha incluido una variable con subíndice pero sin haber definido sus dimensiones mediante una sentencia de DIM, o cargado una tabla desde cinta magnética.

### 3 Subscript error

El subíndice de una variable se encuentra fuera de las dimensiones de la tabla correspondiente.

### 4 Out of memory

No queda suficiente espacio en la memoria de la computadora para completar la sentencia o comando.

### 5 Out of screen

Una sentencia INPUT ha generado más de 23 líneas en la mitad inferior de la pantalla, o se ha utilizado un número de líneas de 22 o más con la palabra clave PRINT AT.

### 6 Number too big

La computadora ha intentado producir un número mayor que  $10^{30}$  aproximadamente.

### 7 RETURN without GOSUB

La computadora ha encontrado una palabra clave RETURN pero no puede ubicar la palabra clave complementaria GOSUB correspondiente.

### 8 End of file

Informe sobre la manipulación de información grabada en la Microdrive.

### 9 STOP statement

La computadora ha encontrado una sentencia conteniendo la palabra clave STOP, CONTINUE, reanudará el programa a partir de la sentencia siguiente.

### A Invalid argument (Argumento erróneo)

Se ha dado un argumento o valor incorrecto.

### B Integer out of range

La computadora ha redondeado un número al número entero más próximo. El resultado queda fuera de los límites de la computadora y no puede ser aceptado por ésta.

### C Not seen in BASIC

La computadora ha encontrado una sentencia que no tiene sentido en BASIC y en el contexto el programa.

### D BREAK-CONT repeats (BREAK - CONT repite)

Se ha oprimido la tecla BREAK. El efecto de la palabra clave CONTINUE es repetir la sentencia donde la computadora se detuvo.

### E Out of DATA (No quedan datos)

Obedeciendo una instrucción READ, la computadora se ha dirigido a la lista de datos o información contenida en una sentencia encabezada por la palabra clave DATA.

### F Invalid file name

Se ha usado SAVE con un nombre con más de 10 caracteres.

### G No room for line

No hay suficiente sitio en la memoria de la computadora para admitir una nueva línea de programa.

### H STOP in INPUT (STOP al ejecutar instrucción)

Al ejecutar una instrucción INPUT, la computadora ha encontrado la palabra clave STOP. La palabra clave CONTINUE tiene el efecto de repetir la sentencia conteniendo INPUT.

### I FOR without NEXT (FOR sin NEXT)

La computadora ha encontrado un ciclo de FOR ... NEXT donde sus límites o los valores definidos por STEP están incorrectos (por ejemplo FOR x = 5 TO 8, sin un valor de STEP) y además no se ha encontrado la palabra clave NEXT complementaria.

### J Invalid I/O device

Informe sobre la manipulación de información grabada en la Microdrive.

### K Invalid colour (Color inválido)

El valor especificado en sentencias conteniendo las palabras claves INK, PAPER, FLASH, BRIGHT, INVERSE u OVER, o el carácter de control correspondiente se encuentra fuera de los límites de la computadora.

### L BREAK into program

Durante la ejecución de un programa se ha oprimido la tecla BREAK. Si se desea continuar ejecutando el programa bastará con ingresar la palabra clave CONTINUE.

### M RANSTOP too good

El valor específico para RANSTOP es demasiado grande o demasiado pequeño.

### N Statement lost

Durante la ejecución del programa la computadora ha encontrado una instrucción indicándole que avance a una sentencia que ya no existe.

### O Invalid stream

Información grabada en la Microdrive.

### P FN without DEF

La palabra clave FN ha sido usada sin su sentencia DEF FN correspondiente.

### Q Parameter error

Una sentencia conteniendo la palabra clave FN contiene una cantidad equivocada de valores o argumentos para ser pasados a la función. (Una cadena en lugar de número o viceversa).

### R Tape loading error

La lectura o verificación no se ha completado correctamente. La computadora no especificará cuál ha sido el error.

## MAS ALLA DEL BASIC

BASIC es un lenguaje general para computadoras que se desempeña bien en la mayoría de las aplicaciones. Existen disponibles programas (software) que le suministran otros lenguajes, tales como FORTH, micro-PROLOG, y LOGO. Estos lenguajes operan de una forma muy diferente a BASIC y le abren nuevas posibilidades a su computadora.

Debido a que BASIC es un lenguaje de propósitos generales, a veces puede llegar a ser algo incómodo cuando se trata de utilizar la computadora para ciertas aplicaciones especiales. También es comparativamente lento. Otros lenguajes pueden ofrecer una mayor flexibilidad combinada con una programación más sencilla y velocidad mayores. El lenguaje FORTH, por ejemplo, le permite definir sus propias palabras para programar. Estas palabras serán incluidas en instrucciones que su computadora comprenderá y ejecutará a velocidades diez veces superiores a las de los comandos equivalentes en BASIC.

En el caso del lenguaje micro-PROLOG, la computadora adquiere la capacidad de comprender frases en inglés sencillas, almacenándolas en su memoria como una base para desarrollar un diálogo con el usuario. LOGO es un lenguaje para computadoras desarrollado con fines educativos.

### El código de máquina o "machine code"

El propósito del lenguaje BASIC es permitir que el usuario pueda darle instrucciones a la computadora, en una forma fácil de comprender. La Unidad Procesadora Central, el poderoso Z80A no entiende BASIC. Las instrucciones en BASIC ingresadas en la computadora, por ejemplo desde el teclado, deben ser traducidas mediante un programa permanente, denominado Intérprete de BASIC (Basic Interpreter). Este programa se encuentra almacenado en una sección de la memoria de la computadora y convierte la instrucción en BASIC en una secuencia de señales de código, las cuáles son enviadas a la CPU.

El intérprete tomará algún tiempo para traducir las instrucciones en BASIC al lenguaje comprendido por la CPU Z80A conocido como código de máquina (machine code en inglés).

Es posible saltarse el intérprete, enviando instrucciones redactadas en código de máquina directamente a la Z80A en la CPU de

la computadora. En este caso su programa será ejecutado con gran rapidez.

Para tener una idea acerca de la rapidez con la computadora ejecuta instrucciones en código de máquina, le sugerimos que ingrese en la computadora, a través del teclado, el programa siguiente.

### FRANJAS INSTANTANEAS

```
10 FOR X=0 TO 15
20 READ N: POKE 65000+X,N
30 NEXT X
40 DATA 30,255,60,1,1,24,22
50 DATA 55
60 DATA 35,11,120,177,200,114,
24,240
70 RANDRIZE USP 65000
```



Pruebe cambiar el valor numérico 55, en la línea número 50 del programa, a cualquier valor entre 1 y 255, para cambiar las franjas.

También es posible producir franjas de colores insertando comandos conteniendo la palabra clave INK. Lo que deseamos demostrarle es la velocidad con que cambian las imágenes visualizadas en la pantalla por la computadora, cuando las instrucciones son en código de máquina.

Este programa funciona porque las sentencias de DATA contienen 16 códigos que son almacenados a partir de la dirección 65000 de la memoria, en ejecución de las instrucciones contenidas en las líneas 10 y 30 del programa. La línea 70 envía los códigos al microprocesador Z80A, lo que produce inmediatamente la imagen en la pantalla.

Para ayudarle a redactar sus propios programas en código de máquina, existen programas especiales denominados ensambladores (assemblers). Los ensambladores operan con una lista de instrucciones para ser ingresadas en la computadora en lugar de la lista de números utilizada por el código de la máquina. El propósito de ensamblador es traducir estas instrucciones a los números del código de máquina. Las instrucciones del ensamblador no son palabras clave en inglés como en BASIC.

## LA TERMINOLOGIA DE LAS COMPUTADORAS: UN BREVE DICCIONARIO

A continuación explicaremos los significados de algunos de los términos asociados con el manejo de computadoras que usted encontrará en esta publicación.

Las palabras en bastardilla han sido explicadas en otra parte de esta sección. Si desea averiguar el significado de algún término que no se encuentra incluido en esta sección le sugerimos que consulte el índice de esta publicación.

**Almacenar** (save): guardar un *programa* o *datos* en un dispositivo de almacenamiento como un cartucho de Microdrive o un cassette.

**Anidación** (nesting): la organización de ciclos dentro de un *programa* tal que uno o más ciclos se ejecutan dentro de otros.

**Argumento** (argument): un valor usado por una *función* para obtener un resultado.

**Atributos** (attributes): códigos que dan los colores de los caracteres.

**BASIC**: el lenguaje de computación utilizado por la Spectrum y la mayoría de las micro-computadoras personales.

**Bit**: un estado "encendido" o "apagado" en código binario. (Es una contracción de las palabras inglesas Binary digit).

**Byte**: un conjunto de ocho bits que representa un número con valor entre 0 y 255. Cada dirección de memoria puede contener un byte.

**Cadena de Caracteres** (string): un grupo de uno o más caracteres encerrados entre comillas (") para distinguirlos de números y variables numéricas.

**Character** (character): cualquier letra, dígito (0 a 9), signo o unidad gráfica, que se puede imprimir.

**Cargar** (load): alimentar la computadora con un *programa* o *datos* guardados en un dispositivo de almacenamiento como un disco o un cassette.

**Ciclo** (loop): una sección de un *programa* que se repite una o más veces.

**Código Binario** (binary code): el tipo de código que utilizan internamente las computadoras. Consiste de una secuencia de estados "encendido" o "apagado", por ejemplo pulsos eléctricos.

**Código de máquina** (machine code): o lenguaje de máquina, es el lenguaje que entiende la

Spectrum. Los programas en BASIC son traducidos a código de máquina por la computadora, a medida que los ejecuta.

**Comando** (command): una instrucción simple que es llevada a cabo por la computadora o un *comando directo*.

**Comando Directo** (direct comand): un conjunto de una o más instrucciones que se ejecuta inmediatamente al serle dado a la computadora.

**Concatenación** (concatenation): la combinación de cadenas de caracteres al sumarias.

**Conjunto de Caracteres** (character set): el conjunto completo de caracteres predefinidos y ciertos códigos de control utilizados por la computadora.

**Constante** (constant): un número o un grupo de uno o más letras u otros caracteres cualesquiera.

**C.P.U.** (unidad central de proceso) (Central Processing Unit): la parte principal de la computadora, que hace los cálculos y controla las otras unidades. La Spectrum usa como CPU un microprocesador Z80A.

**Cursor**: la posición en la pantalla donde se imprimirá lo que sigue. Puede estar marcado por un signo titilante que indica el modo en el que está la computadora.

**Datos** (Data): información que recibe la computadora de un *programa*, o que se le ingresa para producir resultados.

**Dirección** (address): una unidad de memoria. Hay 65536 direcciones en la Spectrum.

**Editar** (Edit): cambiar detalles dentro de un *programa*.

**Expresión** (expression): una combinación de constantes, variables, y palabras claves.

**Falso** (false): cualquier estado o resultado que la computadora decide que es incorrecto o no-verdadero. Falso tiene un valor numérico de 0.

**Función** (function): una operación en la que la computadora toma uno o más valores (o argumentos) y los usa para producir un resultado que es otro valor.

**Gráficos** (graphics): la producción de imágenes tales como dibujar diagramas o esquemas por la computadora.

**Hardware**: la computadora en sí y cualquier otra máquina o dispositivo asociado a ella, como los periféricos.

**Imprimir** (print): mostrar resultados o gráficos en la pantalla o en la impresora.

**Información** (information): palabras, números y signos en alguna combinación que la computadora debe manejar.

**Informe** (report): un mensaje dado por la computadora que informa de sus acciones.

**Ingresar** (enter): dar una instrucción completa o información a la computadora.

**Interfase** (interface): una unidad que conecta la computadora y/o periféricos entre sí, y que

se encarga de que se comuniquen.

**K** (Kilobytes): una medida de la capacidad de memoria de la computadora. 1K o Kilobyte equivale a 1024 bytes ( $2^{10}$ ). La capacidad de la memoria en K es igual a la cantidad total de direcciones de la memoria, cada una de las cuales puede contener un byte. La Spectrum tiene 48K de RAM y 16K de ROM dando un total de 64K.

**Línea** (line): una instrucción o conjunto de instrucciones en un programa. Se le da un número para que se ejecute en el momento correcto dentro de una secuencia de líneas.

**Listado** (listing): las líneas de un programa listadas en orden.

**Lógica** (logic): el proceso por el cual decide la computadora si los resultados están bien o mal, o si los estados son verdaderos o falsos.

**Memoria** (memory): la parte de la computadora que mantiene el programa y los datos cuando se le requiere, y también las instrucciones operativas permanentes del sistema BASIC.

**Modo** (mode): en la Spectrum, uno de los cinco estados que determinan qué palabra clave o caracteres serán producidos por cada tecla en el teclado. Durante la programación, el modo se indica mediante una letra titilante dentro del cursor.

**Operador** (operator): una instrucción que hace aritmética o lógica.

**Palabra clave** (keyword): una instrucción de computadora en BASIC. Puede requerir algunos valores para funcionar.

**Periférico** (peripheral): cualquier dispositivo que se conecta a la computadora.

**Pixel**: el más pequeño punto de color que puede aparecer en la pantalla. (Es contracción de las palabras inglesas "picture cell".)

**Programa** (program): una secuencia de instrucciones para ser ejecutada por la computadora.

**RAM** (memoria de acceso al azar) (Random Access Memory): la parte de la memoria donde se puede poner un programa y datos, y otros valores cambiantes. También conocida como la memoria volátil. Los contenidos de la memoria RAM se borran al desconectar la computadora. La Spectrum tiene 48 K de memoria RAM.

**Registro** (register): una pequeña unidad de memoria separada de la memoria principal. Los registros dentro de la CPU se usan en el proceso de la computación.

**Resolución** (resolution): el grado de detalle en los gráficos de la computadora.

**ROM** (Memoria de lectura solamente) (Read Only Memory): la parte de la memoria que contiene programas permanentes, e instrucciones para la computadora. La Spectrum tiene 16K de ROM.

**Salida** (output): los resultados producidos por

la computadora.

**Scroll**: el movimiento que permite que se vea una impresión que excede el tamaño de la pantalla.

**Sentencia** (statement): una palabra clave usada como instrucción en una línea de programa o la instrucción en ésta si.

**Sintaxis** (syntax): la secuencia correcta de palabras clave, constantes, variables y expresiones requeridas para formar una instrucción BASIC válida.

**Software**: cualquier programa, incluyendo programas permanentes en ROM, discos o cassettes.

**Tabla** (array): un grupo de datos relacionados que se guardan juntos en una sección de la memoria.

**Valor** (value): cualquier número o cadena de caracteres que puede estar dado o representado por una constante, variable o expresión.

**Variable**: una o más unidades de memoria que contienen una constante en particular a ser usadas por la computadora. A cada una se le da un nombre o letra para identificarlas fácilmente. La Spectrum distingue entre variables numéricas y variables de cadena de caracteres.

**Variable de cadena de caracteres** (String variable): una variable que contiene una cadena de caracteres. Las variables de cadenas de caracteres siempre tienen como nombre una letra seguida del signo \$.

**Variable numérica** (numeric variable): una variable que contiene un número. Las variables numéricas tienen como nombre una o más letras.

**Verdadero** (true): cualquier estado o resultado que la computadora decide que es verdadero o correcto. Verdadero tiene un valor numérico de 1.

# INDICE

Los números de página en bastardilla se refieren a ilustraciones o secciones especiales en las páginas indicadas.

Aleatorio (Random). Efectos 30  
Almacenes de programas 44, 45  
Alta resolución. Gráficas 26; 28-29  
Altavoz 43  
Alteración de programas 9  
Amplificación de sonido 37  
Animación 34-35  
Antena, cable y enchufe 4, 5  
Aritméticos. Operadores 22; 22  
ATTR 35

Baja resolución. Gráficas de 26-27  
Barra espaciadora (teclado) 19  
BASIC 18; 49-73  
BASIC de Spectrum, 49  
BEEP 36; 18  
BIN 33  
Binario. Código. 44  
Borde (Border). Color. 25-5; 6  
BREAK 19  
BRIGHT 31

Cálculos 22-23; 22-23  
Cadenas (Strings) 22  
CAPS LOCK 21; 18  
CAPS SHIFT 8, 21, 18  
Caracteres.  
Creación de. 32-33  
Selección 20  
Caracteres Definidos por el Usuario (CDU) 80; 32; 33  
Cargando programas en la computadora 13, 14-15; 14-16  
Cartuchos, ROM 12, 47; 47  
Cassettes, grabador. V: Grabador de cassettes  
Central Processing Unit (CPU) 43, 44, 48, 75, 43; 45  
Ciclos (loops) 26-27, 30  
Cinta. Cable de 46  
Cintas de cassette 12; 44, 45  
Almacenaje 12  
Cuidado de las cintas 12  
Rotulos 14, 39  
Sonido 12  
CIRCLE 28  
Circuitos integrados 42-43

Circuitos lógicos 43  
Código Binario 44  
Código de Máquina 75  
Colisiones 34, 35  
Color 24, 25; 24, 25  
Códigos 24  
Color de borde (Border) 24-5; 6  
Combinaciones 25  
Control, códigos 33  
Mezclas 32  
Pruebas 6; 24  
Teclas 19  
Comas 23, 51  
Comandos 22, 50  
Comillas 23, 51  
Conector de reborde 5, 43, 47  
Conexiones 5  
Grabadora de cassettes 13  
Electricidad 5  
Televisor 4  
Contacto. Puntos de (Interfaces) 45, 46-47  
CPU: V: Central processing Unit  
Cuadrícula de alta resolución 28; 80  
De baja resolución 26, 80  
Cursor. Controles de los 19  
DATA 33  
DELETE 10  
Dos puntos 23, 51  
DRAW 28, 29  
EAR. Enchufe 37; 5, 13  
EDIT 18, 21  
Editando líneas de programa 21  
Ejecución de programas (RUN) 8-9  
Electricidad. Enchufe 9V CD 5, 43  
Enchufes 5  
Enchufe MIC 37; 5, 13  
ENTER 9, 10, 11, 19  
Entrada-Salida. Caminos de 45  
Equipos (hardware). Definición 12  
Equipos accesorios 45, 46-47  
Errores, Corrección 10, 21  
Reportes en la pantalla 82  
EXTEND MODE 8, 21; 18  
Extend mode 21; 20  
Figuras. Diseño de 30-31  
De baja resolución 26-27  
Llenando figuras 29; 29  
FLASH 31  
FOR NEXT 26-27, 29, 30, 31, 34  
FORTH 75  
Funciones 50  
GOTO 23  
Grabadores de cassettes

Cargando programas 14-16  
Como amplificadores 37; 37  
Conexiones 5, 13; 13  
Contador de revoluciones 14.  
Control de tono 14, 15; 16  
Control de volumen 14, 15; 16  
Grabando programas 38-40  
Selección 12  
V: Cintas de cassette  
Grabando programas 13, 38-40  
Gráficas. Animación 34-5  
Alta resolución 26; 28-29  
Baja resolución 26-27  
Color 24-25  
Creando caracteres 32-33  
Dibujos 30-31  
Efectos aleatorios 30  
Llenando figuras 29; 29  
Gráficos 25; 25  
GRAPH 21; 18, 26  
Graphics mode. V: Modo gráficos  
IF THEN 29  
Impresoras 45, 47; 45, 47  
Ingresando Información 8, 9  
Interferencia de radio 4  
Ink. Color de 24-25  
INPUT 23, 29  
Interfaces (puntos de contacto) 45, 46-47  
INV VIDEO 18  
INVERSE 31  
Juego de caracteres 51  
LET 23  
Letra. Modo 21; 20  
Líneas 8  
Borrando líneas 21  
Editando líneas 21  
LIST 21  
Listado de programas 8, 21  
LOAD 14-16  
LOAD. V: Cargando programas  
LOGO 75  
Lopps. V: Ciclos  
Memoria 12; 42, 43, 44-48  
Memoria. Mapa de la 48  
MIC. Enchufe 37; 5, 13  
Micro-PROLOG 75  
MODEM (Modulador-Demodulador) 46  
Modos 20-21  
Modo Gráficas 21; 20  
Modo Letras 21; 20  
Modo Mayúsculas 21; 20  
Modo Palabras Clave 20; 20

- Música 36-37  
Tono 36
- NEW 11, 12; 18  
Números 50  
Nuevos programas 11
- Operadores aritméticos 22; 22
- Palabras clave 9, 18-19, 5, 52-73  
Selección de 19, 20  
Palabras clave. Modo 20; 20  
Palancas de control 45-47  
Pantalla. Movimiento  
vertical (Scroll) 9  
Papel (Paper). Color 24-25  
Paréntesis 23  
Pixel 28  
PLOT 28  
POKE 48  
PRINT 22  
Programación 17-40  
Programas  
Borrando líneas del 21  
Cargando programas  
12, 13; 14-15; 14-15  
Corrección de errores 10  
Edición 21  
Ejecución (RUNning) 8-9,  
44  
Iniciando nuevos pr. 11  
Introducción en la  
computadora 8-9, 44  
Grabando 3, 38-40  
Verificación de 39  
Volviendo a comenzar 10  
Programas pre-grabados 12-13;  
13  
Cargando estos  
programas 14-16; 14-16
- Programas adecuados  
para la Spectrum 12  
Tipos 12
- Puntuación. Signos de 23, 51  
Punto 23, 51  
Punto y coma 23, 51
- RAM (Random Access Memory)  
Memoria RAM 42, 48; 42,  
45  
Cartuchos RAM 4
- RAMTOP 48  
Random, V: Aleatorio  
READ 33  
Reajuste (Reset) 11, 12; 5  
Reiniciando programas 10  
REM 30  
Reportes en la pantalla 74  
RND 26, 30  
ROM (Read Only Memory)  
Memoria ROM 48; 43, 45  
ROM. Cartuchos 12, 47; 46-47  
RS232. Punto de conexión  
(Interface) 47; 45  
RUN, V: Ejecución de  
programas
- SAVE 39-39  
SAVE V: Grabando programas  
Scroll, V: Pantalla. Movimiento  
vertical
- Sentencias 22, 50  
Signos matemáticos 22, 50  
Seleccionando, 19  
Símbolos. Selección 20  
Sintonización del televisor 6, 6  
Software, V: Programas, V:  
Programas Pre-grabados  
Sonido. Efectos de sonido  
36-27
- STEP 29  
Strings (cadenas) 22  
Subrutinas 30-31  
SYMBOL SHIFT 3, 21; 19
- Teclado 18, 19  
Caracteres gráficos 26  
Ingresando información  
8, 9  
Modos 20-21  
Teclas 18-19; 18-19  
Número 19  
Operación 20-21; 20-21
- Televisor  
Codificador 42  
Conexión 5  
Tipos adecuados 4  
Sintonización 6; 6
- Tinta (ink). Color de 24-25  
Tono. Controles en el grabador  
12, 14, 15  
Tono musical 36  
TRUE VIDEO 18
- Uncommitted Logic Array  
(ULA) 42  
Unidad Procesadora Central  
V: Central Processing Unit
- Variables 22-23, 50  
Variables de sistema 48  
Voltaje. Regulador 43  
Volumen. Controles en el  
grabador 12, 14, 15
- Z 80 Microprocesador 43,  
75, 45  
ZX 16K RAM 4  
46-47

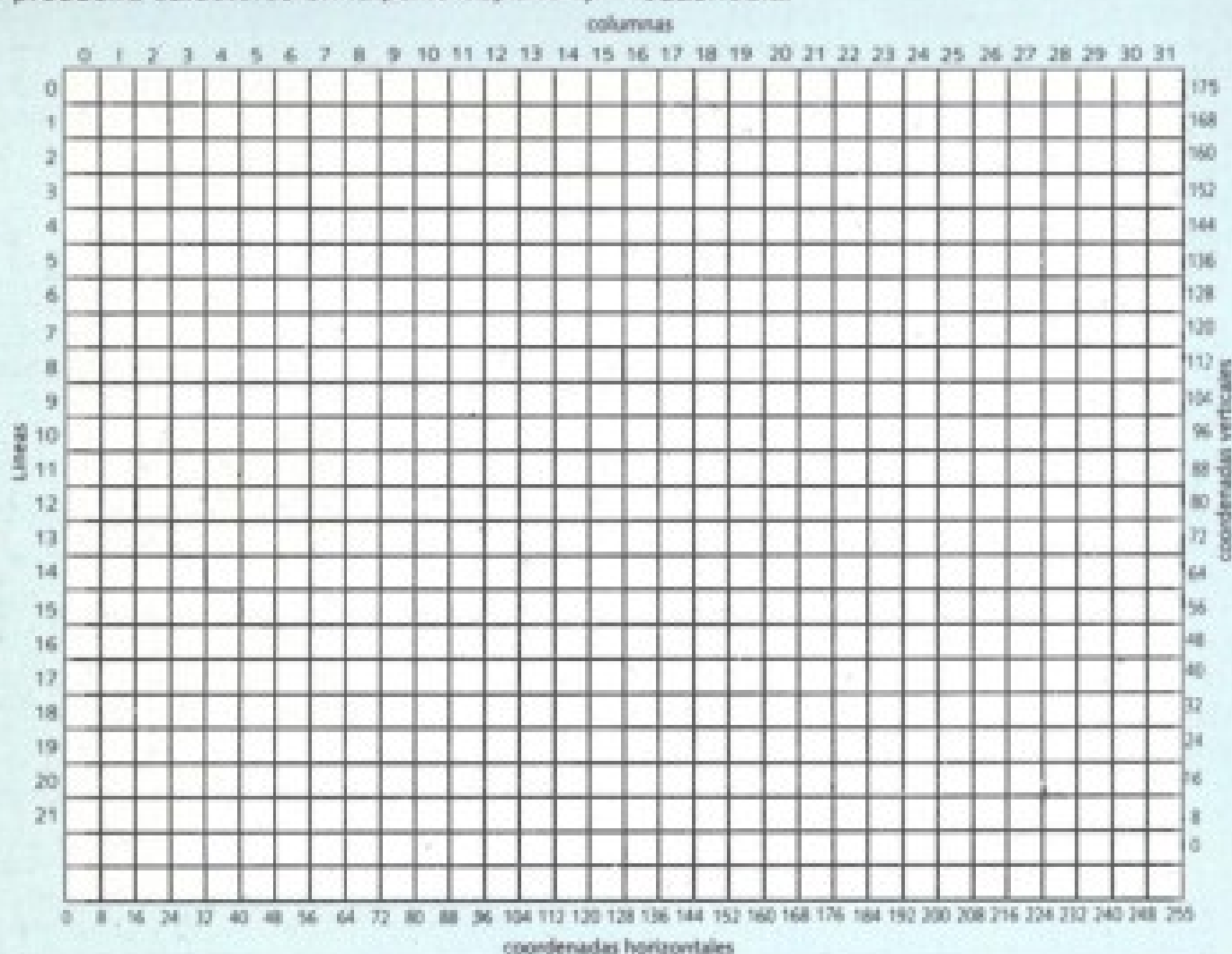
La cuadrícula a continuación muestra las coordenadas para gráficas de alta y baja resolución. Si bien las palabras claves individuales que controlan los efectos gráficos operan solamente con coordenadas de alta o baja resolución, es posible utilizar los dos sistemas al mismo tiempo produciendo una nueva presentación visual.

La cuadrícula de baja resolución se divide en dos partes principales; el área de visualización principal y dos líneas en la parte inferior de la pantalla. La palabra clave PRINT AT producirá caracteres en la parte superior y

la palabra clave INPUT AT producirá palabras en la parte inferior.

Las coordenadas para la pantalla de baja resolución se indican sobre los márgenes superior y de la izquierda de la cuadrícula.

La cuadrícula de alta resolución ocupa solamente el área de visualización principal. Las palabras claves PLOT, DRAW y CIRCLE son empleadas para producir gráficas de alta resolución. Las coordenadas de la cuadrícula de alta resolución se indican sobre los márgenes inferior y de la derecha de la cuadrícula.



**Cuadrícula para las gráficas definidas por el usuario**

Sumar los números en la tabla para obtener el valor decimal de cada línea o hilera horizontal de la cuadrícula, y a continuación utilizar la palabra clave POKE USR para programar el carácter.

[illegible]





Czerweny Electrónica S.A.I.C.  
INDUSTRIA ARGENTINA