

ABS	G
ACS	W
AND	Y
ASN	Q
AT	I
ATN	E
ATTR	L

M	MERGE	T
	MOVE	6

N	NEW	A
	NEXT	N
	NOT	S

BEEP	Z
BIN	B
BORDER	B
BRIGHT	B

O	OPEN #	4
	OR	U
	OUT	O
	OVER	N

CAT	9
CRHS	U
CIRCLE	H
CLEAR	X
CLOSE #	S
CLS	V
CODE	I
CONT	C
COPY	Z
COS	W

P	PAPER	C
	PAUSE	M
	PEEK	O
	PI	M
	PLOT	Q
	POINT	8
	POKE	O
	PRINT	P

R	RAND	T
	READ	A
	REM	E
	RESTORE	S
	RETURN	Y
	RND	T
	RUN	R

S	SAVE	S
	SCREEN \$	K
	SGN	F
	SIN	O
	SQR	H
	STOP	D
	STR \$	Y

T	TAB	P
	TAN	E
	THEN	G
	TO	F

U	USR	L
	VAL	J
	VAL \$	J
	VERIFY	R

LEN	K
LET	L
LINE	3
LIST	K
LIST	V
LN	Z
LOAD	J
L PRINT	C

Esta guía facilita la búsqueda de sentencias, funciones o símbolos, cuando aún no se tiene la suficiente destreza de manejo o cuando se produce alguna duda. Las sentencias están ordenadas por orden alfabético; para una mejor localización, a la izquierda de la sentencia aparece la inicial de ésta, y, a la derecha, la tecla donde está situada.

FUNCIONES DE CONTROL	
BREAK	SPACE
CAPS LOCK	2
DELETE	0
EDIT	1
GRAPHICS	9
INV VIDEO	4
TRUE VIDEO	3

OTRAS FUNCIONES	
CAPS SHIFT	IZDA Z
ENTER	DCHA L
SYMBOL SHIFT	DCHA M

DESPLAZAMIENTOS DEL CURSOR	
↑	7
↓	6
⇒	8
⇐	5

GRAFICOS	
■	1
■	2
■	3
■	4
■	5
■	6
■	7
■	8

COLORES	
BLACK	Ø
BLUE	1
CYAN	5
GREEN	4
MAGENTA	3
RED	2
WHITE	7
YELLOW	6

SIMBOLOS	
@	2
©	P
#	3
\$	4
%	X
&	6
'	A
()	8
! ,	9
-	Y
_	U
{ }	F
[]	G
^	D
~	V
¡ ¨	S
! ¨	O
? ¨	T
· ¨	C
· ¨	M
· ¨	N
· ¨	O
· ¨	Z
· ¨	P
· ¨	7
· ¨	K
· ¨	J
· ¨	B
· ¨	L
· ¨	H
· ¨	5
· ¨	T
· ¨	R
· ¨	W
· ¨	E
· ¨	Q

(Doblar por aquí)

MICROHOBBY

SEMANAL

BASIC

SINCLAIR

Rafael PRADES

Con la aparición del primer número de MICROHOBBY comienza también una serie de capítulos dedicados al estudio del sistema BASIC como lenguaje de programación del ordenador Spectrum. El estudio, que será bastante completo, no se limitará sólo a definir las funciones de los comandos o sentencias, tratará al mismo tiempo de explicar todos los posibles argumentos asociados a éstos. Asimismo, cada tema irá acompañado de una larga y clara lista de ejemplos. El curso está pensado tanto para quienes se han comprado un ordenador Spectrum para pasar el rato y matar maripositos, como para aquellas personas que, teniendo cierta experiencia, deseen ampliar sus conocimientos sobre el tema. Una vez terminado el curso, presentaremos una serie de programitas que serán explicados, analizados y convenientemente acompañados de diagramas de bloques y flujos.

EDITA
HOBBY PRESS, S.A.

BASIC SINCLAIR

Dos personas que quieran comunicarse necesitan un medio de expresión; éste puede ser el lenguaje hablado, escrito, la mímica, dibujos, etc...; de la misma manera, nosotros para poder comunicarnos con nuestro ordenador necesitamos conocer un lenguaje que él entienda. El utilizado por el Spectrum se denomina Lengua de Programación BASIC (iniciales en inglés de Beginner's All purpose Symbolic Instruction Code, que traducido al español significa «Código de instrucciones simbólicas de uso general para principiantes»).

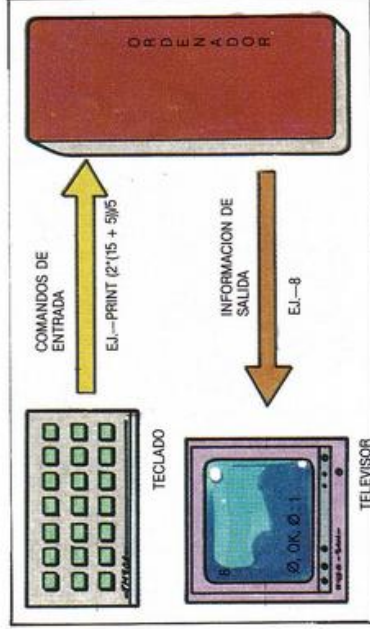
El lenguaje del Spectrum

El BASIC es un lenguaje utilizado también por otros ordenadores personales, aunque cada cual tiene ciertas particularidades que lo diferencian de los demás. En esta serie de artículos sólo vamos a tratar el lenguaje BASIC que el Sinclair Spectrum entiende.

Ya sabemos cuál es el medio de expresión que debemos utilizar con nuestro ordenador, pero ¿cómo se emplea? La respuesta es sencilla: el BASIC consta de una serie de comandos o sentencias que el programador introduce en el ordenador mediante el manejo del teclado. El aparato nos devuelve la información pedida en la pantalla del televisor. Por ejemplo, si quisiéramos que el ordenador nos calculara la operación:

$$2 \times (15 + 5) = 5$$

2 MICROBASIC



El ordenador, una vez recibido el comando que se ha introducido mediante el teclado, realiza las operaciones necesarias y visualiza el resultado en la pantalla del televisor.

y que nos devolviera el resultado, introduciríamos en el teclado el comando:

$$\text{PRINT } 2 * (15 + 5) / 5$$

El ordenador, una vez recibido el comando, calcula la expresión aritmética y posteriormente visualiza en la pantalla del televisor el resultado.

Como decíamos más arriba, el lenguaje BASIC consta de una serie de comandos o sentencias con las que se establece un diálogo o comunicación con el ordenador, pero ¿qué son las sentencias? Los comandos o sentencias son palabras clave derivadas del idioma inglés (PRINT, RUN, STOP...); estos términos, al ser introducidos en el ordenador mediante el teclado, indican a éste qué función debe de realizar.

Normalmente, las sentencias por sí solas no podrían realizar sus funciones si no

nitor la sentencia INT; pero si, estando en modo E, se accionan simultáneamente las teclas SYMBOL SHIFT o CAPS SHIFT y R, aparecerá en la pantalla el comando VERIFY. En ambos casos se observará que después de visualizarse la sentencia correspondiente (INT o VERIFY), el cursor cambia a modo L o C.

Si, por error, al introducir una sentencia que no corresponde al modo E se ha pasado a éste, para volver al modo anterior bastaría con repetir la secuencia CAPS SHIFT + SYMBOL SHIFT.

Las teclas de la fila superior tienen un tratamiento distinto, ya que contienen los gráficos predefinidos en el Spectrum y una serie de funciones de control...

Las funciones de control se encuentran fuera de las teclas pintadas de blanco y en la parte superior. Estas funciones, como por ejemplo EDIT, DELETE, desplazamientos del cursor, etc..., ya irán siendo tratadas en los siguientes párrafos; ahora lo más importante es conocer como acceder a ellas; para ello basta con pulsar simultáneamente las teclas CAPS SHIFT y la tecla correspondiente a la función deseada; es decir, si quisiéramos acceder a la función EDIT bastaría con pulsar, a la vez las teclas CAPS SHIFT y la correspondiente al número 1.

MODO G — Para acceder a los gráficos situados en las teclas con los números 1 a 8 y a los definidos por el usuario, situados en las teclas con las letras de la A a la U, hay que pasar al modo G. Para

acceder a G es necesario hacer uso de la función GRAPHICS, que como ya conocemos se accede oprimiendo simultáneamente la tecla CAPS SHIFT y la que lleva el número 9. Una vez que la operación anterior ha sido realizada, el cursor parpadeante pasa a modo G.

Los símbolos predefinidos están representados en la esquina superior derecha de las teclas 1 a 8; para visualizar cualquiera de estos símbolos en la pantalla sólo es necesario pulsar la tecla correspondiente al gráfico elegido. También se pueden visualizar estos gráficos en negativo; es decir, que las partes claras se conviertan en oscuras y las oscuras en claras; para ello basta con accionar, a la vez que se selecciona un gráfico, la tecla CAPS SHIFT o SYMBOL SHIFT.

Para visualizar los gráficos definidos por el usuario es necesario que estos estén ya definidos con anterioridad; si algún lector no sabe todavía cómo definir sus propios gráficos, no se preocupe; en otro artículo de este curso se tratará este tema, y en otras secciones de «Microhobby» también se comentará. Si los gráficos ya estuvieran definidos, bastaría para visualizarlos con pulsar la tecla correspondiente a la que se le asignó; es decir, si definimos un gráfico y lo asignamos a la tecla con la letra R, para visualizarlo habría que pasar a modo G y pulsar simplemente la tecla R.

Aquellas letras entre la A y la U que no tengan un gráfico asignado, al pasar a modo G y seleccionarlas, se visualizaría en la pantalla la letra correspondiente a la letra pulsada.

da, escrita en mayúsculas, aunque el modo elegido anteriormente fuera el L.

Para poder volver al modo anterior basta con pulsar la tecla GRAPHICS, situada en el número 9, y el cursor volverá a parpadear en el modo anterior. Desde el modo G también se puede pasar directamente al modo E, pulsando simultáneamente CAPS SHIFT y SYMBOL SHIFT.

Para acceder a los signos en rojo, situados en la esquina inferior derecha, basta con pulsar simultáneamente SYMBOL SHIFT y la tecla correspondiente.

A los números del 0 al 9 se puede acceder tanto en modo K como en L o C.

Las sentencias pintadas en rojo y situadas en la parte inferior de las teclas no tienen el mismo tratamiento que sus homólogos de las restantes filas de teclas, ya que para acceder a éstas sólo es posible hacerlo cuando estemos en modo E, pulsando SYMBOL SHIFT más la tecla correspondiente; sin embargo, en las restantes filas de teclas podría ser o bien con CAPS SHIFT o con SYMBOL SHIFT.

Las funciones de color, situadas en la parte exterior de las teclas 0 a 7, indican la correspondencia que hay entre colores y números, ya que cuando en algún comando hay que indicar una función de color, ésta será dada por su número; es decir, si en una sentencia de visualización de pantalla hay que indicar que ésta sea en color rojo (RED), habrá que seleccionar la tecla con el número 2.

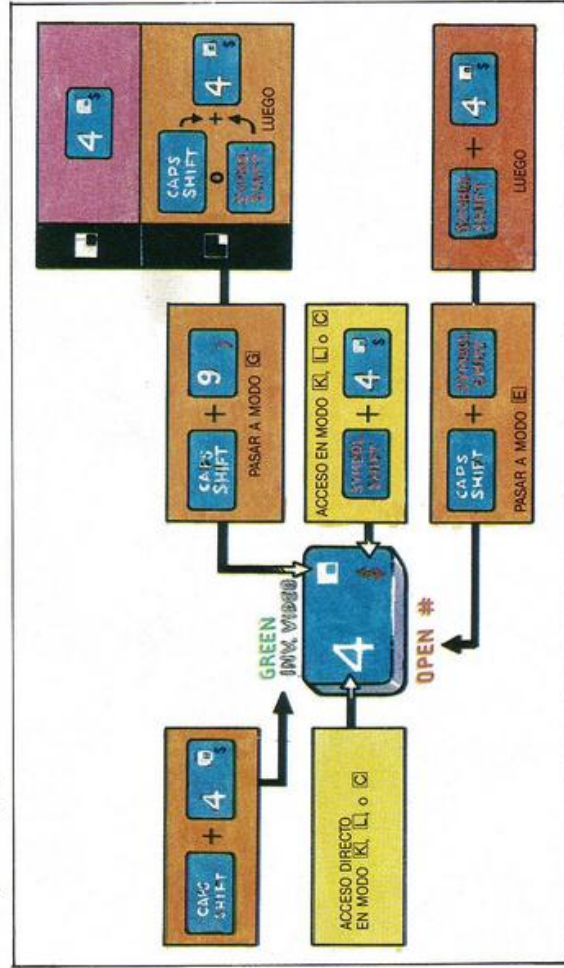
MICROBASIC 7

das en la parte exterior de la tecla, en color verde las de la parte superior y en color rojo las de la inferior, es necesario pasar al modo **E**.

der a este modo basta con pulsar simultáneamente CAPS SHIFT y CAPS LOCK, situado en la tecla correspondiente al número 2. A partir de este momento, el modo **C** sustituye al modo **L**. Presione la tecla **R** y comprobará que la **R** mayúscula aparecerá en lugar de la **r** minúscula.

MODO E.—Para poder acceder a las sentencias situa-

En el ejemplo anterior, si se pasa a modo **E** y se pulsa la tecla R, se visualizaría en la pantalla de su televisor o mo-



Las teclas de la fila superior tienen un tratamiento distinto, ya que contienen los gráficos predefinidos y una serie de funciones de control.

b) El argumento es una variable.
Función: Leer un dato de una tabla y asignarlo a la variable «C».

SENTENCIA	ARGUMENTO
READ	C

c) El argumento es una expresión matemática.
Función: asignar a la variable «X» el resultado de la operación:
 $(15 \times 2) + (8/3)$

SENTENCIA	ARGUMENTO
LET	$x = ((15 * 2) + (8/3)) / 5$

d) Y, por último, el argumento es una cadena de caracteres.

Función: visualizar la siguiente cadena alfanumérica «Revista Microhobby».

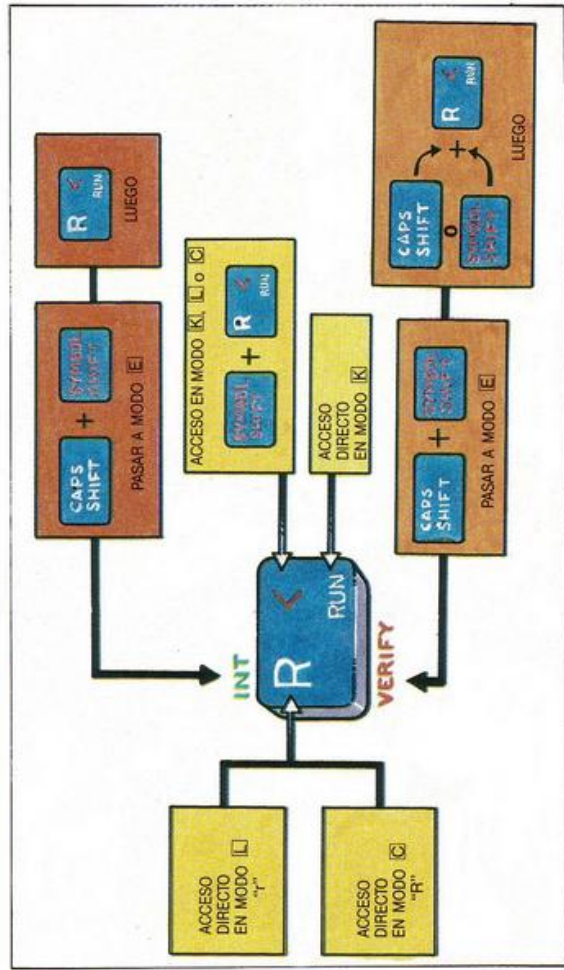
SENTENCIA	ARGUMENTO
PRINT	«Revista Microhobby»

No se preocupe si aún no entiende qué es una subrutina, qué es una variable o cómo se introducen estas sentencias, ya que esto se irá viendo en capítulos siguientes.

Una de las características del Spectrum consiste en que no es necesario teclear los comandos o sentencias letra a

letra, sino que éstos ya están definidos y asignados a una tecla. Por ejemplo: para acceder a la sentencia PRINT no se necesita pulsar las teclas correspondientes a las letras P, R, I, N y T; con sólo accionar la «P» aparece escrita en la pantalla del televisor el comando PRINT.

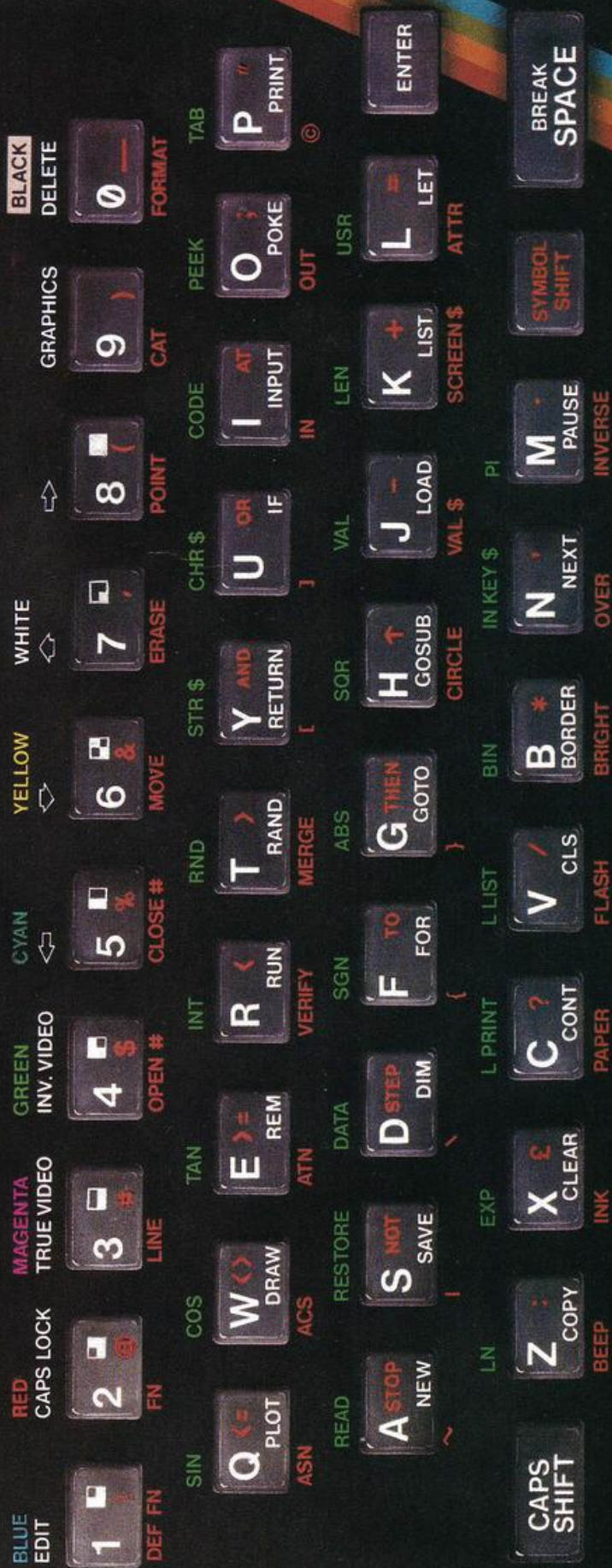
El Spectrum tiene 40 teclas; con éstas sería imposible abarcar los 88 diferentes comandos que existen, así como las diferentes funciones de control, los distintos gráficos



En el Spectrum, las teclas son multifunción. Cada una de ellas tiene diversas utilidades, por lo que es imprescindible un correcto manejo de los «modos».



ZX Spectrum



terior del Spectrum estaba pintada en verde, se pulsa la tecla «EXTEND MODE» y a continuación, cuando el cursor de modo cambia a **E**, se pulsa la tecla seleccionada.

Para acceder a la sentencia inmediatamente inferior, pintada de rojo en el otro modelo, se pulsa «EXTEND MODE» y seguidamente, junto con la tecla seleccionada, cualquiera de las teclas «SYMBOL SHIFT». Cuando termina de visualizarse una sentencia en modo extendido, el cursor parpadeante cambia al modo anterior, **L** o **C**.

Modo G

Para pasar al modo **G** (Graphics Mode), el «Spectrum +» tiene la tecla «GRAPH». Pulsando esta tecla se tiene acceso a los semigráficos situados en las teclas con los números «1» a «8» y a los gráficos definidos por usuario, en las teclas con las letras de la «A» a la «V». Para retornar al modo anterior, es necesario pulsar de nuevo la tecla «GRAPH».

Edición de programas

Para la edición de programas y corrección de errores, el «Spectrum +» tiene teclas independientes con la misma funcionalidad que en el modelo anterior. Estas teclas son «EDIT», «DELETE» y los controles de cursor (arriba, abajo, izquierda y derecha), situados en este modelo a los lados de la barra espaciadora.

Con esta disposición se hace más agradable y rápida la edición y corrección de programas, ya que con sólo pulsar una tecla se consigue la función deseada. ■

Edición de programas

En el momento de introducir una instrucción en el ordenador se nos presentan dos opciones: que se ejecute nada más ser introducida o que quede almacenada en la memoria del ordenador para su posterior ejecución.

En la primera opción, una vez ejecutada la instrucción, si se desea repetir su ejecución es necesario teclear de nuevo; en el segundo caso no es necesario, ya que la primera vez que se introdujo quedó almacenada en la memoria del ordenador y podemos ejecutarla tantas veces como queramos, siempre y cuando no desconectemos la clavija de alimentación de 9V DC.

Para introducir una instrucción de ejecución inmediata, sólo es necesario teclearla y pulsar la tecla ENTER, que indica al ordenador que la instrucción se ha terminado de teclear; en ese momento el ordenador la analiza para que no tenga ningún error de sintaxis; es decir, que el argumento esté correctamente teclado y que esté en concordia con el tipo de sentencia. Si todo ha sido correcto, la instrucción se ejecuta inmediatamente.

Para introducir una instrucción que no se ejecute inmediatamente, es necesario asignarle un número de línea comprendido entre 1 y 9999; este número se tecldea estando el cursor en modo **K**: a continuación se tecldea la instrucción y posteriormente, como en el caso anterior, se pulsa la tecla ENTER. Se analizan los posibles errores que la instrucción podría contener y si todo es correcto, no se ejecuta, sino que pasa a memoria; esto se puede comprobar, ya que

se visualiza en la parte superior de la pantalla.

Para poder ejecutar esa instrucción, es necesario ejecutar con anterioridad la sentencia inmediata RUN; para ello basta con pulsar las teclas R y ENTER. Como hemos dicho anteriormente, esa instrucción se puede repetir, tantas veces como queramos, simplemente introduciendo la sentencia RUN y pulsando ENTER.

Veamos unos ejemplos; pruebe a introducir la siguiente instrucción inmediata:

```
PRINT "Curso BASIC/Spectrum"
```

al terminar de teclear y pulsar ENTER se ejecutará la instrucción y aparecerá en la parte superior de la pantalla, la cadena alfanumérica: Curso BASIC/Spectrum, y en la parte inferior un mensaje que envía el ordenador, indicando que la instrucción ha sido ejecutada correctamente, «OK, 1». Si quiere repetir la ejecución, es necesario repetir también la instrucción.

Para que ésta no se ejecute hasta que usted quiera, debe de asignarle un número de línea, por ejemplo el 100, por tanto teclee:

```
100 PRINT "Curso BASIC/Spectrum"
```

al pulsar ENTER, comprobará que la instrucción tecldeada pasa a la parte superior de la pantalla; también observará que después del número de línea 100, el ordenador ha colocado el símbolo >; éste es un indicador de presencia, conocido también como «prompt» que indica cuál es la última línea editada.

Para poder ejecutar esa instrucción, deberá introducir

RUN, y como siempre pulsar ENTER. El listado de la instrucción se borrará, y ésta se ejecutará de la misma manera que si hubiese sido introducida directamente. Aunque el listado haya desaparecido de la pantalla, no se preocupe: está almacenado en la memoria del ordenador para poder visualizarlo de nuevo, simplemente pulse la tecla ENTER. Podrá volver a ejecutar la instrucción, sin tener que teclear la otra vez, introduciendo la sentencia RUN y pulsando ENTER.

El numerar una sola instrucción, en principio, no tiene mucha utilidad, lo que sí tiene y bastante, es numerar una cantidad más o menos larga de instrucciones; esto es lo que se conoce como programa. Un programa es, por tanto, una secuencia de instrucciones ordenadas que realizan una función determinada.

Un ejemplo de estructura de un programa puede ser el siguiente:

100	instrucción	n.º 1
200	instrucción	n.º 2
300	instrucción	n.º 3
1000	instrucción	n.º 10

En el lenguaje BASIC, las instrucciones se suelen numerar de diez en diez; esto se debe a que muy pocas veces un programa funciona a la primera; siempre es necesario modificar, borrar o añadir alguna instrucción que se nos quedó olvidada en el tintero. Si las tenemos numeradas a intervalos de diez, siempre tenemos la posibilidad de insertar una instrucción, asignándole un número de línea intermedio, ya que el ordenador, cuando eje-

cuta el programa que tiene en memoria, siempre lo hace empujando por la instrucción con el número de línea más bajo y continúa en orden creciente.

En el ejemplo anterior, si queremos insertar una instrucción entre las líneas 10 y 20, le asignaríamos, por ejemplo, el número 15, de manera que el programa quedaría de la siguiente forma:

10	Instrucción	n.º 1
15	Instrucción	n.º 2
20	Instrucción	n.º 3
30	Instrucción	n.º 4
100	Instrucción	n.º 11

En el Spectrum, los números de línea tienen que estar comprendidos entre el 1 y el 9999; cualquier instrucción que se asigne con una numeración distinta, será rechazada por el ordenador. En el caso de que se asigne a una instrucción un número de línea igual a 0, la pantalla se limpiará de caracteres, y en la zona reservada para los informes del ordenador, aparecerá el mensaje: C Nonsense in BASIC, 0: 1. Para volver a recuperar el listado, si lo hubiere, pulse ENTER. Si el número de línea fuera superior a 9999 al pulsar ENTER aparecerá una ? parpadeante a la derecha del número de línea; en el capítulo denominado "Corrección de errores", se dan orientaciones necesarias para poder corregir este fallo.

Se habrá dado cuenta que, según se van introduciendo las instrucciones, el prompt > se va desplazando, y siempre apunta a la última línea editada.

Dentro de una misma línea se pueden introducir varias

10 MICROBASIC

do su localización. Las demás teclas, a excepción del SPACE, mantienen su situación y funcionamiento anterior. La tecla «SPACE» ha sido sustituida, al igual que en otros ordenadores, por una **barra espaciadora** semejante a la que incorporan las máquinas de escribir.

Otra de las diferencias del teclado, es la duplicidad de las teclas «CAPS SHIFT» y «SYMBOL SHIFT», esto proporciona un mejor manejo de estas funciones, ya que al estar situadas a ambos lados del teclado podrán ser utilizadas con cualquiera de las dos manos.

Modos L Y C

Después de haber pulsado una tecla, el modo que presenta el «Spectrum +» por defecto es el **L** (Letter Mode), de este modo al pulsar una tecla, la letra inscrita en su interior se visualiza en minúscula; si desea que aparezca en mayúscula sin tener que cambiar de modo, es necesario pulsar simultáneamente junto con la letra elegida, cualquiera de las teclas «CAPS SHIFT». Cuando el texto a escribir en mayúscula es largo, conviene pasar al modo **C** (Capitals Mode). En el «Spectrum +» para acceder a este modo, basta simplemente con pulsar la tecla «CAPS LOCK»; a partir de ese momento, el modo **C** sustituye al **L** hasta que se pulse de nuevo esta tecla.

Modo E

El «Spectrum +» tiene una tecla específica para pasar al modo extendido **E**; ésta se denomina «EXTEND MODE». Para seleccionar la sentencia situada en la parte superior de la tecla que en el modelo an-



Ejemplo de edición de un programa.

```

10 REM *****
   * CURSO BASIC *
   * EJER: 1 *
   * *****
LS 50 BORDER 1: PAPER 7: INK 2: C
90 GO SUB 1000
90 REM *****
   * REJILLA *
   * *****
100 FOR n=8 TO 248 STEP 8
110 PLOT n,175
120 DRAW 0,-175
130 BEEP 0.05,n/8
140 NEXT n
150 FOR n=167 TO 7 STEP -8
160 PLOT 0,n
170 DRAW 255,0
180 BEEP 0.05,n/8
190 NEXT n
200 REM

```

```

*****
* MENSAJE HORIZONTAL *
*****
410 RESTORE 470
420 FOR n=1 TO 48
430 READ y: READ x
440 PRINT AT y,x;CHR$ 20+CHR$ 1
CHR$ 32
450 BEEP 0.05,x
460 NEXT n
470 DATA 2,3,2,4,2,15,2,16,2,17
,2,22,2,23,2,28,2,29
480 DATA 5,15,5,16,5,17,5,22,5,
23
490 DATA 8,3,8,4,8,9,8,10,8,22,
8,23,8,28,8,29
500 DATA 11,4,11,5,11,10,11,11,
11,16,11,17,11,21,11,22,11,23,11
,27,11,28
510 DATA 14,4,14,5,14,10,14,11,
14,15,14,17
520 DATA 17,4,17,5,17,16,17,17,
17,21,17,22,17,23,17,27,17,28
500 REM

```

```

*****
* TRAGON *
* *****

```

```

610 FOR y=0 TO 21
620 FOR x=0 TO 31
640 IF (x AND y)=0 THEN GO TO 6
60
650 IF SCREEN$ (y,x)=CHR$ 32 TH
EN GO TO 670
660 PRINT AT y,x;CHR$ 16+CHR$ 3
CHR$ 144: BEEP 0.01,20: PRINT A
T y,x;CHR$ 17+CHR$ 4;CHR$ 32: GO
TO 680
670 PRINT AT y,x;CHR$ 16+CHR$ 3
CHR$ 144: BEEP 0.02,-15: PRINT
AT y,x;CHR$ 20+CHR$ 1;CHR$ 16+CH
R$ 2;CHR$ 32
680 NEXT x
690 NEXT y
700 REM

```

```

*****
* LOGO *
* *****

```

```

710 RESTORE 770
720 FOR x=7 TO 24
730 READ dato

```


Ejercicio

Como comprobación de que ha entendido lo explicado hasta este momento, intente editar el programa que a continuación le proponemos; que combine los semigráficos con sonido y color.

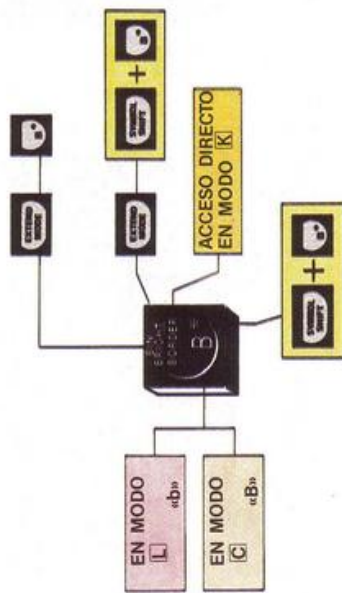
No se preocupe si no entiende el significado de las sentencias o la filosofía del programa; ya que el objeto de éste es practicar el acceso al teclado, la edición y la corrección de los errores que pudieran surgir.

El programa que proponemos es un poco enigmático, hasta que no lo ejecute no sabrá el contenido del mensaje que aparece en pantalla. Aquellos lectores que entiendan algo de BASIC, y les guste lo misterioso, pueden intentar descifrar este mensaje, ya que la clave está en las sentencias DATA.

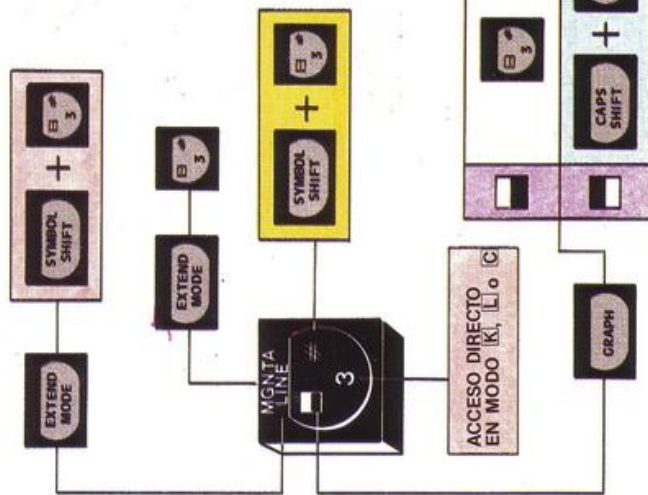
Es fácil de teclear, aunque hay que tener cuidado al pulsar los argumentos de las sentencias DATA, ya que si algún dato estuviese cambiado o faltado, el mensaje se vería desvirtuado. Cuando termine de teclear las 78 líneas de que se compone el programa, podrá ejecutarlo, pulsando RUN y ENTER.

Si todo es correcto, cuando termine de ejecutarse el programa, el ordenador enviará el siguiente mensaje:

9 STOP statement, 999 : 1
si el ordenador enviara otro distinto, revise las instrucciones o los números de línea, ya que seguramente habrá cometido algún error. Si tiene dudas, no sienta reparo en volver a repasar los capítulos anteriores. Esperamos que no tengan



Descripción del manejo de los modos en el nuevo teclado.



La fila superior tiene un tratamiento ligeramente distinto.

ningún tipo de dificultad y que el programa lo edite y ejecuten correctamente.

Tedado del «ZX Spectrum +»

El «ZX Spectrum +», también conocido como «Spectrum Plus», presenta algunas diferencias con respecto a su

homólogo. Aparte de su aspecto exterior, que evidentemente es distinto, se percibe que existe una diferencia en cuanto al número de teclas, ya que éste tiene 58 en lugar de 40. En la tecla adjunta se observa cuáles son las nuevas teclas incorporadas y en la figura representativa del teclado

```
*****
* MENSAJE VERTICAL *
*****
210 RESTORE 270
220 FOR n=1 TO 88
230 READ y: READ x
240 PRINT AT y,x:CHR$ 20+CHR$ 1
:CHR$ 32
250 BEEP 0.05,y
260 NEXT n
270 DATA 3,2,4,2,5,2,6,2,7,2,3,
5,7,5
280 DATA 2,8,3,8,4,8,5,8,6,8,7,
8,2,11,3,11,4,11,5,11,6,11,7,11
290 DATA 2,14,3,14,4,14,5,14,6,
14,7,14,8,14,6,16,7,17,3,18,4,18
,8,18
300 DATA 3,21,4,21,7,21,3,24,6,
24,7,24
310 DATA 3,27,4,27,5,27,6,27,7,
27,3,30,4,30,5,30,6,30,7,30
320 DATA 11,3,12,3,13,3,14,3,15
,3,16,3,17,3,12,6,13,6,15,6,16,6
330 DATA 12,9,13,9,14,9,15,9,16
,9,17,9,12,12,13,12,14,12,15,12,
16,12,17,12
340 DATA 12,15,13,15,16,15,12,1
8,15,18,16,18
350 DATA 12,22,13,22,14,22,15,2
2,16,22
360 DATA 12,26,13,26,14,26,15,2
6,16,26,12,29,16,29
400 REM
```

Corrección de errores

Mientras efectúa la laboriosa tarea de edición de un programa, seguro que alguna vez se habrá olvidado de introducir alguna instrucción: un número de línea que no correspondía o bien se habrá encontrado que en el momento de insertarla en memoria una palabra, ya que aparecía una parpadeante, indicando un error de sintaxis. En este capítulo, se va a tratar de aclarar las posibles dudas que se tengan sobre borrado, modificación o inserción de nuevas líneas.

Para hablar de las posibles correcciones a efectuar en

10	Instrucción n.º 1
20	Instrucción n.º 2: 3:4
30	Instrucción n.º 5
100	Instrucción n.º 12

Un ejemplo de utilización del separador, podría ser el siguiente:

nuestro programa, vamos a distinguir si las instrucciones están insertadas en memoria o no, ya que su tratamiento depende de este detalle.

Para modificar una instrucción que estamos tecleando, es necesario hacer uso de varias de las funciones situadas en la fila superior de teclas; éstas son: los desplazamientos del cursor, izquierda (←) y derecha (→), y la función DELETE. Los desplazamientos del cursor no necesitan casi explicación, ya que como su propio nombre indica, desplazan el cursor parpadeante en la dirección que indica la flecha, y la función DELETE borra el carácter situado a la izquierda del cursor. Estas funciones se seleccionan apretando la tecla CAPS SHIFT simultáneamente con la de función correspondiente; al igual que ocurre con cualquier otra tecla, si se mantienen apretadas ambas durante cierto tiempo, la función se repite. En el modo G, para seleccionar la función DELETE no se necesita pulsar CAPS SHIFT; simplemente apretando la tecla con el número 0 se borra el carácter anterior al cursor.

Una vez explicadas las funciones, para poder realizar una modificación, es necesario desplazar el cursor a la izquierda, hasta situarlo en el lugar que queramos; ya en él podemos, o bien insertar algún carácter que se nos olvidó o bien hacer uso de la función DELETE. Si quisiéramos seguir escribiendo al final de la línea será necesario ir desplazando el cursor hacia la derecha, hasta llegar a él.

En caso de que apareciera una 2 parpadeante al inten-

tar insertar una instrucción, la secuencia de operación sería la misma; la interrogación nos facilita la búsqueda del error, ya que se coloca al lado de él.

Para borrar una línea completa que estemos tecleando podríamos, haciendo uso de la función DELETE, ir borrando carácter por carácter, pero cuando la línea es larga resulta más cómodo operar de otra manera: se accede a la función EDIT; para ello basta accionar simultáneamente la tecla CAPS SHIFT y la corres-

pondiente al número 1; en ese momento nos desaparece la línea que queríamos borrar, y en su lugar aparece una copia de la última línea editada; es decir, la que en el listado contenía el prompt >. Para reestablecer esta línea a la memoria, basta con pulsar ENTER.

Una vez que las sentencias están en memoria, para insertar una nueva simplemente basta, como ya dijimos en el capítulo "Edición de programas", asignarle una numeración intermedia; es decir, si queremos insertar dos nuevas

instrucciones entre las líneas 110 y 120, podríamos asignarles los números de línea 113 y 117; de esta manera, el ordenador las ejecutará después de la 110 y antes de la 120.

Para borrar una instrucción, basta con teclear su número de línea que tenía antes; este método es cómodo cuando las instrucciones son pequeñas, pero cuando, por el contrario, son largas, es más cómodo hacer uso de la función EDIT. Cuando se hace uso de esta función, una copia de la línea

que contenga el prompt > aparece en la zona inferior de la pantalla; para poder desplazar el prompt hasta la instrucción que deseamos modificar, es necesario hacer uso de las funciones de desplazamiento del cursor, arriba (↑) y abajo (↓), situadas en las teclas con los números 7 y 6 respectivamente; estas funciones, al igual que EDIT, se realizan con ayuda de la tecla CAPS SHIFT.

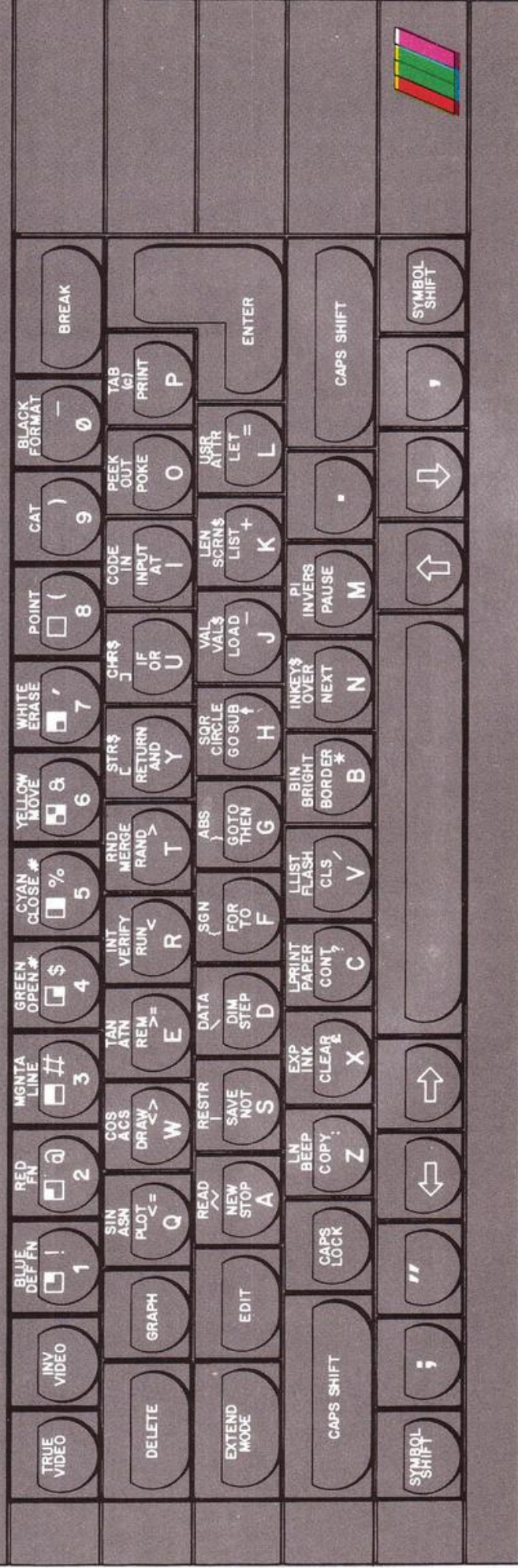
Una vez que la línea a modificar está en la parte inferior, ésta se rectifica de la misma manera que cuando todavía

no está en memoria, es decir, con los desplazamientos de cursor derecha e izquierda y con la función DELETE; cuando la línea está corregida se pulsa ENTER. Si se usa la función EDIT para modificar un número de línea, al pulsar ENTER, no desaparece la instrucción con la numeración antigua, sino que se mantienen ambas en memoria, como se puede constatar por la información que se visualiza en pantalla. Esto puede servir para copiar una instrucción tantas veces como queramos,

Existen otro método para modificar una línea sin tener que estar desplazando el prompt de línea en línea, este método lo explicaremos cuando veamos la sentencia LIST.

sinclair

ZX Spectrum+



Aspecto del teclado profesional del ZX-Spectrum +, incluyendo 18 nuevas teclas.

en la cinta, con la siguiente destrucción de los programas.

Oscilación

La mayoría de los aparatos de cassette comerciales disponen de un *circuito monitor*, que permite al usuario escuchar en el altavoz la grabación que se está efectuando desde

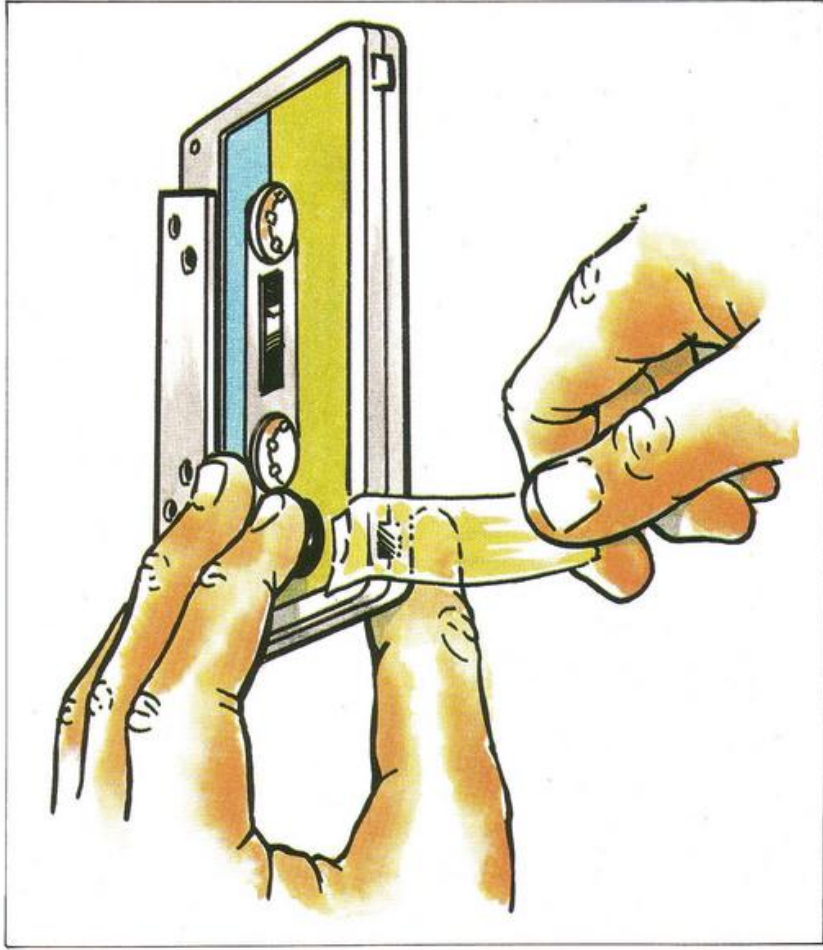
el de grabación se verá alterada, impidiendo que el programa pueda ser cargado posteriormente.

La solución consiste en abrir el bucle. Para ello desconecte cualquiera de las dos clavijas que unen la entrada EAR con la salida de auricular o altavoz externo.

té bien grabada, pueden presentarse dos situaciones:

—Que el nivel de grabación de la cinta sea distinto al de reproducción. Para subsanar esta anomalía, basta con aumentar ligeramente el volumen del cassette y proceder a cargar de nuevo.

—Que el ajuste de la cabeza



Tapado de lengüetas.

una radio, desde otro cassette o desde un plato giradiscos. Cuando desea grabar un programa y conecta los dos pares de clavijas (EAR y MIC), la señal de grabación procedente del ordenador puede retornar de nuevo a él debido al circuito monitor. Este bucle formado empezará a oscilar, y la se-

Ajuste

Si emplea una cinta grabada en un aparato de cassette distinto al que maneja, puede ocurrir que no le sea posible cargar un programa, ya que aparece el correspondiente mensaje de error.

Suponiendo que la cinta es-

ta grabadora/reproductora fuera distinto. Para reajustar su cassette, utilice un destornillador del tipo estrella y retoque la altura de la cabeza hasta conseguir que el programa se cargue sin errores. Si desea volver al ajuste original, inserte una cinta grabada en ese cassette y retoque de nuevo.

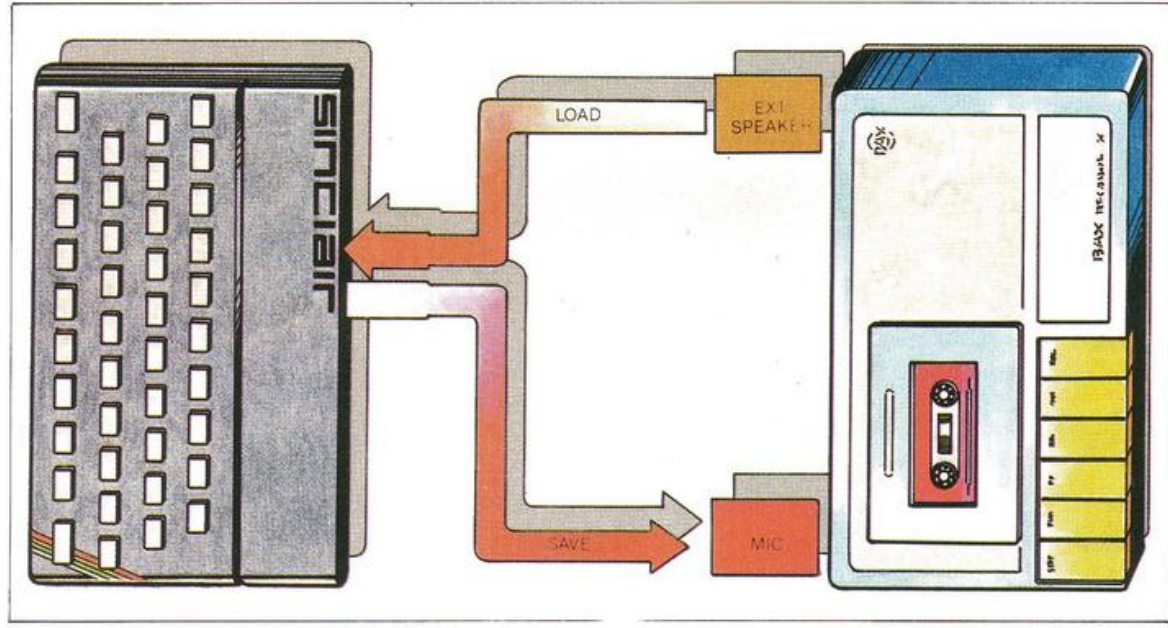
ALMACENAMIENTO DE PROGRAMAS

Mientras no desconecte el ordenador, podrá ejecutar cuantas veces quiera el programa almacenado en memoria, por ejemplo, el propuesto en el capítulo anterior. Si desea volver a ejecutarlo en alguna otra ocasión, no parece lógico volver a teclearlo o tener el ordenador enchufado para que el contenido de la memoria no se pierda; por este motivo, los ordenadores personales tienen algún sistema de almacenamiento de programas. El Spectrum tiene la posibilidad de hacerlo en cintas de *cassette* comerciales, en cartuchos para *Microdrive* o conocidos como *Diskettes*; en esta ocasión sólo vamos a tratar el almacenamiento en cintas de cassette.

Una vez que tengamos editado el programa, y sepamos con certeza que éste se ejecuta correctamente, pasaremos a grabarlo. Antes, es necesario conectar el cassette según se indica en el capítulo 6 del Manual de Instrucciones del Spectrum.

Para efectuar la grabación es necesario hacer uso del comando **SAVE**, que tiene como argumento el nombre que deseemos poner al programa. Debe ir entre comillas y no superar la cantidad de diez caracteres, éstos pueden ser letras, números o símbolos.

Pongamos un ejemplo, que queramos grabar el programa del ejercicio anterior y que de-



Grabación/recuperación.

seamos llamarle "EJER/1", la estructura de la instrucción sería:

SAVE "EJER/1"

para ejecutarla es necesario pulsar la tecla ENTER. Si hubiéramos asignado otro nombre al programa y nos apareciera el mensaje:

F Invalid file name, 0:1

significaría que el nombre es más largo de diez caracteres o que lo intentamos almacenar como cadena vacía; es decir:

SAVE ""

Si no aparece el mensaje de error, es porque el nombre asignado es correcto, y en su lugar aparecerá el mensaje:

Start tape, then press any key

Este mensaje indica que se ponga el aparato de cassette en posición de grabar, es decir, pulsando las teclas PLAY y RECORD; si la tecla RECORD no entrara, es que la cinta utilizada está protegida contra posibles grabaciones, por tanto, es necesario utilizar otra cinta que no lo esté.

Cuando el aparato esté en marcha, pulsar cualquier tecla del Spectrum, excepto CAPS SHIFT o SYMBOL SHIFT, y la grabación empezará a efectuarse.

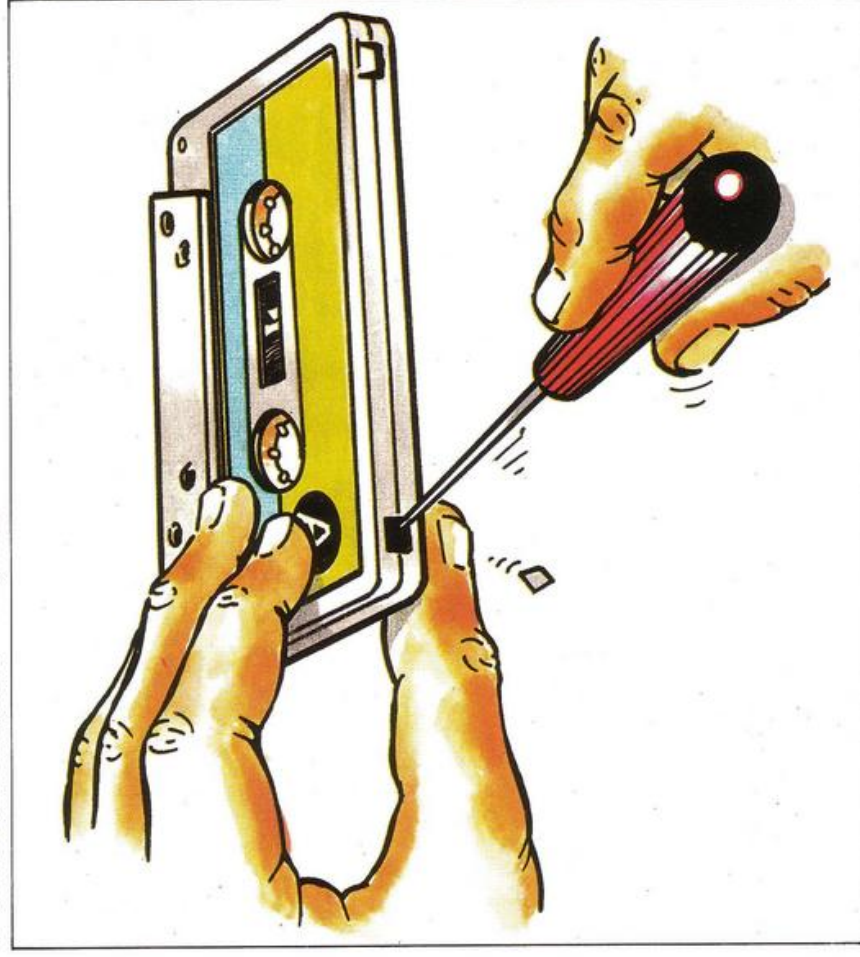
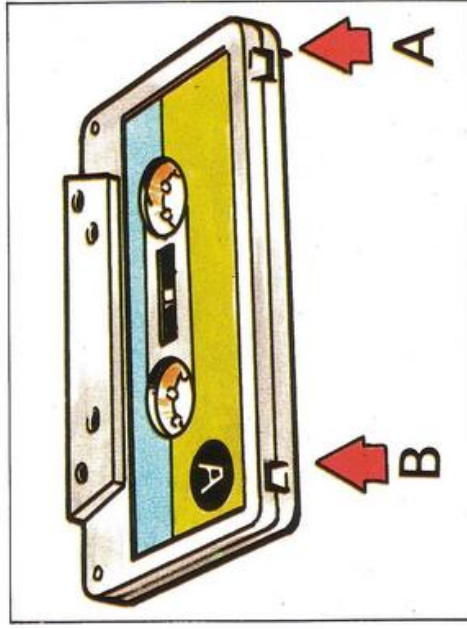
Mientras tanto, en el contorno de la pantalla se verán configuraciones de bandas coloreadas horizontales, que se van desplazando. Cuando la grabación termina, el ordenador envía el mensaje 0 OK, 0; 1; en ese instante el aparato de cassette debe pararse. Hay

Conservación de cintas

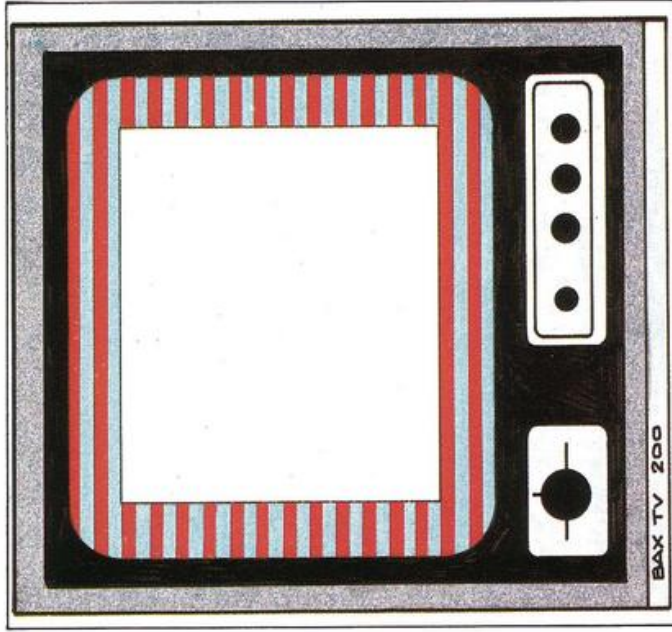
Para no deteriorar los programas grabados, es necesario tener un cuidado especial con las cintas. Para una mejor conservación de éstas, conviene tener en cuenta los siguientes consejos:

El colocar una cinta en las proximidades de un televisor, puede traer graves consecuencias, ya que éstos, internamente, llevan incorporados transformadores y bobinas de alta tensión que generan campos magnéticos. Si son lo suficientemente potentes pueden alterar la información contenida

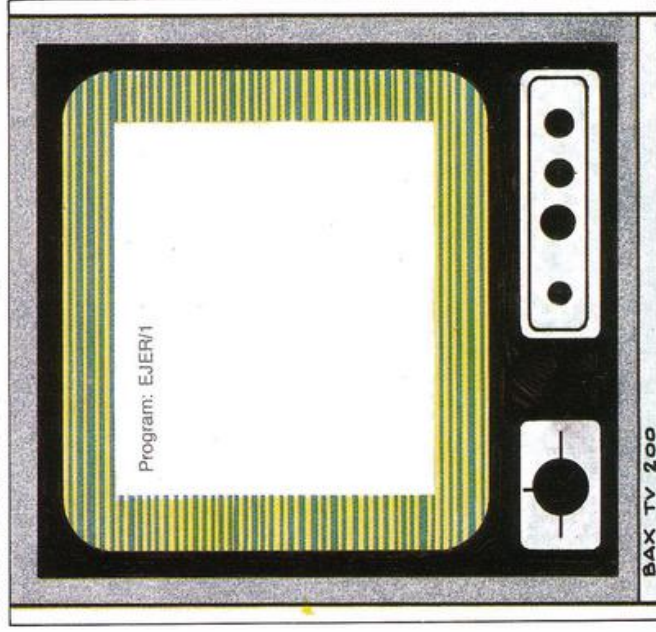
Lenguetas.



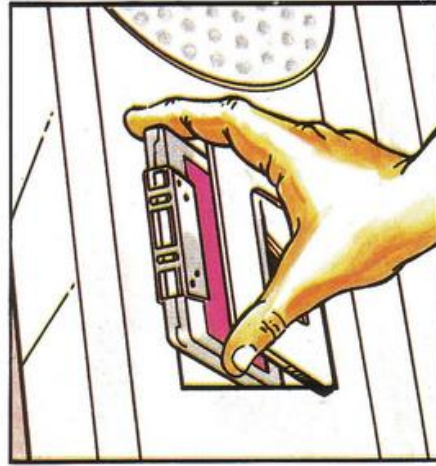
Protección de programas.



Señal de preajuste del nivel de grabación.



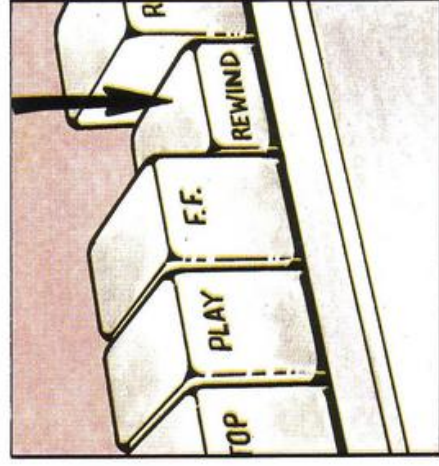
Señal de "grabación/recuperación".



Inserte con cuidado.

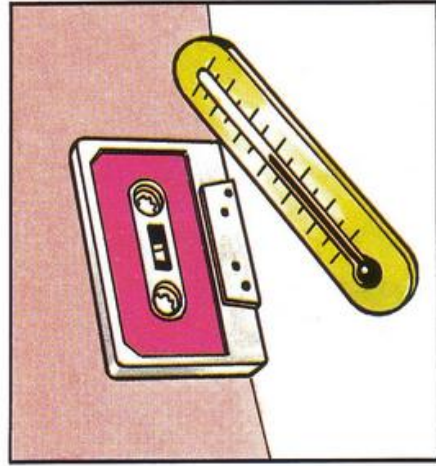


Evite campos magnéticos.

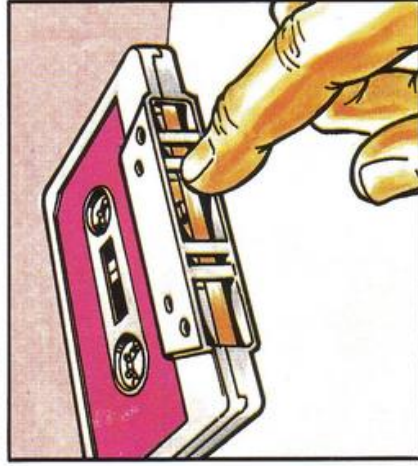


Rebobine al final.

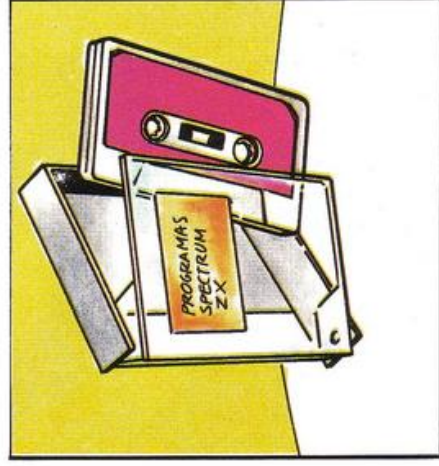
22 MICROBASIC



Conserve a 10° C. - 52° C./48%-80% Hr.



No tocar.



Conserve la cinta en su estuche.

que poner especial atención en no empezar a grabar en la zona transparente del comienzo de la cinta, ya que en esa zona no se puede grabar.

Verificación

Antes de realizar cualquier otra tarea, es necesario cerciorarse de que el programa ha sido correctamente grabado en la cinta; para ello, se utiliza la sentencia VERIFY. Esta comando lo que hay grabado en la cinta con el contenido de la memoria.

Para verificar el programa, es necesario rebobinar la cinta de cassette hasta un punto anterior al comienzo de la grabación; para este fin es útil contar con un aparato de cassette que disponga de contador.

Utilizando el ejemplo ante-

rior, la estructura de la instrucción sería la siguiente:

VERIFY "EJER1"

Al pulsar ENTER, el contenido de la pantalla cambiará de color alternativamente. A partir de este momento, ya se puede poner en marcha el cassette pulsando la tecla PLAY. Cuando la cabecera del programa, es decir el principio, sea encontrado, aparecerá en la pantalla el siguiente mensaje:

Program: EJER1

y a continuación se verificará la grabación.

En el contorno de la pantalla se visualizarán las típicas configuraciones de bandas horizontales mencionadas anteriormente. Cuando la verificación ha sido efectuada, el

ordenador nos enviará, si la grabación es correcta, el mensaje "OK, 0.1".

Si ésta no fuera correcta, pueden ocurrir dos cosas; primera, que la cabecera del programa no sea detectada, y por consiguiente, el ordenador se quedará en un bucle sin fin esperando encontrarla. Lo segundo que puede ocurrir, es que la cabecera sea detectada, pero que algún dato del programa grabado no coincida con el contenido de la memoria. En este caso el ordenador nos enviará el mensaje:

R Tape loading error, 0.1

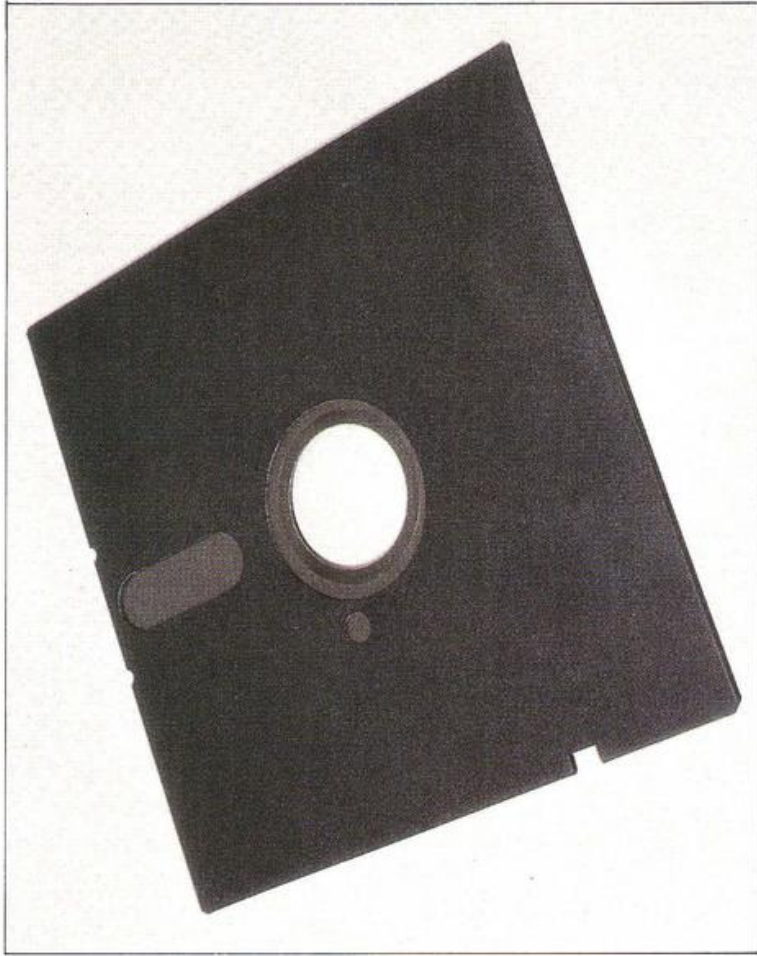
En ambos casos, volver a repetir la secuencia de verificación, y si el error persiste, grabar la cinta de nuevo.

Para salir del bucle sin fin del primer caso, es necesario hacer uso de la función



Almacenamiento en cinta o cassette.

MICROBASIC 19



Almacenamiento en Diskette.

BREAK, situada en la tecla espaciadora (SPACE); mantener esta tecla pulsada hasta que aparezca el mensaje:

D BREAK-CONT repeats, Ø : 1

Una vez que esté grabado y verificado el programa, éste quedará almacenado *permanente* en la cinta, siempre y cuando no hagamos otra grabación en la misma zona y mientras la cinta sea conservada en las debidas condiciones.

Recuperación de programas

Si tenemos almacenado algún programa en cinta, pode-

mos volverlo a ejecutar cuando queramos; para ello es necesario copiar la información contenida en la cinta, en la memoria del ordenador; este proceso se conoce como *carga* o *recuperación* de programas.

El comando de carga es el denominado LOAD, éste, como es lógico, tiene por argumento el nombre del programa que deseamos recuperar. El asignar un nombre a un programa, con el comando SAVE, no fue una cosa caprichosa, sino que responde a una necesidad; cuando en una cinta tenemos grabados varios programas, uno a continuación de otro, la única forma de que el ordenador sepa cual debe cargar es

por el nombre que cada uno tiene asignado y que lo diferencia de los demás.

La estructura de la instrucción, siguiendo con el mismo ejemplo, es:

LOAD "EJER1"

Antes de poner el cassette en marcha, es necesario rebobinar la cinta hasta un punto anterior al comienzo del programa. Para empezar la carga, se necesita pulsar la tecla ENTER una vez introducido el comando y, a continuación, pulsar la tecla PLAY del aparato de cassette; a partir de este momento, tanto los mensajes de funcionamiento correcto, como los de error, son los mis-

mos que los proporcionados por la sentencia VERIFY.

Tanto con la sentencia VERIFY como con la LOAD, si la cinta se rebobinó hasta un punto lejano del comienzo del programa, al conectar el cassette irán apareciendo en pantalla los nombres de aquellos programas que el ordenador encuentre antes de llegar al especificado en el argumento de la sentencia.

Una vez que el programa está copiado en la memoria del ordenador, el aparato de cassette debe ser parado y la cinta puede retirarse para ser utilizada en otra ocasión.

Hay una variante de la sentencia LOAD, en la que el argumento es una cadena de caracteres vacía; con esta estructura el ordenador carga el primer programa que encuentre, aunque no se le especifique el nombre de éste; el formato es el siguiente:

LOAD ""

En este capítulo hemos explicado las estructuras básicas de las sentencias SAVE, VERIFY y LOAD, en otro posterior, se tratarán con más detalle.

Protección de programas

Cuando una cinta de cassette está grabada con programas definitivos, es conveniente proteger ésta contra posibles regrabaciones accidentales.

Las cintas de cassette disponen en su parte posterior de dos lengüetas, una a cada lado, que sirven para indicar si la cinta es de *lectura/escritura* o por el contrario si es de *sólo lectura*. Cuando las lengüetas están intactas, la cinta permite ser regrabada cuanto veces se quiera, por eso se dice que es de *lectura/escritura*.

ra, en cambio, cuando son arrancadas, la cinta no permite posteriores regrabaciones y por tanto es de *sólo lectura*.

Cuando la lengüeta A es arrancada, los programas de la cara 1 están protegidos contra escritura, si por el contrario es la B serán protegidos los de la cara 2. Para arrancar dichas lengüetas es conveniente la utilización de un destornillador a modo de palanca.

Si una nueva regrabación fuese necesaria, los agujeros donde se encontraban las lengüetas deben ser tapados con cinta adhesiva.

ADVERTENCIA

Cuando se usan cintas de cassette protegidas contra escritura, la tecla RECORD de su cassette no permite ser pulsada. No intente forzarla, ya que podría dañar el aparato.



Almacenamiento en cartucho "Microdrive".

formadas por una secuencia de caracteres encerrados entre comillas; la estructura general es:

STRING
"cadena de caracteres"

Los caracteres pueden ser letras mayúsculas o minúsculas (a, z, l...), números (0, 4, 0...), caracteres especiales (©, #, 1,...) o comandos BASIC (VERIFY, BIN, OUT,...).

Ejemplos:

- "Curso BASIC/SINCLAIR"
- © MICROHOBBY & RAFAEL PRADES"
- "Enero tiene 31 días"
- "3,1415927 es el valor de PI"

Cuando una cadena no tiene ningún carácter ("") se denomina *vacía* o *nula*, la cadena (" ") no se considera vacía, ya que contiene el carácter correspondiente al espacio.

Si dentro de una cadena ha de ir incluido el carácter de las comillas ("), éste deberá tersear por duplicado, como indicador de que no es el final de la cadena.

Ejemplos:

- "El "Spectrum"" es un ordenador personal"
- "El significado de "Anticuerpo""es..."

Si visualizamos estas dos cadenas, anteponiendo el comando PRINT, observaremos que las comillas no aparecen repetidas.

Variables numéricas

Este tipo de variables está formada únicamente por letras y números, pudiendo estar constituida su longitud por distinta cantidad de caracteres. El primer carácter debe ser obligatoriamente una letra.

Ejemplos:

- Color
- V12
- puntuación
- xlp12

No existe ninguna diferencia entre las variables escritas en letras mayúsculas o minúsculas; los siguientes ejemplos se refieren a la misma variable escrita de distintas formas:

- FUERZA
- Fuerza
- fuerza
- FuErZa

En ocasiones, para facilitar una posterior interpretación del significado de la variable, puede introducirse el carácter espacio tantas veces como se quiera. No existe ninguna diferencia entre una variable escrita con espacios y otra que no lo esté.

Ejemplos:

- RESISTENCIA DEL AIRE
- Resistenciadel Aire
- Resistenciadelaire

El valor asignado a una variable de tipo numérico debe ser una constante u otra variable, ambas, lógicamente, de tipo numérico.

Ejemplos:

- KILO = 12
- PESO = KILO
- Potencia = 30
- grados = 27
- Temperatura = grados

Variables alfanuméricas

Las variables alfanuméricas o de cadena, están constituidas por una sola letra, mayúsculas o minúsculas, indistintamente, seguida del símbolo *dólar* "\$".

Ejemplos:

- M\$
- J\$
- X\$
- T\$

El valor asignado a una variable alfanumérica debe ser, o una constante o una variable, ambas del tipo cadena alfanumérica.

Ejemplos:

- S\$ = "Producto"
- T\$ = S\$
- K\$ = "1024 Kbytes"
- N\$ = "3518E + 14"

Constantes y variables

Los datos que el Spectrum procesa son de dos tipos: *constantes y variables*; ambos forman parte del argumento de algunas instrucciones.

Las constantes son datos que, durante la ejecución de un programa, no varían su valor, mientras que las variables pueden tomar distinto valor. La forma de representar una constante es por su valor; sin embargo, las variables se indican con un nombre simbólico.

Los datos también pueden dividirse en *numéricos y string* o *cadena de caracteres*. Podemos tener, por tanto, las siguientes combinaciones:

DATOS	
CONSTANTES	NUMERICAS
	STRING
VARIABLES	NUMERICAS
	STRING

Veamos unos ejemplos:

a) La fórmula que calcula el área de una superficie esférica es:

$$S = 4 \pi r^2$$

Como constantes numéricas tenemos el número "4", el exponente del radio (r) "2" y el valor de pi (π) "3,1415927"; las variables numéricas están indicadas por los símbolos "r" (radio) y "S" (superficie), que pueden tomar distintos valores durante la ejecución de un programa.

b) El argumento de la instrucción que imprime "El autor de RIMAS y LEYENDAS es BECQUER", es:

"El autor de "¡L\$," es "N\$ las constantes de cadena, son los string:

```
10 REM "CURSO BASIC/SINCLAIR"
20 REM
25 POKE 23658,8
30 BORDER 2: PAPER 2: INK 7: CLS
40 DRAW 0,175: DRAW 255,0: DRAW
  0,-175: DRAW -255,0
42 PLOT 48,135: DRAW 0,-15: DR
  AW 15,0: DRAW 0,3: DRAW -12,0: D
  RAW 0,9: DRAW 12,0: DRAW 0,3: DR
  AW -15,0
44 PLOT 72,135: DRAW 0,-15: DR
  AW 15,0: DRAW 0,15: DRAW -15,0:
  : DRAW 75,132: DRAW 0,-9: DRAW 9,0
  : DRAW 0,9: DRAW -9,0
46 PLOT 96,135: DRAW 0,-15: DR
  AW 15,0: DRAW 0,23: DRAW -3,0: D
  RAW 0,-8: DRAW -11,0: PLOT 99,13
  2: DRAW 0,-9: DRAW 9,0: DRAW 0,9
  : DRAW -9,0
48 PLOT 120,135: DRAW 0,-15: D
  RAW 15,0: DRAW 0,3: DRAW -12,0:
  DRAW 0,3: DRAW 12,0: DRAW 0,9: D
  RAW -15,0: PLOT 123,132: DRAW 0,
  -3: DRAW 9,0: DRAW 0,3: DRAW -9,
  0
50 PLOT 144,143: DRAW 0,-23: D
  RAW 15,0: DRAW 0,15: DRAW -12,0:
  DRAW 0,8: DRAW -3,0: PLOT 147,1
  9: DRAW -9,0: DRAW 9,0: DRAW 0,
  0
52 PLOT 168,135: DRAW 0,-15: D
  RAW 15,0: DRAW 0,3: DRAW -12,0:
  DRAW 0,12: DRAW -3,0: PLOT 168,1
  43: DRAW 0,-4: DRAW 3,0: DRAW 0,
  4: DRAW -3,0
54 PLOT 192,135: DRAW 0,-15: D
  RAW 3,0: DRAW 0,12: DRAW 9,0: DR
  AW 0,-12: DRAW 3,0: DRAW 0,15: D
  RAW -15,0
56 PRINT FLASH 1; AT 12,9: "PARE
  LA CINTA"
58 PLOT 0,23: DRAW 255,0
60 PRINT AT 20,1: " ";
62 FOR X=1 TO 28
64 READ A
66 PRINT CHR$ A; BEEP 0.05,A/2
68 NEXT X
70 DATA 127,32,77,73,67,82,79,
  72,79,66,66,89,32,38,32,82,55,70
  ,65,69,76,32,80,82,65,68,69,83
  ..
72 PRINT AT 12,9: "
74 PRINT #0; AT 1,1: "PULSE UNA
  TECLA PARA CONTINUAR"
76 PAUSE 0
78 BEEP 0.2,20
100 BORDER 1: PAPER 1: INK 7: C
  LS
200 REM "PLANTILLA"
201 REM
202 REM
210 DRAW 0,175: DRAW 255,0: DRA
  W 0,-175: DRAW -255,0
212 PLOT 0,151: DRAW 255,0
218 PRINT AT 1,1: " ";
```



```

220 RESTORE
222 FOR X=1 TO 28
224 READ A: CHR$ A;
226 PRINT X
230 NEXT X
232 DRAW 63,128
234 DRAW 129,0: DRAW 0,-9: DRAW
-129,0: DRAW 0,9
250 PRINT AT 6,12: "BINARIO"
260 PLOT 63,103
270 DRAW 128,0: DRAW 0,-24: DRA
-128,0: DRAW 0,24
280 FOR X=71 TO 183 STEP 8
290 PLOT X,103: DRAW 0,-24
300 NEXT X
330 PLOT 63,48
340 DRAW 57,0: DRAW 0,-9: DRAW
-57,0: DRAW 0,9
360 PLOT 63,26
370 DRAW 57,0: DRAW 0,-13: DRAW
-57,0: DRAW 0,13
380 PRINT AT 16,8: OVER 1;"DECI
MAL"
470 REM "ENTRADA DE DATOS"
480 REM
490 REM
500 LET N$="0"
510 PRINT #0;"AT 1,0; FLASH 1;">
520 FOR X=1 TO 6
530 PAUSE 0
540 LET A$=INKEY$
550 IF X<>1 THEN GO TO 700
560 IF A$="." THEN LET S$=A$: P
RINT #0;S$; GO TO 745
570 IF A$="0" AND A$<="9" THEN
LET S$="": LET N$=A$: PRINT #0;
S$+N$; GO TO 745
580 GO TO 530
590 IF CODE A$=13 THEN BEEP 0.0
5:20: GO TO 800
710 IF A$="0" AND A$<="9" THEN
GO TO 730
720 GO TO 530
730 PRINT #0;A$;
740 LET N$=N$+A$
745 BEEP 0.05,20
750 NEXT X
770 REM "VERIFICACION"
780 REM
790 REM
800 LET N=VAL (N$)
810 IF N<0 OR N>65535 THEN PRIN
T #0;AT 1,0;"Numero no valido.";
FOR X=1 TO 200: NEXT X: PRINT #
0;AT 1,0;
TO 500
812 REM "DECIMAL"
814 REM
816 REM
818 PRINT #0;AT 1,0;"Espera un
momento por favor."
820 PRINT AT 19,8;S$
830 LET Y=19
840 IF N<10 THEN LET X=14

```

"El autor de "y" es "

y como variables, los símbolos L\$ y N\$; éstos, respectivamente, almacenan el título y el autor del libro; en un momento determinado podrían tomar los valores L\$ = "EL QUIJOTE" y N\$ = "CERVANTES", en este caso el mensaje a imprimir sería:

El autor de EL QUIJOTE es CERVANTES

Constantes numéricas

En el BASIC/Sinclair, los números se representan en cuatro formatos:

- NOTACION ENTERA.
- NOTACION FRACCIONARIA.
- NOTACION EXPONENCIAL.
- NOTACION BINARIA.

Notación entera

En este formato se representan los números enteros, que por definición son aquellos que no contienen parte decimal.

Los números enteros siguen las siguientes reglas:

- Pueden ser positivos y negativos.
- Precisión máxima de ocho dígitos.
- El signo "+" o espacio en blanco indica número positivo y el signo "—" negativo.

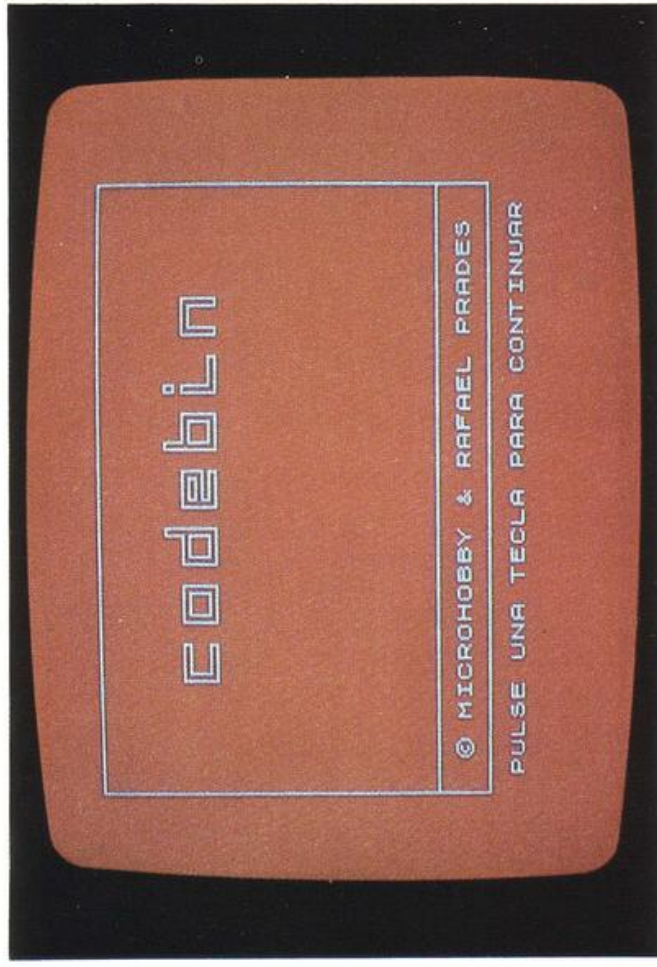
Por tanto, un número entero es- tá comprendido dentro del ran- go + 99999999 y - 99999999.

Ejemplos correctos:

- + 15934
- 823
- 9357
- 1
- + 12345678

Ejemplos incorrectos:

- 12.347 (Número con de- cimales).



Portada del programa CODEBIN.

el gráfico de la figura 4 que lo explica con detalle. El n.º a transformar es el 19, éste se divide entre dos, el cociente resultante se vuelve a dividir otra vez en dos, y así sucesivamente hasta que el último cociente sea 1; el número binario se forma con este cociente y con los restos obtenidos en las divisiones, tal como indica la figura.

Binario-decimal

Un número se transforma en su correspondiente decimal multiplicando cada símbolo (0 ó 1) por el valor de la posición que ocupa en base 2; sumando todos los valores se obtiene el decimal. La figura 5 representa gráficamente esta transformación.

Ejercicio

Si desea convertir un número binario en decimal, sin te-

ner que hacer cálculos engorrosos, utilice el siguiente comando directo para números positivos:

PRINT BIN "notación binaria" y para los negativos:

PRINT -BIN "notación binaria"

ejemplos:

PRINT BIN 1010

PRINT -BIN 111101

Si, por el contrario, lo que desea es pasar un número decimal a binario, edite el programa que le proponemos en este capítulo.

Cuando lo salve en cinta, hágalo de la forma:

SAVE "Codebin" LINE 10

De esta manera se autocargará al utilizar el comando LOAD y no hará falta teclear RUN una vez cargado.

Realizada la presentación del programa en pantalla, pul-

sando cualquier tecla, se pasa a la ejecución del programa principal. En la parte inferior aparece un prompt parpadeante " " indicando que el ordenador está preparado para la introducción del número decimal. Para teclear un negativo, es necesario que vaya precedido por el signo " - ". Si el valor del dato fuera superior a ± 65535, el ordenador nos presentará un mensaje de error y se visualizará de nuevo el prompt. Cuando el dato es correcto, aparece un mensaje de espera, ya que el ordenador necesita un tiempo para realizar los cálculos; finalizado éste, es presentado de forma binaria el n.º deseado. Para realizar un nuevo cálculo, pulse la tecla "S" e introduzca el nuevo valor.

Constantes alfanuméricas

También son conocidas como cadenas o strings. Están


```

850 IF N>=10 AND N<=99 THEN LET
  X=13
860 IF N>=100 AND N<=999 THEN L
  ET X=12
870 IF N>=1000 AND N<=9999 THEN
  LET X=11
880 IF N>=10000 THEN LET X=10
890 PRINT AT Y,X;N
900 REM
901 REM "BINARIO"
910 DIM I(16)
915 LET Z=N
920 FOR Y=15 TO 0 STEP -1
930 IF Z>=INT (2^Y) THEN LET Z=
  Z-INT (2^Y); LET I(Y+1)=1
940 NEXT Y
950 PRINT AT 10,6;S$
960 PRINT AT 10,7; OVER 1;" ";
970 FOR X=16 TO 1 STEP -1
980 PRINT OVER 1;I(X);
990 NEXT X
1000 PRINT #0;AT 1,0;"Desea otro
  calculo (S/N)
1010 PAUSE 0
1020 LET A$=INKEY$
1030 IF A$="S" THEN BEEP 0.2,20;
  GO TO 1055
1040 IF A$="N" THEN BEEP 0.2,20;
  STOP
1050 GO TO 1010
1055 PRINT AT 10,7; OVER 1;" ";
1060 FOR X=16 TO 1 STEP -1
1070 PRINT OVER 1;I(X);
1080 NEXT X
1090 PRINT AT 10,6;" "
1100 PRINT AT 10,8;" "
1150 PRINT #0;AT 1,0;" "
1200 GO TO 500

```

NUMERO	NOTACION EXPONENCIAL
43012	43.012×10^3
370000	37×10^4
93.23	9.323×10^2
93.23	9.323×10^1
1000000	1×10^6

El formato que utiliza el Spectrum para representar la notación exponencial, es el indicado en la fig. 1.

La *mantisa* es un número de ocho dígitos de precisión máxima.

La letra *e* (minúscula) o *E* (mayúscula) es un símbolo que indica que la base de potenciación es decimal (10).

El *exponente* es un número de dos dígitos de precisión máxima, que expresa la potencia a la que hay que elevar la base.

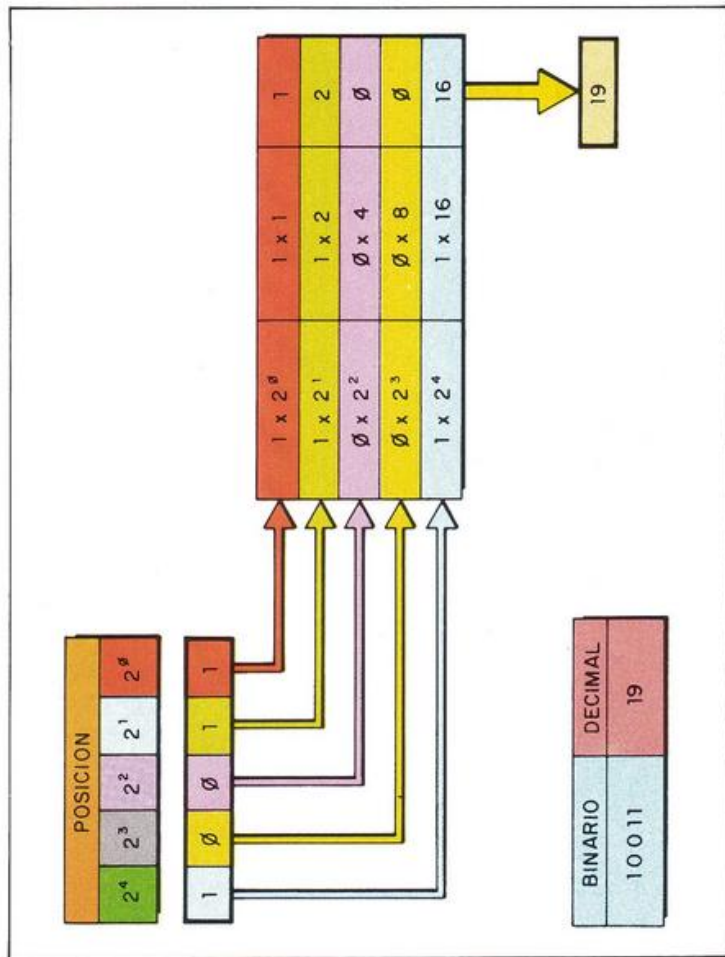
Tanto la mantisa como el exponente tienen sus propios signos "+" o "-".

En el ejemplo de la **figura 1**, podemos comprobar la notación 3.7281234 E-14 es equivalente a $3.7281234 \times 10^{-14}$. Desplazando la coma (punto en BASIC) y rectificando por tanto el valor del exponente, podríamos tener muchas combinaciones y todas con el mismo valor, por ejemplo:

1. $0.0003728 \times 10^{-10}$
2. $0.0037281 \times 10^{-11}$
3. $0.0372812 \times 10^{-12}$
4. $0.3728123 \times 10^{-13}$
5. $3.7281234 \times 10^{-14}$

De todas éstas, el Spectrum utiliza únicamente la 5, ya que proporciona el mayor número de cifras significativas en la

MICROBASIC 27



número inferior a éste y superior a "X" (1.46936794E — 39).

número inferior a éste y superior a "X" (1.46936794E — 39).

Cada número tiene un valor dependiendo de la posición que ocupa; por ejemplo, en el n.º 2537, el 2 ocupa la posición de las unidades de millar y tiene el valor 2000, el 5 las centenas (500), el 3 las decenas (30) y el 7 las unidades (7); sumando todos, obtenemos el valor 2537.

Otro sistema de numeración es el que utiliza internamente el ordenador y que se llama BIN (65535).

20 PRINT a
30 GO TO 10

Notación binaria

Para expresar un número, normalmente empleamos el sistema decimal, en el que se manejan diez símbolos (0, 1, 2, ..., 9). En los sistemas de base 2, el valor de los símbolos también es opcional, ya que depende de la posición que ocupen.

	BIN	BIN
10	10	10
1111	1111	1111

Decimal-binario

En el Spectrum, los datos que deban ser introducidos en el programa se pueden expresar cualquiera de las formas siguientes:

935749103 (Tiene nueve di-
nitos) que el ordenador efectúa un
redondeo

32G14 El campo de los números
decimales está comprendido
entre + 0.0001 y + 99999999
/- 0.0001 y - 99999999.

Notación decimal

Sirve para representar aquellos números que contienen decimales. En BASIC, la "coma" Cuando el dígito situado a la izquierda del punto decimal es cero, su representación puede ser omitida; es decir, la cifra .15 es la misma que 0.15.

decimal" se sustituye por el "punto decimal", aunque su re-

presentación se denomine de "coma fija". Al igual que los números enteros, la precisión de los datos de salida es de

Otra forma de expresar números es mediante la notación

935749103 (Tiene nueve dígitos)

32G14 (Contiene un carácter no numérico).

Notación decimal

Sirve para representar aquellos números que contienen decimales. En BASIC, la "coma decimal" se sustituye por el "punto decimal", aunque su representación se denomine de "*coma fija*". Al igual que los números enteros, la precisión de los datos de salida es de ocho cifras significativas; sin embargo, los datos introducidos por teclado (entrada) pueden tener mayor longitud, va-

número inferior a éste y superior a "X" (1.46936794E — 39) valor, utilizando las potencias de base 10 (fig. 3).

Cada número tiene un valor dependiendo de la posición que ocupa; por ejemplo, en el n.º 2537, el 2 ocupa la posición de las unidades de millar y tiene el valor 2000, el 5 las centenas (500), el 3 las decenas (30) y el 7 las unidades (7); sumando todos, obtenemos el valor 2537.

10	PRINT	a	Otro sistema de numeración es el que utiliza internamente el ordenador y que se llama <i>binario</i> , ya que sólo utiliza dos símbolos, el 0 y el 1.
20	PRINT		
30	GO TO	10	

Notación binaria

3, 4, 5, 6, 7, 8 y 9) con los que se puede expresar cualquier

30 MICROBASIC

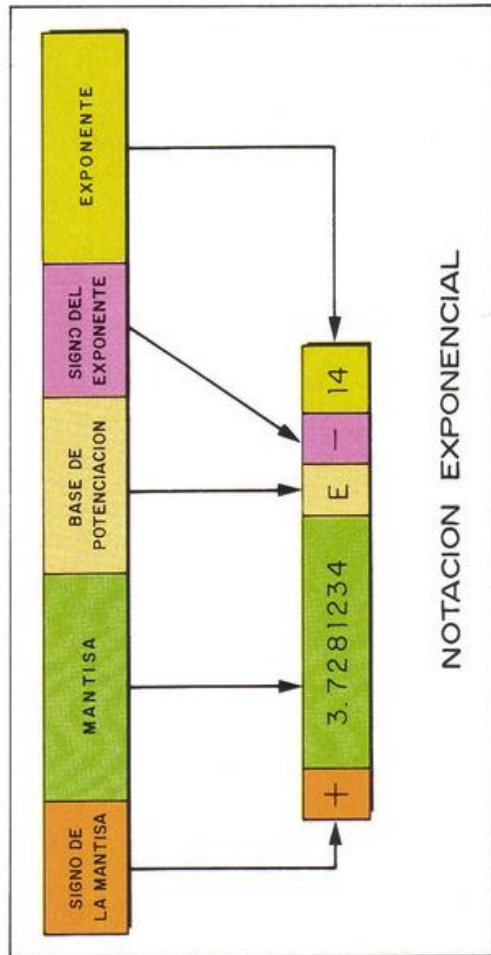


Fig. 1.

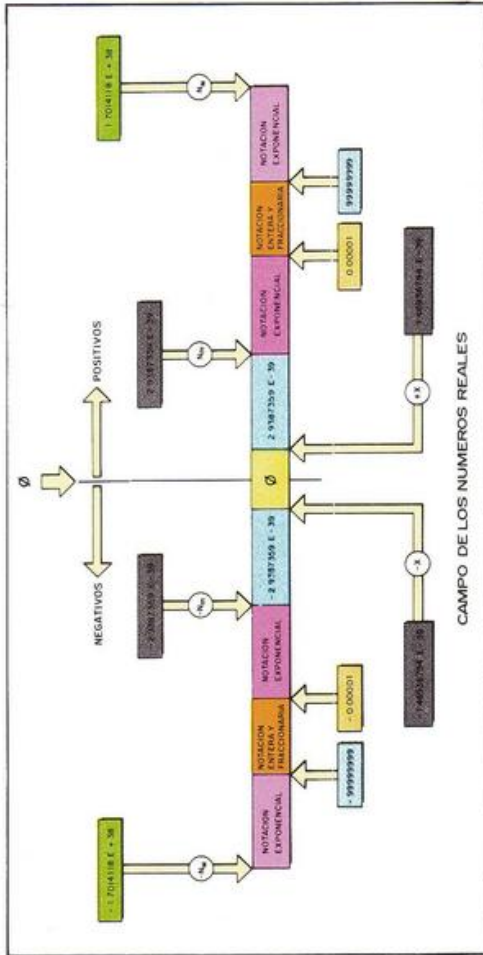


Fig. 2.

mantisa (un dígito distinto de cero, punto decimal y hasta siete dígitos más); este tipo de representación se denomina *coma flotante*.

La notación exponencial la utiliza el ordenador para representar números cuyo valor sea inferior a 0.000001 o superiores a 99999999 , dentro del campo de los números positivos.

En la figura 2, se puede observar el campo de los números

reales; a la derecha del \emptyset se encuentran los números positivos y a la izquierda los negativos; ya que ambas partes son iguales, bastará con explicar una de ellas.

El mayor número positivo que se puede representar es "NM", cuyo valor es igual a $1.7014118 \text{ E} + 38$; cuando el ordenador realiza una operación y el resultado es superior

a "NM", se presenta en pantalla el mensaje:

- 6 Number too big, a : b
- a = línea de programa donde se originó el error.
- b = n.º de instrucción dentro de la línea.

El menor número positivo es "Nm", y su valor es $2.9387359 \text{ E} - 39$; cualquier

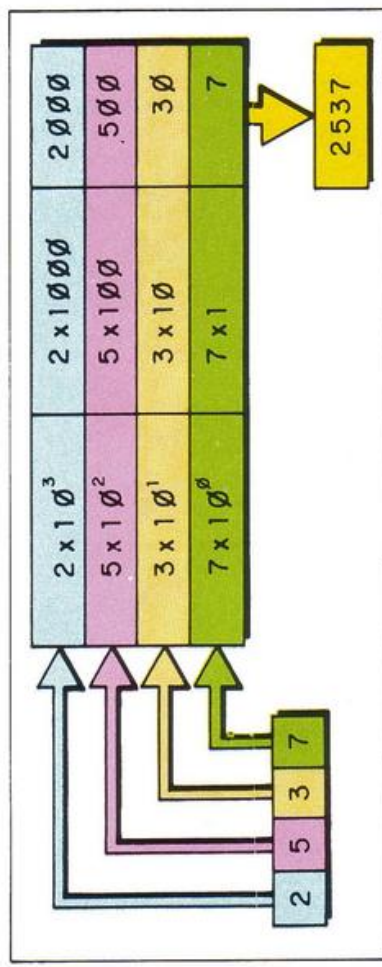


Fig. 3.

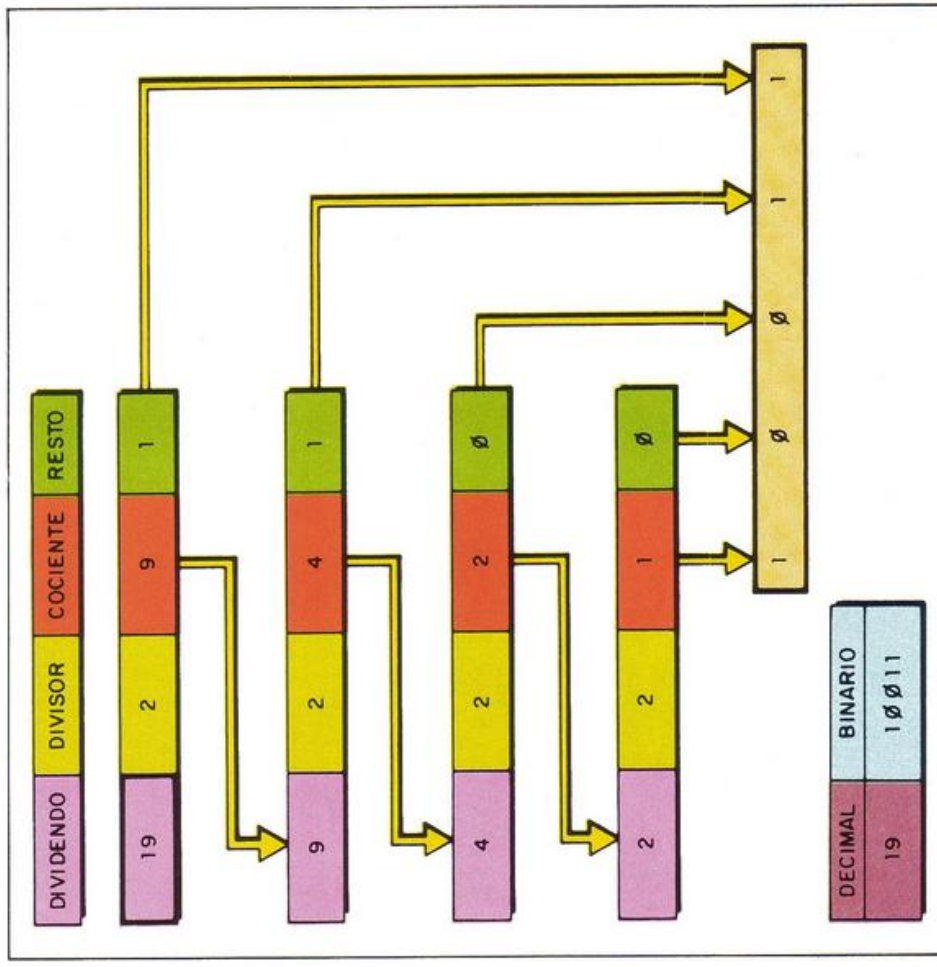
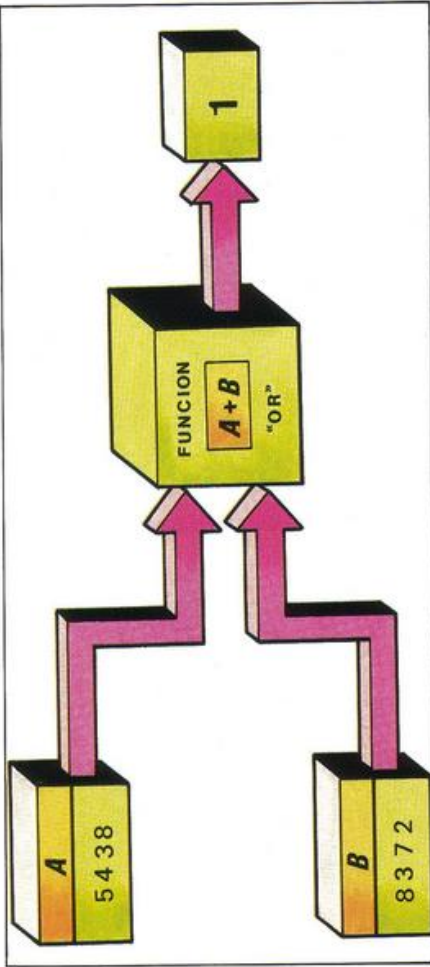


Fig. 4.



OPERADORES

Los operadores son símbolos que expresan el tipo de operación que ha de realizarse, bien entre dos constantes, bien entre una variable y una constante, etc. Veamos unos ejemplos.

OPERACION	SÍMBOLO
Suma	+
Resta	-
Multiplicación	*
División	/
Potenciación	↑

Ejemplos:

$10\phi 2 * 7$ (10 ϕ 2 por 7)
 $3\uparrow 2$ (3 al cuadrado)
 $456 / 2$ (456 entre 2)

2 Variable not found

Al igual que en álgebra, se pueden utilizar los paréntesis, pero no las multiplicaciones implícitas, es decir, la expresión $(5 + X) (8 - Y)$, en BASIC, se escribe:

$(5 + X) * (8 - Y)$

Transmisión del ASCII

En el lenguaje BASIC del Spectrum, la transmisión de caracteres ASCII puede efectuarse o directamente en este código o anteponiendo la sentencia "CHR \$" al correspondiente código decimal.

Ejemplos:

ASCII	DECIMAL
PRINT "A"	PRINT CHR\$ 65
PRINT "a"	PRINT CHR\$ 97

El primer método ofrece ciertas ventajas con respecto al segundo.

- Es legible directamente.
- Ocupa menos sentencias.
- Más rápido de ejecución.

si por el contrario todo lo que desea es semi-camufilar el mensaje, será conveniente utilizar el segundo.

Ejecute la siguiente instrucción directa y compare el resultado con el proporcionado por el programa número "3".

```
PRINT "Maese Pérez el organista"
```

PROGRAMA 3

```

10 REM *****
   *          CURSO
   * BASIC/SINCLAIR
   * "CHR$"
   * *****
LS 20 BORDER 7: PAPER 7: INK 0: C
30 REM *****
   * IMPRESION DE
   * CARACTERES
   * *****
35 RESTORE
40 FOR X=1 TO 24
50 READ codigo decimal
60 PRINT CHR$ codigo decimal;
70 NEXT X
80 REM *****
   * TABLA DE
   * CARACTERES
   * *****
90 DATA 77,97,101,115,101,32,8
0,101,114,101,122,32,101,108,32,
111,114,103,97,110,105,115,116,9
7

```

Expresiones aritméticas

Son conjuntos de constantes y variables unidas entre sí por operadores aritméticos.

Ejemplos:

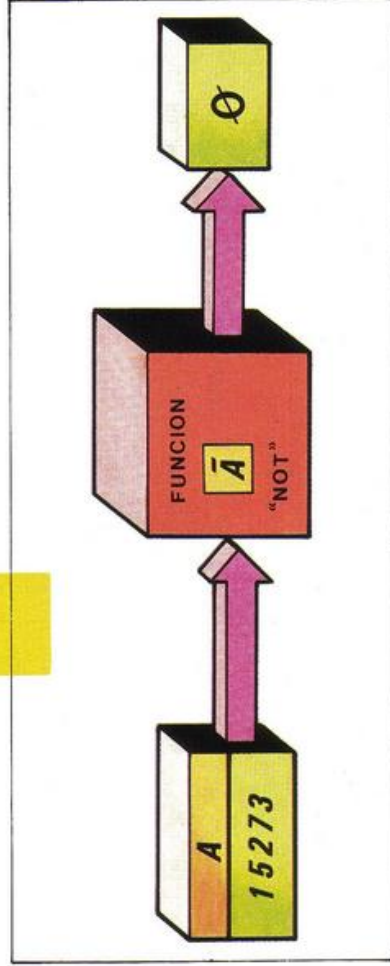
$(3 + 5) / 7$
 $8\uparrow 2 / 10\phi * 5$
 $gr * X\uparrow 3 / tz$

voltaje \uparrow 2 / resistencia * tiempo

Las variables utilizadas tienen que estar definidas previamente que estar definidas previamente

Cálculo de expresiones

El ordenador cuando realiza un cálculo, lo hace siempre atendiendo al valor de prioridad que tenga cada operación. De las cinco operaciones aritméticas básicas, la potenciación tiene mayor prioridad que el resto, después la multiplicación y la división, ambas con el mismo valor, y por último, la suma y la resta, también con idéntica prioridad.



Función "AND". Estructura 1.

Cuando en una expresión hay dos operaciones de la misma prioridad, el ordenador efectúa los cálculos de izquierda a derecha.

Vamos por pasos cómo el ordenador calcula la siguiente expresión para $x = 3$ y $y = 2$:

- $16 + x + 3 \cdot 3\phi - y / 2$
 - $16 + 3 + 3\phi - 2\phi / 2$
 - $16 + 9\phi - 2\phi / 2$
 - $16 + 9\phi - 1\phi$
- Resultado: 96

Utilizando los paréntesis se puede alterar el orden de evaluación de las operaciones, ya que éstos se calculan primero. Aprovechando la expresión anterior, vamos a observar que el resultado varía colocando los paréntesis en las operaciones de menos prioridad.

- $(16 + x) \cdot (3\phi - y) / 2$
 - $16 + (3\phi - 2\phi) / 2$
 - $18 \cdot (3\phi - 2\phi) / 2$
 - $18 \cdot 1\phi / 2$
- Resultado: 9 ϕ

Operadores de relación

Permiten realizar las comparaciones entre operandos (constantes o variables) tanto numéricos como de cadena.

OPERADOR	SÍMBOLO
Igual	=
Distinto	<>
Mayor	>
Menor	<
Mayor o igual	>=
Menor o igual	<=

Las operaciones realizadas con estos operadores sólo tie-

nen dos resultados o soluciones, es decir:

— Si la condición impuesta por el operador se cumple, es decir, que es *verdadera* (true), el valor del resultado es «1».

— Si por el contrario, la condición no se cumple, es decir, que es *falsa* (false), el valor se vuelve «0».

Ejecute los siguientes comandos directos y comprobará lo explicado anteriormente.

COMANDO	CONDICIÓN	RESULTADO
PRINT 3 = 7	Falsa	0
PRINT 10 > 99	Verdadera	1
PRINT 8 < 7	Falsa	0
PRINT 10 >= 7	Verdadera	1

El símbolo «=» también sirve para asignar un valor a una variable.

Operadores lógicos

Se utilizan para realizar las operaciones lógicas a nivel de expresión.

OPERACION	FUNCIÓN
Producto lógico	AND
Suma lógica	OR
Negación	NOT

En otras ocasiones, estos operadores se utilizan para relacionar dos expresiones mediante una condición, por ejemplo:

— Que el ordenador realice una determinada tarea si se cumplen varias relaciones.

RELACION	OPERADOR	RELACION
$x >= \phi$	AND	$x <= 9$

En este caso sólo se cumplirá la condición cuando la variable x sea mayor o igual a 9, es decir, que esté comprendida entre ϕ y 9.

— Que se realice la tarea cuando simplemente alguna de las condiciones se cumpla.

RELACION	OPERADOR	RELACION
$a = 10\phi$	OR	$t > 1$

La condición se cumple, bien cuando el valor de la variable a sea 10ϕ , bien cuando el valor de t sea mayor que 1 ó bien cuando se cumplan ambas condiciones.

Función «AND»

La estructura de esta función es la que se muestra a continuación, siendo a y b dos expresiones numéricas.

a AND b

Estas expresiones solamente pueden tomar los valores ϕ o distinto de ϕ , este último le vamos a representar como «>> ϕ ». Teniendo en cuenta todas las posibles combinaciones que pueden tomar a y b , vamos a mostrar su *tabla de verdad*:

a	b	RESULTADO
ϕ	ϕ	ϕ
ϕ	>> ϕ	a (ϕ)
>> ϕ	ϕ	ϕ
>> ϕ	>> ϕ	a

de ésta se deduce que si el valor de b es igual a ϕ , el resultado de la función es ϕ , independientemente del valor de a . Si el valor de b es distinto de ϕ , la función asume el valor de a .

del Spectrum, otros ordenadores disponen de otro juego distinto.

El código transparente está comprendido entre el ϕ y el 31, en decimal. Dentro de éste, existen caracteres que no son utilizados, cuando el Spectrum representa uno de estos códigos aparece una interrogación (?) en su lugar.

Los códigos comprendidos entre el 32 y el 127 forman el ASCII convencional de todo ordenador. En esta zona se encuentran los caracteres correspondientes a las 26 letras mayúsculas, a sus homólogas las minúsculas, a los diez dígitos (ϕ al 9), al carácter «espacio» y a una serie de símbolos (ϕ , &, &...) y signos ortográficos («., " ' , ? , ... »).

A pesar de ser la zona convencional del ASCII, el Spectrum tiene dos caracteres particulares, los correspondientes a la «Libra» (£) y al «Copy-right» (©), códigos 96 y 127, respectivamente.

Los gráficos predefinidos, es decir, los símbolos que están dibujados sobre las teclas con los números «1» a «8», y los complementarios, tienen un código comprendido entre el 128 y el 143. A continuación se encuentran los códigos correspondientes a los «DGU» (Gráficos Definidos por el Usuario), que como ya se indicó en el capítulo primero, éstos están asignados a las teclas con las letras de «A» y la «U» y su código está comprendido entre el 144 y el 164.

Al final del juego de caracteres se encuentran los *tokens*, término que simboliza a las 88 palabras clave (RUN, STOP, CLEAR...) una vez codificadas, éstas forman la lista de comandos, sentencias y funciones. También se encuentran en esta zona tres caracte-

PROGRAMA 2

```

10 REM *****
    *          CURSO          *
    * BASIC/SINCLAIR          *
    * ASCII/DECIMAL          *
    * *****

20 BORDER 1: PAPER 1: INK 7: C
L$
22 LET FLAG$2=23658
24 LET PIP=23609
30 REM

*****
* ENTRADA DE *
* CARACTERES *
*****

32 POKE PIP, 100
34 POKE FLAG$2, 0
40 INPUT "CADENA: "; A$
50 IF LEN A$ > 20 THEN GO TO 40
60 POKE PIP, 0
70 REM

*****
* VISUALIZACION *
*****

80 FOR X=1 TO LEN A$; CODE A$(X)
90 PRINT A$(X); " ";
100 NEXT X
110 REM

*****
* CONTINUACION *
*****

120 POKE FLAG$2, 8
130 PRINT #0; AT 1, 0; "Quiere vol
ver a empezar (S/N)";
140 PAUSE 0; LET D$=INKEY$
150 IF D$="S" THEN BEEP 0.05, 20
: GO TO 10
160 IF D$="N" THEN BEEP 0.05, 20
: CLS: STOP
170 BEEP 0.2, -15: GO TO 140

```


El programa «2» visualiza el código decimal correspondiente a la cadena ASCII introducida, de un máximo de 20 caracteres.

Organización del ASCII

Dentro del código ASCII, usado por el Spectrum, pueden encontrarse diversas zo-

nas, teniendo cada una de ellas unas características distintas:

- CODIGO TRANSPARENTE.
- CODIGO ASCII CONVENCIONAL.
- CODIGO ASCII ESPECIFIC.
- GRAFICOS PREDEFINIDOS.

— GRAFICOS DE USUARIO.

— TOKENS.

El denominado código transparente está formado por una serie de comandos y funciones de control, como por ejemplo, el control de los cursores, el del color, la función EDIT, etc. Estos caracteres ASCII de control son específicos

OPERACION	SIMBOLO	PRIORIDAD	EJEMPLO
FRAGMENTACION	T0	12	"Juan" (1 TO 2)
POTENCIACION	↑	10	10 2
NEGACION	—	9	— 15
MULTIPLICACION	*	8	7 * 13
DIVISION	/		152 / 2
ADICION	+	6	10 + 4
SUBSTRACCION	—		8 — 5
OPERADORES RELACIONALES	=	5	10 4
	>		
	<		
	<>		
	>=		
OPERADORES LOGICOS	NOT	4	NOT 5
	AND	3	10 AND 1
	OR	2	7 OR 0

Ejemplos:

OPERACION	RESULTADO
132 AND 0	0
80 AND 10	80
0 AND 25	0
1 AND 90	1

La función AND, también puede tomar la estructura:

a\$ AND b

en este caso el resultado de la función será una cadena vacía

las expresiones a y b sólo pueden tomar valores numéricos. Con todas las combinaciones se obtiene la siguiente tabla de verdad:

a	b	Resultado
0	0	a (0)
0	<> 0	1
<> 0	0	a
<> 0	<> 0	1

Mirando detenidamente la tabla se observa que si b to-

Función «NOT»

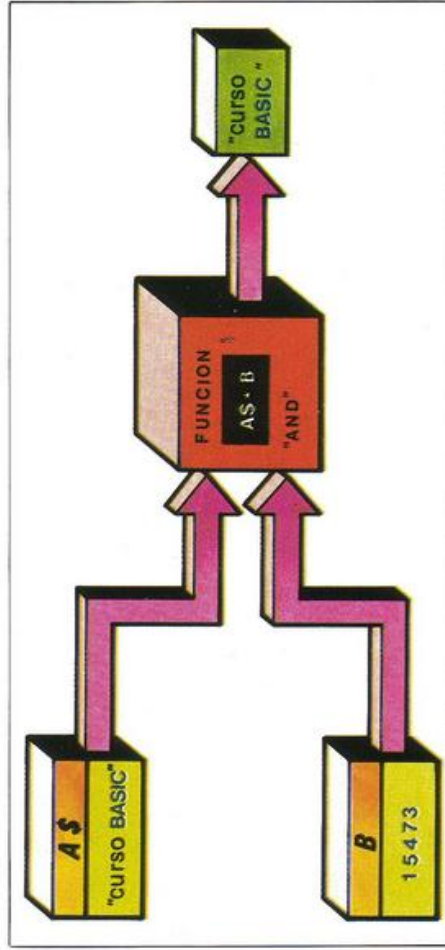
La función NOT afecta solamente a una expresión y su estructura es:

NOT a

su tabla de verdad es:

a	RESULTADO
0	1
<> 0	0

Como se puede apreciar, el resultado de la función es la



Función «AND». Estructura 2.

cuando b sea 0 y será la cadena a\$ cuando b sea distinto de 0.

Ejemplos:

OPERACION	RESULTADO
"PEDRO" AND 30	"PEDRO"
"JUAN" AND 0	

Función «OR»

La estructura de la función OR es la siguiente:

a OR b

ma el valor 0, la función resultante adquiere el de a, sin embargo, si el valor de b es diferente de 0, el valor de la función es 1, independientemente del que tenga a.

Ejemplo:

OPERACION	RESULTADO
0 OR 30	1
20 OR 0	20
55 OR 7	1
0 OR 0	0

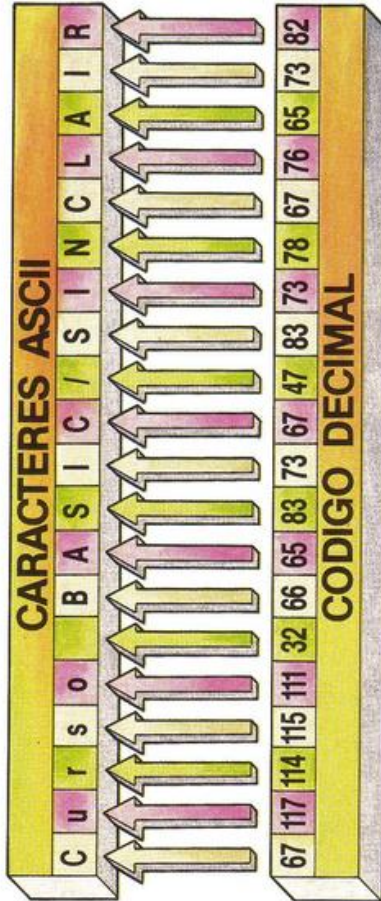
negación de la expresión a, es decir, vale 1 si a es igual a 0 y 0 si a es distinto de 0. Ejemplo:

OPERACION	RESULTADO
NOT 7	0
NOT 0	1

Anteponiendo el signo «—» a la función NOT, el resultado cambia de signo.

OPERACION	RESULTADO
— NOT 4	0
— NOT 0	— 1

CARACTERES ASCII



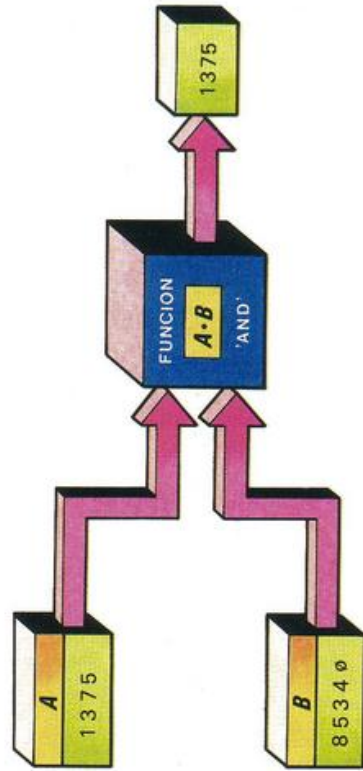
Ejercicio

b) 3 OR (NOT 3)
c) 3 OR ϕ
Resultado = 3

— NOT ((X OR Y) AND (NOT Z))

(X AND Y) OR (NOT X)

a) (3 AND 5) OR (NOT 3)



36 MICROBASIC

10 REM

```

* * * * *
CURSO
BASIC/SINCLAIR
"ASCII"
* * * * *

```

20 BORDER 1: PAPER 1: INK 7: C
L5

```

*****
30 REM
*****
IMPRESION DE
*****
CARACTERES
*****
*****

```

```

40 FOR Y=32 TO 255 STEP 44
50 FOR X=Y TO Y+43
60 IF X<100 THEN PRINT " "
70 PRINT X;" ";CHR$(X);
80 IF X=255 THEN GO TO 200
90 NEXT X
100 PRINT #0;AT 1,0;"Pulse
tecla para continuar."
110 PAUSE 0: BEEP 0.05,20
115 CLS
120 NEXT Y
130 REM

```

```

** ** ** ** ** ** 
** CONTINUACION ? **
** ** ** ** ** 

```

```

210 POKE 23558,8
220 PRINT #0;AT,0;"Desa o bte
ner un nuevo listado (S/N)."
230 PUSE 0;"LET D$=INKEY$
240 IF D$="S" THEN BEEP 0.05,20
GO TO 10
250 IF Q$="N" THEN BEEP 0.05,20
CLS
260 BEEP 0.2,-15: GO TO 230

```

ASCIL es la abreviatura, en inglés, de «American Standard Code for Information Interchange» que, traducido al idioma español, significa «Código normalizado Americano para intercambio de información».

El ASCII completo consta de 256 caracteres, cuyo código está comprendido entre 0 y 255.

Edite el programa número «1» que, una vez ejecutado, visualiza en pantalla los caracteres ASCII usados por el ZX Spectrum y comprendidos entre el código 32 y el 255.

Manejo de la tabla

Para conocer el código decimal correspondiente a un determinado carácter ASCII, basta con sumar los números de fila y columna indicados en la tabla.

código = fila + columna

Ejemplos:

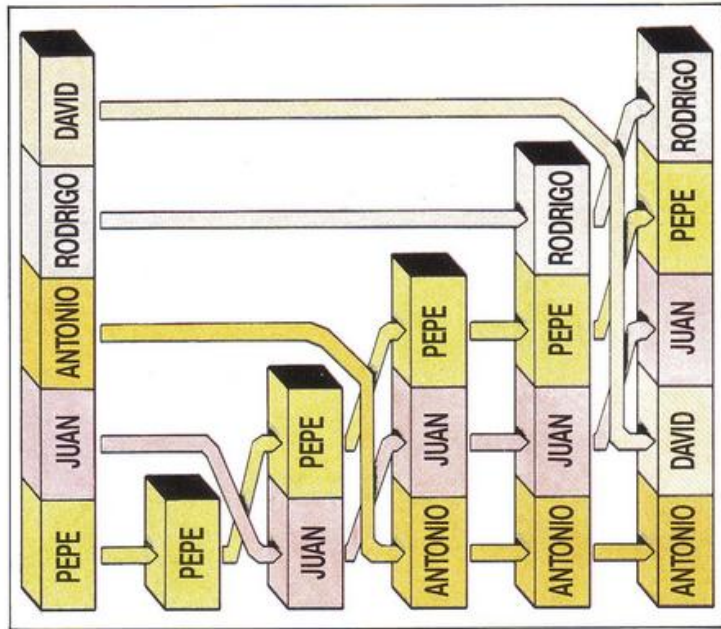
CARACTER ASCII	NUMERO		CODIGO
	FILE	COLUMNA	
±	30	5	35
USR	190	2	192
M	70	7	77
8	50	6	56

"ANTONIO", "DAVID", "JUAN", "PEPE"
Y "RODRIGO"

Prioridades

En el capítulo dedicado al cálculo de expresiones, se mencionó que al calcular el ordenador las operaciones aritméticas, tenía en cuenta la *prioridad* que éstas tenían. Una vez repasadas todas las operaciones (aritméticas, lógicas, de cadena, etc...) se ofrece en la página 38 un resumen de lo revisado. El ordenador asigna las prioridades con un número comprendido entre 1 y 16.

PROGRAMA 4



Metodología de ordenación de cadenas.

[illegible]

ORGANIZACION DEL JUEGO DE CARACTERES ASCII DEL SPECTRUM

"ASCII" USADO EN EL ZX SPECTRUM

COLUMNS												
0	1	2	3	4	5	6	7	8	9			
0	cursor abajo control	cursor over control	DELETE AT control	ENTER TAB control		numero	PRINT comma control	EDMT PAPER control	cursor tab FLASH control	cursor actin. BRIGHT control		
10	INVERSE		espacio	!		"	\$	%	&			
20		(*	+		,	.	/	0			
30		2	3	4	5	6	7	8	:			
40		<	=	>	?	@	A	B	C			
50		F	G	H	I	J	K	L	M			
60		P	Q	R	S	T	U	V	W			
70		Z	I	/	J	↑	—	£	a			
80		d	e	f	g	h	i	j	k			
90		n	o	p	q	r	s	t	u			
100		x	y	z	—	—	~	©	v			
110												
120												
130												
140												
150												
160												
170												
180												
190												
200												
210												
220												
230												
240												
250												

LETRA	ASCII
B	66
b	98

B < b

luego a\$ < b\$

Si fueran iguales los códigos del primer carácter, sería necesario pasar a comparar los siguientes hasta encontrar uno diferente.

Ejemplo:

a\$	F L O J O
b\$	F L O T A R

LETRA	ASCII
J	74
T	84

J < T

luego a\$ < b\$

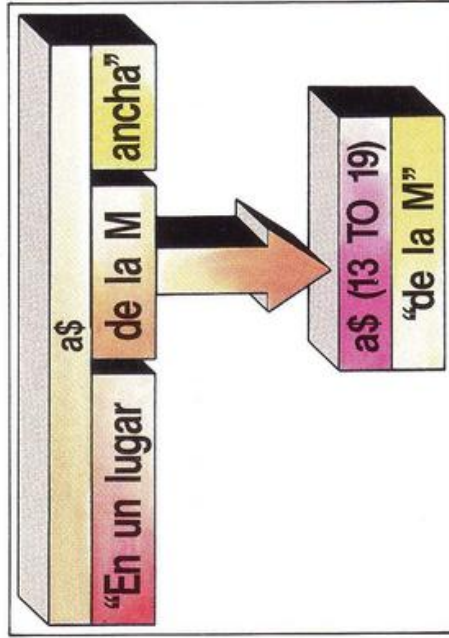
En este proceso de comparación puede ocurrir que los caracteres de una cadena se acaben antes que los de otra, en este caso, es mayor la que tenga mayor longitud.

Ejemplo:

a\$	F I N I T O
b\$	F I N

luego a\$ > b\$

Si las dos cadenas se acaban sin encontrar ningún carácter distinto, significa que ambas son iguales.



Fragmentación central.

Ejemplo:

a\$	C R E P I T A R
b\$	C R E P I T A R

luego a\$ = b\$

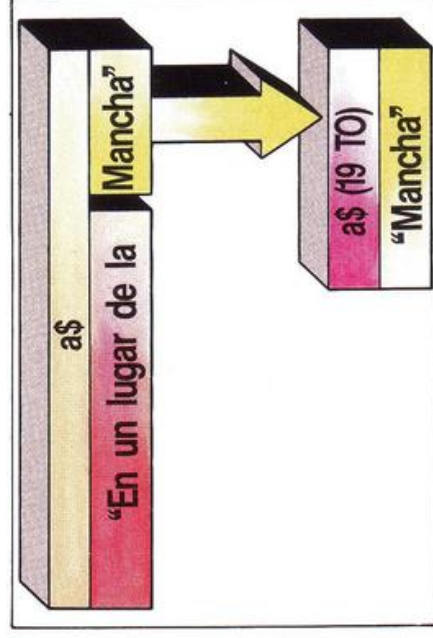
El carácter «espacio» también tiene su correspondiente código ASCII, por este motivo, al comparar cadenas alfanuméricas debe ser tenido en

cuenta. Aparentemente, en el siguiente ejemplo, las dos cadenas son iguales, sin embargo no lo son, ya que una de ellas contiene al final un espacio.

Ejemplo:

a\$	A B A N I C O
b\$	A B A N I C O

luego a\$ < b\$



Fragmentación derecha.

mente visualiza en la pantalla todas las posibles combinaciones de concatenación. Introduzca, por ejemplo, su nombre y dos apellidos, como parámetros y observe el resultado.

Subcadenas

Con el término de *subcadena* se designa a un grupo de caracteres consecutivos extraídos de una cadena. Por ejemplo «dicc» es una subcadena de «diccionario», también lo son «onar» y «ario»; sin embargo «dicic», «noar» y «aroi» no lo son, ya que sus caracteres no han sido extraídos consecutivamente.

Otros ejemplos son:

CADENAS	SUBCADENAS
«Coche de carreras»	«che de»
«Angel, Pepe, Luis»	«Pepe»
«lunesmartesmiércoles»	«lunes»
«Ordenador personal»	«Ordena»

Fragmentación

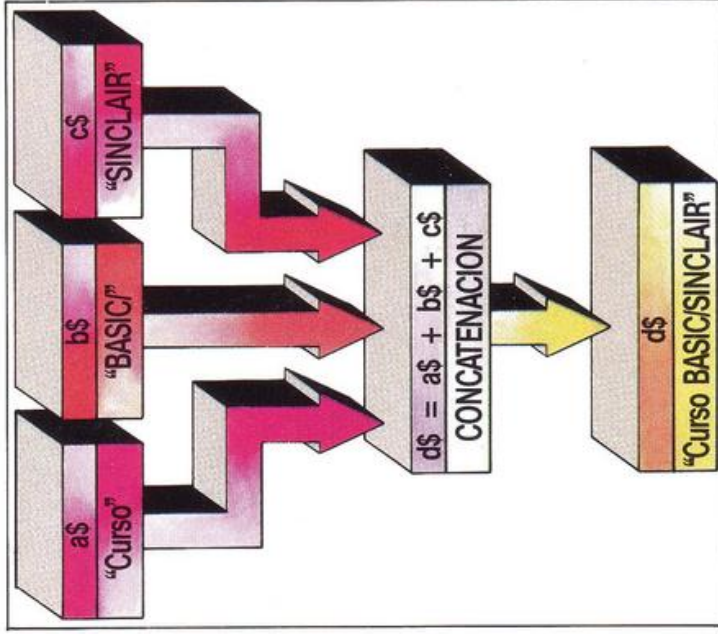
Para extraer una subcadena se utiliza la operación de *fragmentación*. La estructura general es:

FRAGMENTACION
cadena (a TO b)

donde «a» es el número del primer carácter a extraer y «b» el último, entre ambos limitan la longitud de la subcadena; «a» y «b» pueden ser constantes o variables numéricas, pero siempre positivas, de lo contrario, el ordenador enviará el mensaje:

B Integer out of range

La cadena puede ser o bien



Concatenación.

una constante o bien una variable.

Ejemplos:

FRAGMENTACION	RESULTADO
«Curso BASIC» (1 TO 5)	«Curso»
«Monitor TV» (6 TO 9)	«tor T»
«MICROHOBBY» (6 TO 10)	«HOBBY»
«Cartucho» (4 TO 5)	«tu»

Si «b» fuera menor que «a», el resultado de la fragmentación sería una cadena vacía.

Ejemplos:

FRAGMENTACION	RESULTADO
«Juan» (2 TO 1)	Cadena vacía ("")
«Pepe» (20 TO 10)	

Cuando «a» es 0 o «b» es mayor que el correspondiente a la longitud de la cadena, el

ordenador envía el siguiente mensaje:

3 Subscript wrong

Ejemplo:

- «Televisor» (0 TO 5)
- «Teclado» (2 TO 30)

El programa número «2» permite la introducción de una cadena de quince caracteres como máximo, es necesario además introducir los límites de la fragmentación. La subcadena resultante es visualizada en la pantalla.

Fragmentación específica

- Existen tres tipos de fragmentación que tienen una estructura distinta:
- Fragmentación de un solo carácter.

- Fragmentación izquierda.
- Fragmentación derecha.

Cuando se quiere extraer un solo carácter de una cadena, la estructura es la siguiente:

FRAGMENTACION
cadena (n)

donde «n» es el número de carácter, es equivalente a escribir «cadena» (n TO n).

Ejemplos:

FRAGMENTACION	RESULTADO
"Cassette" (5)	"g"
"Cable" (2)	"a"
"Circular" (3)	"r"
"Cigarrillos" (6)	"i"

La fragmentación izquierda consiste en la extracción de los «n» primeros caracteres de una cadena, su estructura es:

FRAGMENTACION
cadena (TO n)

equivalente a «cadena» (1 TO n)

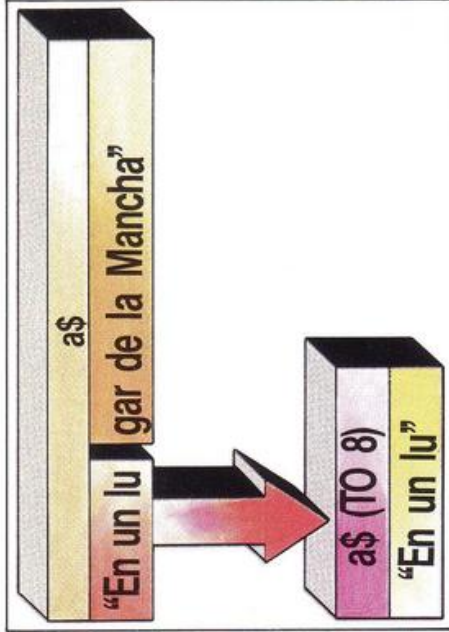
Ejemplos:

FRAGMENTACION	RESULTADO
"CARAMELO" (TO 3)	"CAR"
"boche" (TO 2)	"co"
"Botella" (TO 5)	"Botel"
"Corazon" (TO 4)	"Coraz"

La fragmentación derecha permite extraer los últimos caracteres de una cadena a partir de «n», su estructura es la siguiente:

FRAGMENTACION
cadena (n TO)

es equivalente a «cadena» (n TO fin).



Fragmentación izquierda.

PROGRAMA 2

```

10 REM *****
   *          CURSO
   * BASIC/SINCLAIR
   * *****
   * FRAGMENTACION *****
   * *****
20 BORDER 1: PAPER 1: INK 7: C
L$
25 POKE 23658,8
30 REM *****
   * INTRODUCCION *****
   *          CADENA *****
   * *****
100 INPUT "CADENA: "; A$
105 IF LEN A$ > 15 THEN GO TO 100
110 PRINT "CADENA "; A$
115 LET longitud=LEN(A$)
120 PRINT "LONGITUD: "; longitud
130 REM *****
   * INTRODUCCION *****
   *          LIMITES *****
   * *****
140 PRINT AT 21,0;"LIMITE INFER
   * (1-"; longitud);
145 INPUT LINE I$
150 IF I$ < "1" OR I$ > "9" THEN GO
   * TO 145

```

```

160 IF VAL (I$) > longitud THEN G
   * O TO 145
162 LET inferior=VAL (I$)
164 PRINT AT 4,0;"LIMITE INFERI
   * OR "; inferior
170 PRINT AT 21,7;"SUP"
175 INPUT LINE S$
180 IF S$ < "1" OR S$ > "9" THEN GO
   * TO 175
190 IF VAL (S$) > longitud THEN G
   * O TO 175
192 LET superior=VAL (S$)
194 PRINT AT 5,0;"LIMITE SUPERI
   * OR "; superior
196 PRINT AT 21,0;"
200 REM *****
   * VISUALIZACION *****
   *          SUBCADENA *****
   * *****
210 PRINT AT 0,0;"SUBCADENA: "
   * "A$(inferior TO superior)";
220 REM *****
   * CONTINUACION? *****
   * *****
225 POKE 23658,8
230 PRINT AT 21,0;"QUIERE CONTI
   * NUAR (S/N)";
240 PAUSE 0
250 LET D$=INKEY$
260 IF D$="S" THEN CLS : GO TO
   * 100
270 IF D$="N" THEN CLS : STOP
280 GO TO 240

```

Ejemplos:

FRAGMENTACION	RESULTADO
"Zumos" (3 TO)	"mos"
"pescados" (5 TO)	"ados"
"Puros" (2 TO)	"uros"
"A1257" (4 TO)	"57"

Asignación de sub cadenas

Una variable de cadena puede ser modificada parcialmente utilizando la asignación de sub cadenas. Supongamos que la variable a\$ tiene asignado un valor.

"Evolucion de los anfibios" utilizando la expresión:

a\$ (18 TO 25) = "caballos"
la variable a\$ tendrá un nuevo valor

Este método de recortar o rellenar con espacios se llama «asignación procustea». Utilizando este método se puede asignar un nuevo valor a toda la variable manteniendo la longitud de ésta.

Ejemplo:

```

a$ = "*****"
asignando
a$ ( ) = "$$$"
el nuevo valor de a$ será
"$$$"

```

Comparación de cadenas

Las cadenas al igual que los números pueden compararse haciendo uso de los operadores relacionales. Para averiguar si una cadena es mayor, menor o igual a otra, se comparan los códigos "ASCII" correspondientes al primer carácter de cada una de ellas; si estos son diferentes, será mayor la que tenga un código más alto.

Ejemplo 1:

a\$	b\$
P A L O M A	G O R R I O N

LETRA	ASCII
P	80
G	71

P < G

luego a\$ > b\$

Ejemplo 2:

a\$	b\$
B U I T R E	b u i t r e

"Evolucion de los caballos"
Cuando a una subcadena se le asigna una longitud de caracteres mayor que los expresados, automáticamente los caracteres sobrantes, de la derecha, son recortados.

Ejemplo:
a\$ = "*****"

asignando

a\$ (2 TO 5) = "\$\$\$\$\$"
la variable a\$ tendrá el valor
".\$\$\$\$\$"

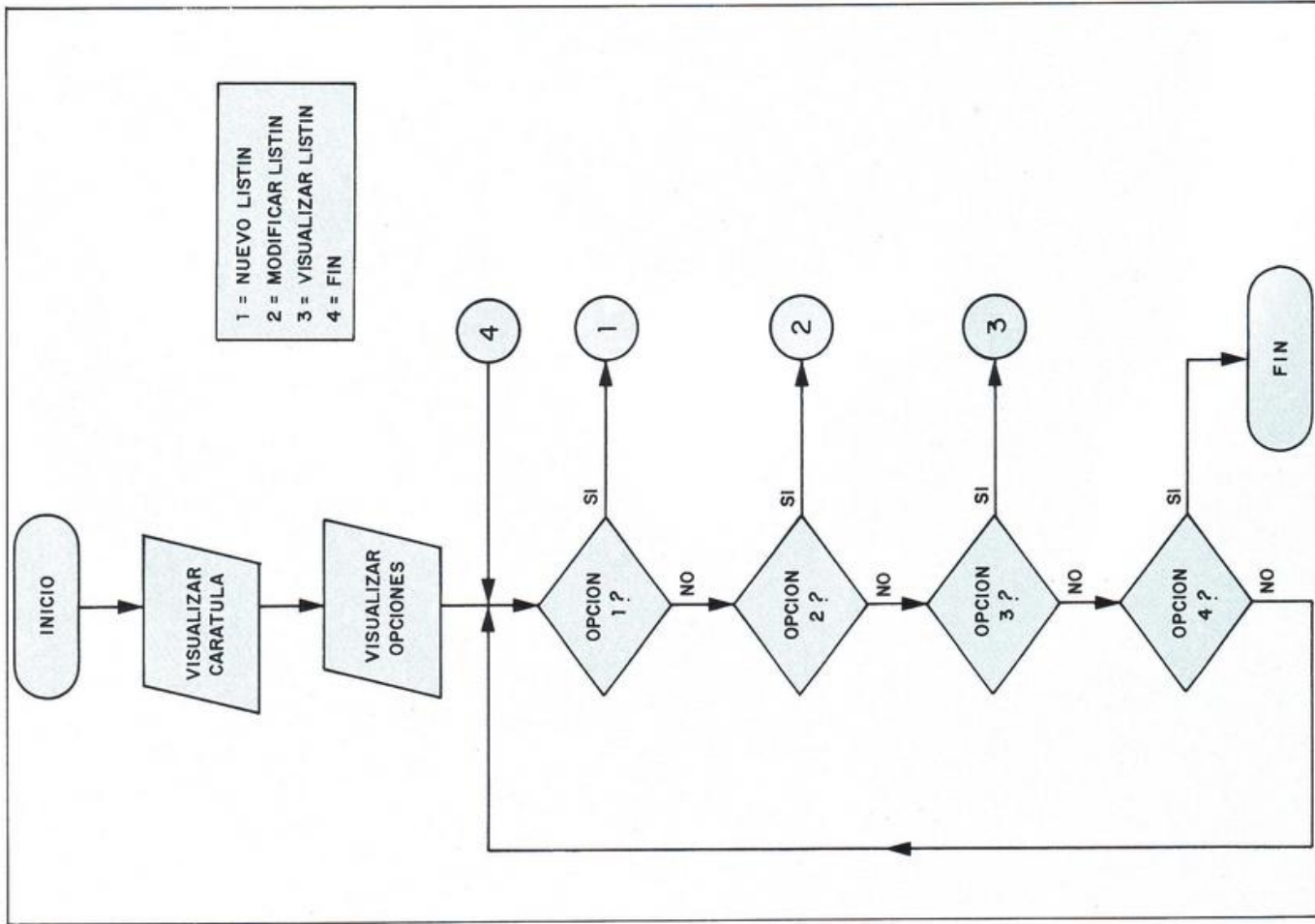
Cuando por el contrario la longitud sea inferior, la subcadena se rellena con espacios.

Ejemplo:

a\$ = "*****"

asignando

a\$ (2 TO 5) = "\$\$"
a\$ tendrá el valor
".\$\$ *****"



Menú de opciones.

```

*****
260 FOR X=2 TO orden
270 INPUT "CADENA: ";D$
280 IF LEN (D$)>longitud THEN G
O TO 270
290 REM
*****
***** BORRADO CADENAS *****
*****
510 FOR Y=1 TO 20
520 PRINT AT Y,11,"
535 LET N$(Y)="
540 GO TO 210
1000 REM
*****
***** SUB. DETECCION *****
*****
1010 FOR Z=X-1 TO 1 STEP -1
1020 IF D$>N$(Z) THEN GO TO 1050
1030 NEXT Z
1040 REM
*****
***** ORDENA *****
*****
1050 FOR Y=orden-1 TO Z+1 STEP -
1
1060 LET N$(Y+1)=N$(Y)
1070 PRINT AT Y+1,11;N$(Y)
1080 NEXT Y
1090 REM
*****
***** INSERTA *****
*****
1100 LET N$(Y+1)=D$
1110 PRINT AT Y+1,11;D$
1120 RETURN

```


Elaboración de programas

Antes de introducir las instrucciones necesarias para elaborar un programa, es necesario haber efectuado antes una serie de pasos. Existen diversas técnicas para ayudar al diseñador de programas a estructurar en alguna medida su trabajo y así poder conseguir unos resultados satisfactorios.

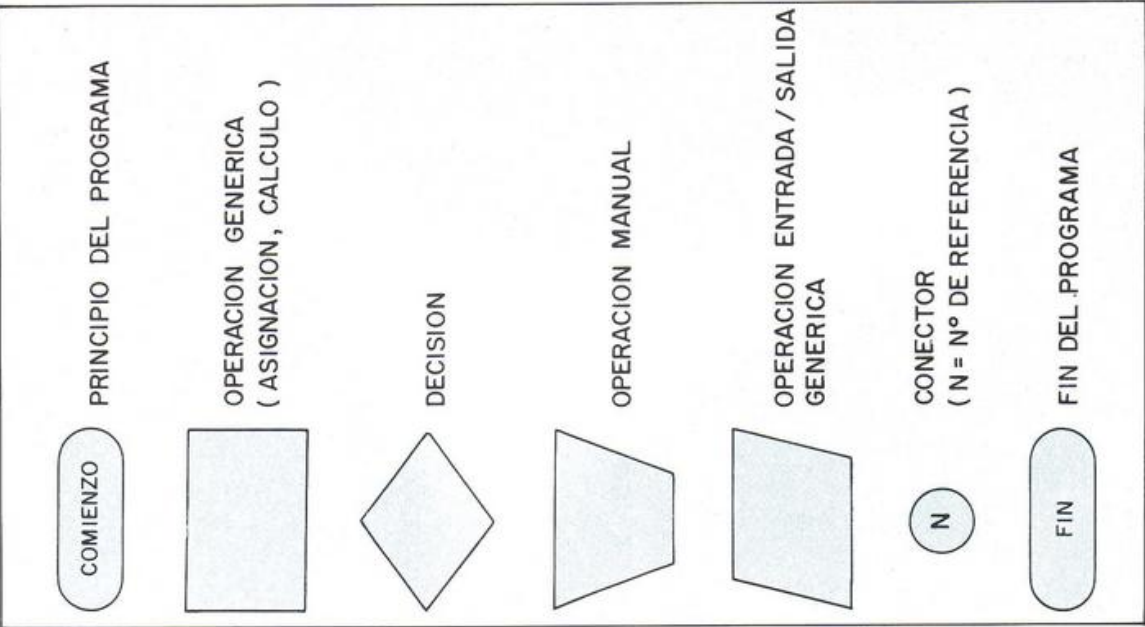
Básicamente todas las técnicas tienden a examinar globalmente el programa desde una perspectiva superior, e ir descendiendo, poco a poco, hasta llegar a elaborarlo por completo. En este proceso se pueden distinguir tres fases:

- Análisis del programa
- Síntesis.
- Representación gráfica.

Análisis

El primer paso es analizar el problema y determinar las funciones de nuestro programa, es decir, lo que deseamos que haga. Una vez conocido esto, podremos averiguar cuáles son los datos que debemos proporcionar al programa. Veamos un ejemplo.

Supongamos que deseamos confeccionar un programa que visualice en pantalla o nos saque por impresora un listado ordenado de nuestra agenda telefónica, ¿cuáles van a ser los datos que vamos a necesitar?, en principio los «nombres» y los «teléfonos» de nuestras amistades, con esta información se confecciona un boceto similar al de la figura.



Simbología utilizada en los diagramas de flujo.

ENTRADA	PROCESO	SALIDA
INTRODUCCION DE DATOS POR TECLADO	ALMACENAMIENTO EN MEMORIA	GRABACION EN CINTA

EDITAR UN NUEVO «LISTIN TELEFONICO

ENTRADA	PROCESO	SALIDA
LECTURA DE DATOS EN CINTA	MODIFICACION	NUEVA GRABACION

MODIFICACION DEL «LISTIN»

ENTRADA	PROCESO	SALIDA
LECTURA DE DATOS EN CINTA	ORDENACION DE DATOS	VISUALIZACION O IMPRESION

VISUALIZACION DEL «LISTIN»

ENTRADA	PROCESO	SALIDA
INTRODUCCION DE LA OPCION	SELECCION	PRESENTACION DE LA OPCION

MENU DE OPCIONES

Síntesis.

En un paso siguiente podremos determinar los programas que vamos a necesitar; siguiendo con nuestro ejemplo necesitaremos:

[illegible]

12 BORDER 2: PAPER 2: INK 6: C
14 PRINT FLASH 1; AT 7, 10; "PARE
LA CINTA".
16 PLOT 104, 56: DRAW 16, 31: DR
18 PLOT 15, 0: DRAW 15, -31: DR
20 PLOT 128, 71: DRAW 15, 0: DR
22 PLOT 96, 80: DRAW 15, 0: DR
24 PLOT 19, 0: DRAW 7, -11: DR
26 PLOT 15, 0: DRAW -17, 17: DR
28 PLOT 144, 72: DRAW 24, 0: DR
30 PLOT 0, -41: DRAW -80, 0

```

24 PRINT INVERSE 1; AT 16,13;"
ISTIN"
26 PLOT 0,0: DRAW 0,175: DRAW
255,0: DRAW 0,-175: DRAW -255,0
27 PLOT 0,152: DRAW 255,0
29 RESTORE
30 PRINT AT 1,1;" ";
31 FOR X=1 TO 28
32 READ ch;
34 PRINT CHR$ ch;: BEEP 0.05,
ch/2
36 NEXT X
38 DATA 127,32,77,73,67,82,79,
72,79,66,66,89,32,38,32,82,65,70,
65,69,76,32,80,82,65,66,69,83
"
40 PRINT AT 7,10;
42 PRINT #0; AT 1,1;"Pulse una
tecla para
44 PAUSE 4
46 BEEP 0.2,20

```

```

100 REM
** ** ** ** ** ** ** *
** ** ** ** ** *
** DEFINICIONES **
** ** ** ** *
** ** ** ** *
110 BORDER 1: PAPER 1: INK 7: C
L$
120 DIM N$(20,22)
130 PIP=33609
140 LET ELQSS=23658
145 LET dicos=0
150 LET ordena=0
160 GO TO 500 REM

```

...nombres y teléfonos que de-
seamos almacenar.

Una vez elegido, el programa nos pide la introducción del primer nombre, y a continuación el teléfono; recuerde que debe pulsar «ENTER» cada vez que introduzca un dato. El nombre debe constar como máximo de 15 caracteres, y estos sólo pueden ser o letras mayúsculas o espacios, y el teléfono debe estar constituido como máximo por 7 cifras.

Una vez introducidos, el programa nos pedirá los siguientes y así hasta que se completen el listín, que puede estar formado por 20 nombres como máximo. Si no desea rellenar todo el listín, cuando no tenga más nombres para introducir tecleea palabra FIN.

Posteriormente se nos presenta la opción de sacar un listado por impresora y, a continuación, el programa da las instrucciones necesarias para grabar y verificar una cinta con los datos editados.

Con la opción 2 se puede modificar una cinta ya editada. Lo primero que hace es leer la cinta con los datos. A continuación se nos presenta otro menú con tres opciones, una para poder modificar un nombre ya existente, para lo cual nos pide el número de línea que deseamos modificar. Otra opción permite introducir nuevos nombres, y por último, la tercera imprime un listado y graba una cinta de la misma manera que la opción 1.

La opción 3 es la encargada de ordenar los nombres alfabéticamente y presentarlos en pantalla. Los datos son leídos de la cinta editada. También permite sacar un listado ordenado en impresora.

Para salir del programa seleccione la opción 4.

Síntesis

Las funciones de cada uno de los programas mencionados anteriormente, pueden ser esquematizados usando la técnica desarrollada por la compañía IBM y denominada HIPO, iniciales de «Hierarchy Input/Process/Output», que traducido al español significa Jerarquía Entrada/Proceso/Salida. Esta técnica consiste en definir un programa mediante tres bloques principales.

- Entrada: definición de los datos a utilizar.

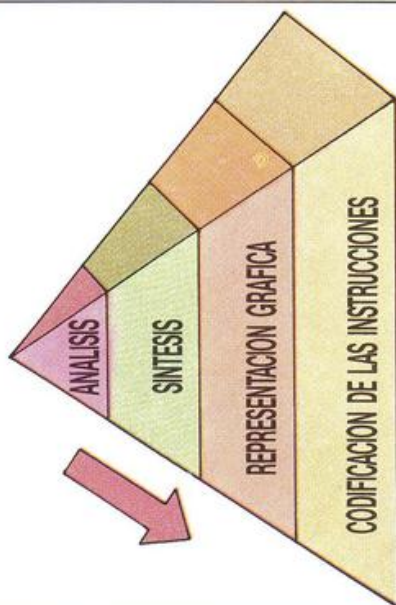
— Proceso: descripción esquematizada de los procesos o cálculos a realizar.

— Salida: especificación de los datos a imprimir o visualizar.

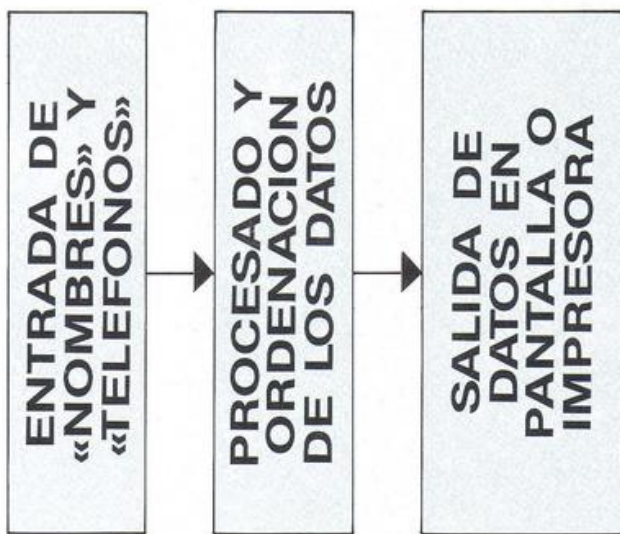
En fases posteriores cada bloque se desarrolla en otros del mismo tipo, más completos, pero todavía sin llegar al detalle.

Representación gráfica

Es el último paso antes de la edición de un programa. Es-



Jerarquía «top-down» o descendente.



Fase de análisis.


```
*****
* NUEVO LISTIN *
* *****
```

```
206 CLS
208 DIM N$(20,22)
210 FOR Y=1 TO 20
220 GO SUB 900
230 IF D$="FIN" THEN GO TO 350
240 LET N$(Y,1 TO 15)=D$
250 PRINT D$
260 GO SUB 920
270 LET N$(Y,16 TO 22)=D$
280 NEXT Y
290 GO SUB 5000
300 CLS
310 GO SUB 5200
320 LET datos=1
330 LET ordena=0
340 GO TO 500
350 REM
```

```
*****
* OPCIONES *
* *****
```



Carátula del listin.

La representación utiliza una serie de símbolos standard para facilitar su análisis por otras personas. El gráfico resultante se denomina «Diagrama de flujos», traducción de la palabra inglesa «Flow chart».

Los símbolos utilizados indican las operaciones que de-

invertdo) realiza una comparación entre dos o más valores y, en función del resultado, elige una de las dos salidas posibles. La operación manual se refiere a toda tarea que el ordenador no pueda efectuar y que se necesite de una persona para realizarla, por ejemplo, poner en funcionamiento una impresora, introducir una cinta para su lectura o grabación, etc.

La operación genérica entrada/salida representa a cualquier función de lectura, grabación, impresión, etc., es decir, de entrada o salida de datos. Cuando el diagrama es bastante grande o complejo y se dibuja en varias hojas, la unión de éstas se realiza con el símbolo denominado conector, dentro de éste debe figurar un número para utilizarlo como referencia.

Dentro de los símbolos debe incluirse una descripción corta y clara de la operación, cálculo, decisión, etc. que realizan. Los símbolos deben unirse mediante líneas con unas flechas que indiquen la dirección de ejecución del programa.

Programa

El programa «LISTIN» (número 4) realiza las funciones expresadas en los diagramas de flujo que acompañan este artículo. Una vez editado gráfico en cinta con el siguiente comando directo:

```
SAVE "LISTIN" LINE 10
```

de esta forma se autoejecuta al cargarlo.

Después de la presentación aparece un menú con diversas opciones. La número 1 permite crear un fichero con los

```
*****
```

```
502 POKE PIP,100
505 POKE FLAGS,8
506 CLS
510 PRINT AT 4,10;"OPCIONES"
520 PRINT AT 8,5,1;"- NUEVO LIS"
530 PRINT AT 12,5,2;"- MODIF. L"
540 PRINT AT 16,5,3;"- LISTIN"
550 PRINT AT 20,5,4;"- FIN"
560 PAUSE 0.1; LET P$=INKEY$
570 IF P$="1" THEN BEEP 0.05,20
580 IF P$="2" THEN BEEP 0.05,20
590 IF P$="3" THEN BEEP 0.05,20
600 IF P$="4" THEN BEEP 0.05,20
610 GO TO 550
620 CLS
630 POKE PIP,0
640 BEEP 0.2,-15; GO TO 550
```

```
650 REM
*****
* MODIFICACION *
* *****
```

```
660 IF datos=1 THEN GO TO 590
```

```
670 GO SUB 5500
680 LET datos=1
690 CLS
700 LET ordena=0
710 FOR X=1 TO ultimo
720 GO SUB 2000
730 IF X=1 THEN PRINT " "
740 PRINT "N$(X,1 TO 15); "
750 NEXT X
760 PRINT " "
770 IF P$="M" THEN MODIFICA
780 IF P$="H" THEN BEEP 0.05,20
790 GO TO 800
800 IF P$="I" THEN BEEP 0.05,20
810 IF P$="F" THEN BEEP 0.05,20
820 BEEP 0.2,-15; GO TO 780
830 INPUT "Que linea? "; LINE
```

```
840 IF P$=" " THEN GO TO 800
850 IF P$="2" THEN BEEP 0.2,-15; GO TO 800
860 FOR X=1 TO LEN P$
870 IF P$(X) < "0" OR P$(X) > "9" THEN BEEP 0.2,-15; GO TO 800
880 NEXT X
890 LET linea=VAL (P$)
900 IF linea < 1 OR linea > ultimo THEN BEEP 0.2,-15; GO TO 800
910 GO SUB 900
920 LET N$(linea,1 TO 15)=D$
930 PRINT AT linea-1,3,N$(linea,1 TO 15);
940 GO SUB 920
950 LET N$(linea,16 TO 22)=D$
960 PRINT " ";N$(linea,16 TO 22)
970 GO TO 770
980 IF ultimo=20 THEN GO TO 850
990 LET ultimo=ultimo+1
1000 PRINT #0,AT 0,0;"
```

```
1010 GO SUB 900
1020 IF ultimo<10 THEN PRINT AT ultimo-1,0;" "
1030 PRINT AT ultimo-1,0;ultimo;
1040 GO TO 770
1050 PRINT N$(ultimo,1 TO 15);
1060 GO SUB 920
1070 LET N$(ultimo,16 TO 22)=D$
1080 PRINT N$(ultimo,16 TO 22)
1090 GO TO 770
```

```
850 BEEP 0.2,-15
855 PRINT #0,AT 0,0;"No puede i"
860 FOR T=1 TO 200: NEXT T
870 GO TO 770
880 PRINT #0,AT 0,0;"
891 GO SUB 5000
892 CLS
894 GO SUB 5200
896 GO TO 500
900 REM
```

```
*****
* ENTRADA NOMBRE *
* *****
```

```
901 INPUT "NOMBRE: "; LINE D$
902 IF D$="" THEN GO TO 900
903 GO TO 900
904 IF D$>15 THEN BEEP 0.2,-15; GO TO 900
905 FOR X=1 TO LEN D$
906 IF D$(X) < "A" AND D$(X) > "Z" THEN BEEP 0.2,-15; GO TO 900
907 NEXT X
908 GO TO 900
909 RETURN
910 REM
```

```
*****
* ENTRADA TELEF. *
* *****
```

```
921 INPUT "TELEFONO: "; LINE D$
922 IF D$="" THEN GO TO 920
923 IF LEN D$>7 THEN BEEP 0.2,-15; GO TO 920
924 FOR X=1 TO LEN D$
925 IF D$(X) < "0" OR D$(X) > "9" THEN BEEP 0.2,-15; GO TO 920
926 NEXT X
927 RETURN
```

```
1000 REM
```

```
*****
* LISTIN *
* *****
```

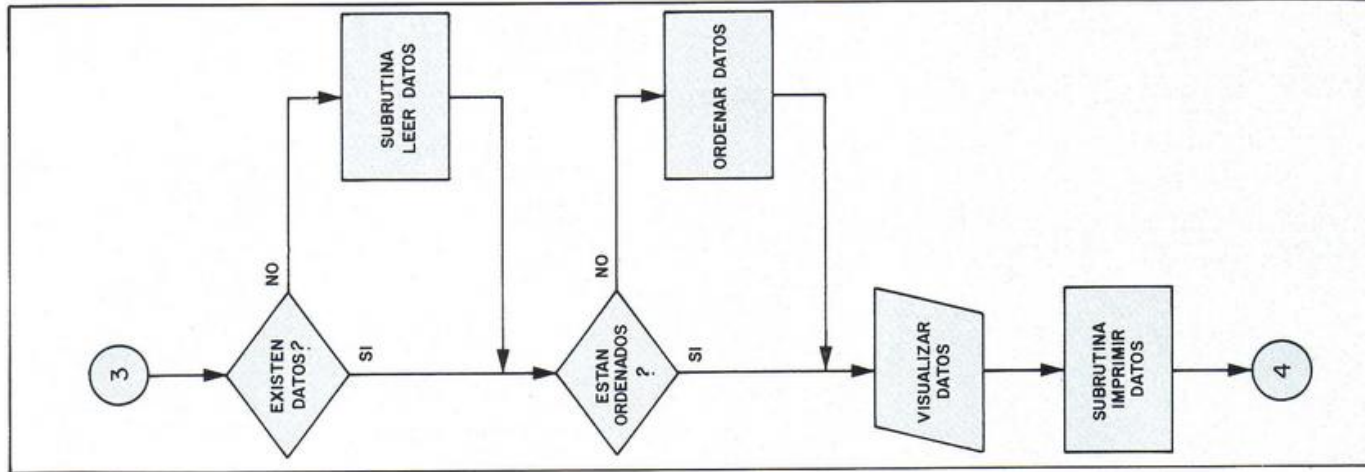
```
1002 IF datos=1 THEN GO TO 1015
1010 GO SUB 5500
1015 CLS
1020 IF ordena=1 THEN GO TO 1060
1030 PRINT AT 10,8;"ESPERE UN MOMENTO"
1040 GO SUB 2000
1050 DIM C$(20,22)
1060 LET C$(1)=N$(1)
1070 FOR X=2 TO ultimo
1080 IF N$(X) < C$(X-1) THEN GO TO 1054
1090 LET C$(X)=N$(X); GO TO 1054
1100 FOR Z=X-1 TO 1 STEP -1
1110 IF N$(X) < C$(Z) THEN GO TO 1054
1120 NEXT Z
1130 LET C$(X)=N$(X)
1140 NEXT X
1150 GO SUB 5000
1160 GO TO 500
1170 REM
```

```
*****
* ULTIMO *
* *****
```

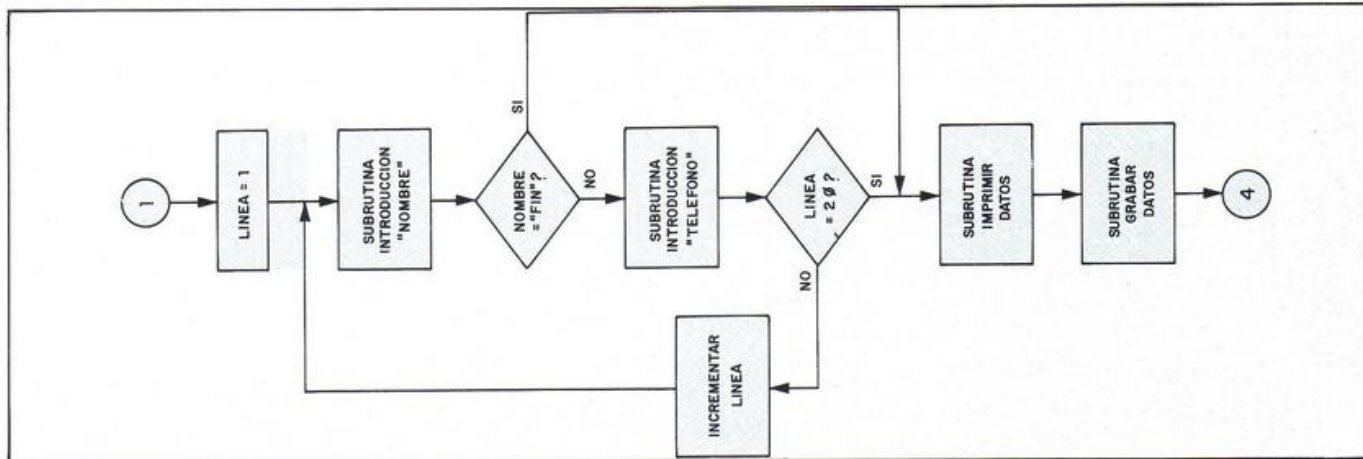

impresión de alguno de los comandos de entrada/salida; también, por ejemplo, se puede conocer si un punto determinado de la pantalla tiene color de *tinta* o *papel*.

Grabar y leer datos de un fichero.	Auxiliares
Mover datos entre ficheros.	Con estas sentencias se puede controlar el formato de

Con estas sentencias se puede controlar el formato de



Visualización del «listin telefónico» ordenado.



Editar un nuevo «listin telefónico».

expansión trasero. En ésta se pueden obtener:

- Listado de programas.
- Copia de los gráficos presentes en pantalla.
- Mensajes.

INTERFACE - 1

El INTERFACE - 1 es un dispositivo que también se adapta al conector de expansión del Spectrum. Con el manejo de ciertas instrucciones y a través de este interface se puede controlar:

- Un máximo de ocho Microdrive (unidades de almacenamiento).
 - La red local (mediante la cual se pueden atender hasta 64 Spectrum).
 - Salida RS-232 (para conectar periféricos que utilicen este sistema de transmisión de datos en serie).
- También prolonga el conector de expansión de manera que se puedan añadir más periféricos, aunque esté conectado el INTERFACE - 1.

Manejo Microdrive

A través del INTERFACE - 1 se pueden realizar las siguientes operaciones con los Microdrive:

- Formatear o inicializar cartuchos.
- Almacenar y verificar programas en cartucho.
- Cargar programas.
- Borrar ficheros de datos y programas.
- Combinar programas residentes en memoria con los archivados en cartucho.
- Catalogar cartuchos, es decir, obtener un directorio o listado de los programas contenidos en él.
- Abrir y cerrar ficheros de datos.

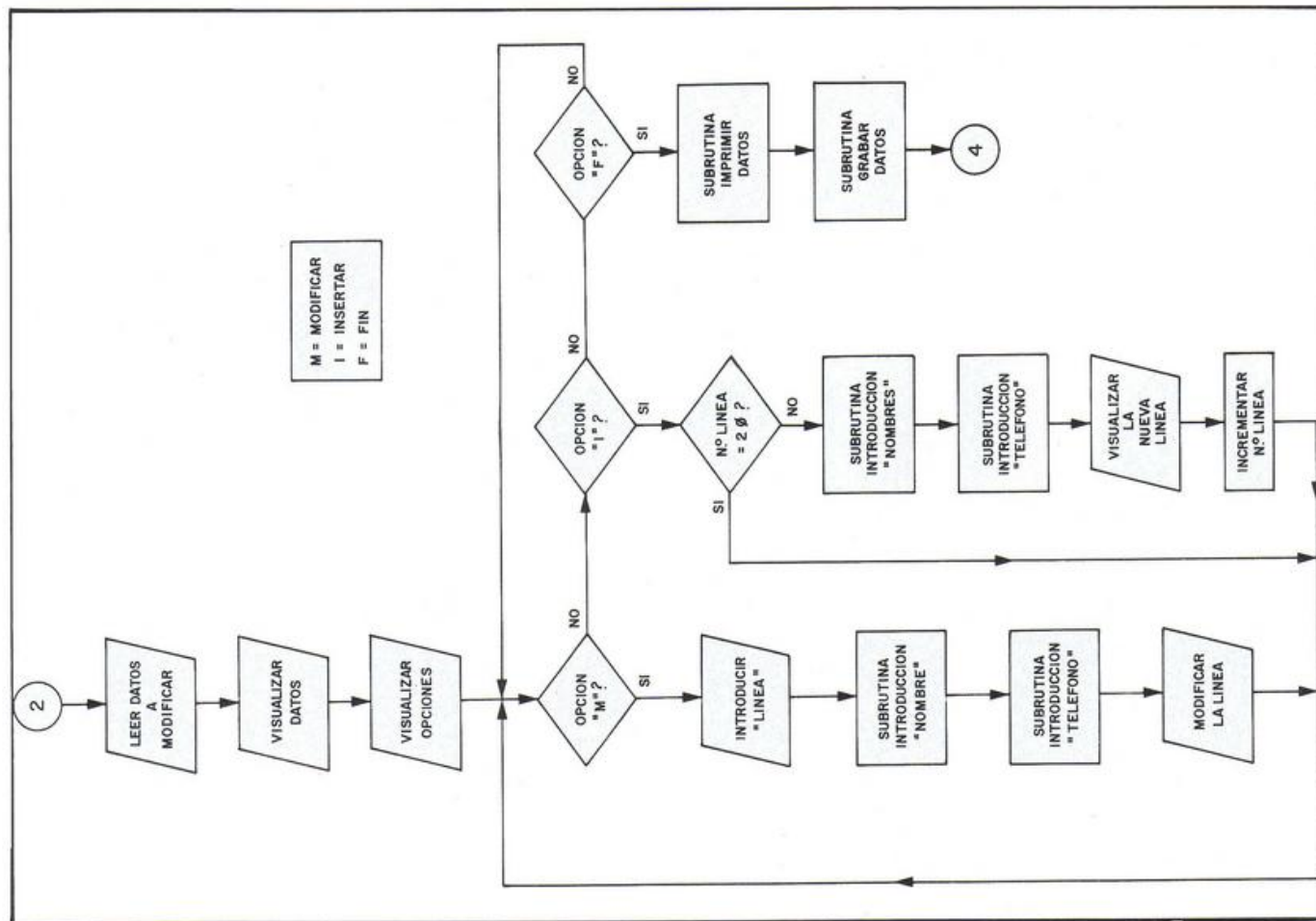
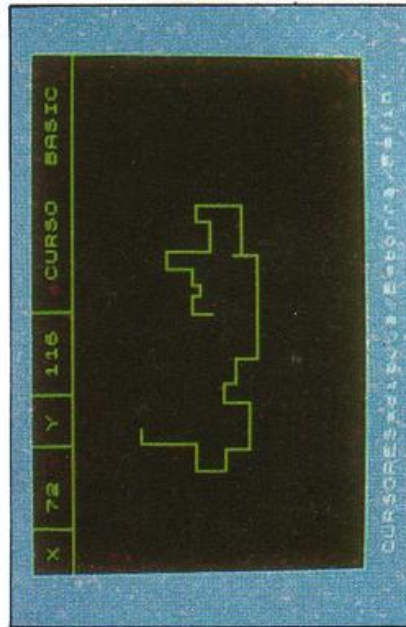
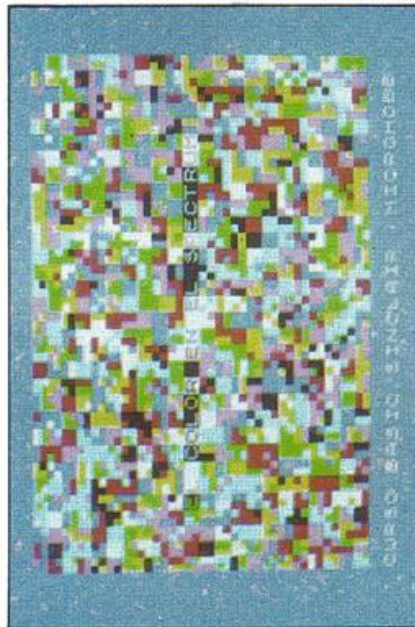
```

500 FOR y=175 TO 0 STEP -2
510 PLOT OVER 1;0,Y
520 DRAW OVER 1;255,0
530 NEXT y
540 GO SUB 750
600 FOR x=0 TO 255 STEP 1
610 PLOT OVER 1;x,0
620 DRAW OVER 1;0,175
630 NEXT x
640 GO SUB 750
700 GO TO 100
740 REM

*****
* RETARDO *
*****

750 FOR z=0 TO 300
760 NEXT z
770 RETURN

```



Modificar un «listin telefónico» ya existente.


```

10 REM
**
**
**      CURSO
**
**      BASIC/SINCLAIR
**
**      GRAFICAS
**
**
**
**
15 BORDER 2: PAPER 2: INK 7: C
LS
20 DRAW 255,0: DRAW 0,175: DRA
W -255,0: DRAW 0,-175
25 PRINT AT 5,7;"PROGRAMA ""GR
AFICAS""
30 PRINT FLASH 1; AT 12,9;"PARE
LA CINTA"
35 PLOT 0,23: DRAW 255,0
40 PRINT AT 20,1;" ";
42 RESTORE
43 FOR X=1 TO 28
50 READ LOGO
55 PRINT CHR$ LOGO;: BEEP 0.05
, LOGO/2
60 NEXT X: GO SUB 750
65 DATA 127,32,77,73,67,82,79,
72,79,66,66,89,32,38,32,82,65,70
,65,69,76,32,80,82,65,68,69,83
70 PRINT AT 12,9;"
75 PRINT #0; AT 1,1;"Pulse una
tecla para continuar"
80 PAUSE 0
85 REM
**
** DIBUJO
**
**
90 BORDER 1: PAPER 1: INK 6: C
LS
100 FOR X=-127 TO 128 STEP 2
110 PLOT OVER 1; 127,0
120 DRAW OVER 1; X,175
130 NEXT X
140 GO SUB 750
200 FOR X=-127 TO 128 STEP 2
210 PLOT OVER 1; 127,175
220 DRAW OVER 1; -X,-175
230 NEXT X
240 GO SUB 750
300 FOR Y=88 TO -87 STEP -2
310 PLOT OVER 1; 0,87
320 DRAW OVER 1; 255,Y
330 NEXT Y
340 GO SUB 750
400 FOR Y=88 TO -87 STEP -2
410 PLOT OVER 1; 255,87
420 DRAW OVER 1; -255,Y
430 NEXT Y
440 GO SUB 750

```

Pueden definirse hasta ocho colores básicos con estos comandos, aunque pueden simularse hasta 56 más. Con ellos se puede alterar el color de:

- El borde de la pantalla (zona donde no se imprimen caracteres).
- El fondo, también conocido como *papel* (zona en la cual se visualizan los caracteres o los gráficos).
- El color del propio carácter o gráfico, también conocido como *tinta*.

Estos colores pueden ser controlados en dos gamas de brillo.

La impresión se puede variar con otra serie de comandos que controlan:

- El parpadeo de los caracteres.
- La inversión del color de tinta por el de papel.
- La sobreimpresión de caracteres.

Sonido

El Spectrum tiene, con el empleo de una sola instrucción, capacidad para producir una amplia variedad de notas musicales; esta posibilidad es bastante interesante, ya que nos permite, con su uso, amenizar los programas. Estos sonidos son escuchados a través del altavoz interno, aunque también es posible enchufar un amplificador a los conectores EAR o MIC para incrementar su volumen.

Manejo impresora

Puede incorporarse una impresora ZX u otra, con el *interface* adecuado, al conector de

```

**
**
**      a cinta en el cassettey graba los
**      datos.
2200 FOR X=1 TO 400: NEXT X
2225 PRINT #0; AT 0,0;
2230 SAVE "telefono" DATA N$( )
2240 PRINT #0; AT 0,0; Grabacion
2250 FOR X=1 TO 200: NEXT X
2260 PRINT #0; AT 0,0; Rebobine l
a cinta (verificación)
2270 VERIFY "telefono" DATA N$( )
2280 CLS
2290 PRINT #0; AT 0,0; Grabacion
correcta
2300 FOR X=1 TO 200: NEXT X
2305 POKE FLAG2,8
2310 PRINT #0; AT 0,0; Desea grab
ar otra cinta (S/N)
2320 PAUSE 0: LET P$=INKEY$
2330 IF P$="S" THEN BEEP 0.05,20
GO TO 2200
2340 IF P$="N" THEN BEEP 0.05,20
CLS
2350 BEEP 0.2,-15: GO TO 5320
5500 REM
*****
**
** LECTURA DE
**
** DATOS
**
*****
5505 CLS
5510 PRINT #0; AT 0,0; Coloque la
cinta con los datos.
5520 LOAD "telefono" DATA N$( )
5530 CLS
5540 PRINT #0; AT 0,0; Lectura co
rrecta.
5550 FOR X=1 TO 200: NEXT X
5560 RETURN

```


El juego de sentencias

Después de haber repasado en capítulos anteriores los conocimientos esenciales del lenguaje de programación BASIC, empezamos en éste a dar una visión global del juego de sentencias del Spectrum; posteriormente irán siendo estudiadas una a una.

Clasificación

El juego de comandos, sentencias y funciones, consta de 88 palabras claves, éstas pueden clasificarse, según su funcionalidad, de la siguiente forma:

- Comandos de control.
- Comandos de programación.
- Comandos de entrada/salida.
- Manejo de cadenas.
- Funciones aritméticas.
- Funciones lógicas.
- Comandos de dibujo.
- Comandos de control de color e impresión.
- Sonido.
- Manejo impresora.
- Manejo periférico (Interfase - 1).
- Auxiliares.

Comandos de control

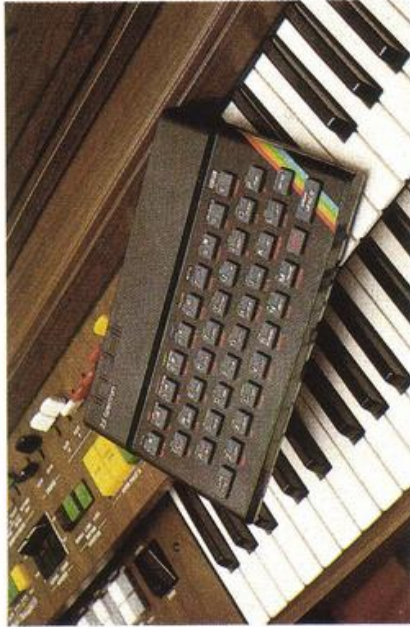
Esta serie de comandos normalmente son introducidos de forma directa en el ordenador, es decir, que no forman parte de las instrucciones de un programa, aunque en algunas ocasiones sí pueden formar.

- Algunas de las funciones de estos comandos son:
- Ejecución de un programa.
 - Listado por pantalla de las instrucciones.
 - Carga de programas.
 - Borrado de pantalla.
 - Borrado de memoria.
 - Etc...

Comandos de programación

La función principal de estas sentencias es, manejar la información interna, realizar cálculos, tomar decisiones, etc. Algunas de las funciones más concretas son:

- Asignación de variables.
- Dimensionado de Matrices.
- Parada en la ejecución de un programa.
- Temporizaciones.
- Control de bucles.



- Decisiones.
- Salto a una línea determinada.
- Salto a subrutina.
- Etc...

Comandos de entrada/salida

Son todos aquellos que indican al ordenador qué tipo de operación de lectura o escritura de datos debe realizar, como por ejemplo:

- Visualización en pantalla.
- Entrada de datos por teclado.
- Reconocimiento de la última tecla pulsada.
- Lectura de los datos de una tabla.
- Escritura en una posición de memoria.
- Lectura de una posición de memoria.
- Entrada o salida de datos por el conector trasero.

Manejo de las cadenas

- útiles en el manejo de cadenas como:
- Conocer la longitud de una cadena.
 - Asignar a una variable numérica el valor de una cadena.
 - Pasar a decimal el valor de un carácter ASCII.
 - Pasar a Código ASCII un número decimal.
 - Etc...

Funciones aritméticas

Con esta serie de funciones, el Spectrum se convierte en una útil calculadora científica, ya que permite efectuar las siguientes operaciones:

- Raíz cuadrada.
- Logaritmos naturales.
- Seno.
- Coseno.
- Tangente.
- Arco seno.
- Arco coseno.
- Arco tangente.
- Generación de números aleatorios.
- Conversión de binario a decimal.
- Conocer el signo de una expresión.
- Averiguar su valor absoluto o su parte entera.
- Operar con las constantes matemáticas «pi» y «e».

Funciones lógicas

Al estudiar los operadores lógicos, fueron repasadas estas funciones:

- AND.
- OR.
- NOT.

Comandos de dibujo

Con estas sentencias se puede realizar cualquier tipo de gráfico en alta resolución. Se puede dibujar mediante:

PROGRAMA 1

```

10 REM
   *****
   *          *
   *      CURSO          *
   * *****
   * BASIC/SINCLAIR      *
   * *****
   * "COLOREAR"          *
   * *****
   * *****
20 BORDER 1: PAPER 1: CLS
30 PRINT #0; AT 1,0;"CURSO BASI
C/SINCLAIR MICROHOBBY"
40 FOR Y=0 TO 21
50 FOR X=0 TO 31
60 LET grafico=INT (RND*15)+12
70 LET color papel=RND*7
80 LET color tinta=RND*7
90 PRINT AT Y,X; PAPER color P
apel; INK color tinta;CHR$ grafi
CO
100 NEXT X
110 LET color borde=RND*7
120 BORDER color borde
130 NEXT Y
140 PRINT AT 10,4; PAPER 9; INK
8;"EL COLOR EN EL SPECTRUM";
150 PAUSE 0
160 BORDER 7: PAPER 7: INK 0: C
LS

```



- Puntos.
 - Líneas.
 - Circunferencias.
 - Arcos de circunferencia.
- Los argumentos usuales que hay que indicar en este tipo de sentencias son las coordenadas «X» e «Y» de la posición a dibujar.
- También existe la posibilidad de realizar gráficos, utilizando los *predefinidos*, que incorpora el Spectrum, y los *definidos* por el usuario (GDU).

Ejemplos:

```
10 PRINT "PROGRAMA 4"
20 PRINT "DIBUJANDO"
30 PRINT "CURSO"
40 PRINT "DIBUJANDO"
50 PRINT "CURSO"
60 PRINT "DIBUJANDO"
70 PRINT "CURSO"
80 PRINT "DIBUJANDO"
90 PRINT "CURSO"
100 PRINT "DIBUJANDO"
```

- c) Visualizar variables numéricas previamente asignadas.

SENTENCIA	ARGUMENTO
PRINT	Variable

Ejemplos:

```
10 LET A=20
20 LET B=30
30 LET C=40
40 LET D=50
50 LET E=60
60 LET F=70
70 LET G=80
80 LET H=90
90 LET I=100
100 LET J=110
110 LET K=120
120 LET L=130
130 LET M=140
140 LET N=150
150 LET O=160
160 LET P=170
170 LET Q=180
180 LET R=190
190 LET S=200
200 LET T=210
210 LET U=220
220 LET V=230
230 LET W=240
240 LET X=250
250 LET Y=260
260 LET Z=270
270 LET AA=280
280 LET AB=290
290 LET AC=300
300 LET AD=310
310 LET AE=320
320 LET AF=330
330 LET AG=340
340 LET AH=350
350 LET AI=360
360 LET AJ=370
370 LET AK=380
380 LET AL=390
390 LET AM=400
400 LET AN=410
410 LET AO=420
420 LET AP=430
430 LET AQ=440
440 LET AR=450
450 LET AS=460
460 LET AT=470
470 LET AU=480
480 LET AV=490
490 LET AW=500
500 LET AX=510
510 LET AY=520
520 LET AZ=530
530 LET BA=540
540 LET BB=550
550 LET BC=560
560 LET BD=570
570 LET BE=580
580 LET BF=590
590 LET BG=600
600 LET BH=610
610 LET BI=620
620 LET BJ=630
630 LET BK=640
640 LET BL=650
650 LET BM=660
660 LET BN=670
670 LET BO=680
680 LET BP=690
690 LET BQ=700
700 LET BR=710
710 LET BS=720
720 LET BT=730
730 LET BU=740
740 LET BV=750
750 LET BW=760
760 LET BX=770
770 LET BY=780
780 LET BZ=790
790 LET CA=800
800 LET CB=810
810 LET CC=820
820 LET CD=830
830 LET CE=840
840 LET CF=850
850 LET CG=860
860 LET CH=870
870 LET CI=880
880 LET CJ=890
890 LET CK=900
900 LET CL=910
910 LET CM=920
920 LET CN=930
930 LET CO=940
940 LET CP=950
950 LET CQ=960
960 LET CR=970
970 LET CS=980
980 LET CT=990
990 LET CU=1000
1000 LET CV=1010
1010 LET CW=1020
1020 LET CX=1030
1030 LET CY=1040
1040 LET CZ=1050
1050 LET DA=1060
1060 LET DB=1070
1070 LET DC=1080
1080 LET DD=1090
1090 LET DE=1100
1100 LET DF=1110
1110 LET DG=1120
1120 LET DH=1130
1130 LET DI=1140
1140 LET DJ=1150
1150 LET DK=1160
1160 LET DL=1170
1170 LET DM=1180
1180 LET DN=1190
1190 LET DO=1200
1200 LET DP=1210
1210 LET DQ=1220
1220 LET DR=1230
1230 LET DS=1240
1240 LET DT=1250
1250 LET DU=1260
1260 LET DV=1270
1270 LET DW=1280
1280 LET DX=1290
1290 LET DY=1300
1300 LET DZ=1310
1310 LET EA=1320
1320 LET EB=1330
1330 LET EC=1340
1340 LET ED=1350
1350 LET EE=1360
1360 LET EF=1370
1370 LET EG=1380
1380 LET EH=1390
1390 LET EI=1400
1400 LET EJ=1410
1410 LET EK=1420
1420 LET EL=1430
1430 LET EM=1440
1440 LET EN=1450
1450 LET EO=1460
1460 LET EP=1470
1470 LET EQ=1480
1480 LET ER=1490
1490 LET ES=1500
1500 LET ET=1510
1510 LET EU=1520
1520 LET EV=1530
1530 LET EW=1540
1540 LET EX=1550
1550 LET EY=1560
1560 LET EZ=1570
1570 LET FA=1580
1580 LET FB=1590
1590 LET FC=1600
1600 LET FD=1610
1610 LET FE=1620
1620 LET FF=1630
1630 LET FG=1640
1640 LET FH=1650
1650 LET FI=1660
1660 LET FJ=1670
1670 LET FK=1680
1680 LET FL=1690
1690 LET FM=1700
1700 LET FN=1710
1710 LET FO=1720
1720 LET FP=1730
1730 LET FQ=1740
1740 LET FR=1750
1750 LET FS=1760
1760 LET FT=1770
1770 LET FU=1780
1780 LET FV=1790
1790 LET FW=1800
1800 LET FX=1810
1810 LET FY=1820
1820 LET FZ=1830
1830 LET GA=1840
1840 LET GB=1850
1850 LET GC=1860
1860 LET GD=1870
1870 LET GE=1880
1880 LET GF=1890
1890 LET GG=1900
1900 LET GH=1910
1910 LET GI=1920
1920 LET GJ=1930
1930 LET GK=1940
1940 LET GL=1950
1950 LET GM=1960
1960 LET GN=1970
1970 LET GO=1980
1980 LET GP=1990
1990 LET GQ=2000
2000 LET GR=2010
2010 LET GS=2020
2020 LET GT=2030
2030 LET GU=2040
2040 LET GV=2050
2050 LET GW=2060
2060 LET GX=2070
2070 LET GY=2080
2080 LET GZ=2090
2090 LET HA=2100
2100 LET HB=2110
2110 LET HC=2120
2120 LET HD=2130
2130 LET HE=2140
2140 LET HF=2150
2150 LET HG=2160
2160 LET HH=2170
2170 LET HI=2180
2180 LET HJ=2190
2190 LET HK=2200
2200 LET HL=2210
2210 LET HM=2220
2220 LET HN=2230
2230 LET HO=2240
2240 LET HP=2250
2250 LET HQ=2260
2260 LET HR=2270
2270 LET HS=2280
2280 LET HT=2290
2290 LET HU=2300
2300 LET HV=2310
2310 LET HW=2320
2320 LET HX=2330
2330 LET HY=2340
2340 LET HZ=2350
2350 LET IA=2360
2360 LET IB=2370
2370 LET IC=2380
2380 LET ID=2390
2390 LET IE=2400
2400 LET IF=2410
2410 LET IG=2420
2420 LET IH=2430
2430 LET II=2440
2440 LET IJ=2450
2450 LET IK=2460
2460 LET IL=2470
2470 LET IM=2480
2480 LET IN=2490
2490 LET IO=2500
2500 LET IP=2510
2510 LET IQ=2520
2520 LET IR=2530
2530 LET IS=2540
2540 LET IT=2550
2550 LET IU=2560
2560 LET IV=2570
2570 LET IW=2580
2580 LET IX=2590
2590 LET IY=2600
2600 LET IZ=2610
2610 LET JA=2620
2620 LET JB=2630
2630 LET JC=2640
2640 LET JD=2650
2650 LET JE=2660
2660 LET JF=2670
2670 LET JG=2680
2680 LET JH=2690
2690 LET JI=2700
2700 LET JJ=2710
2710 LET JK=2720
2720 LET JL=2730
2730 LET JM=2740
2740 LET JN=2750
2750 LET JO=2760
2760 LET JP=2770
2770 LET JQ=2780
2780 LET JR=2790
2790 LET JS=2800
2800 LET JT=2810
2810 LET JU=2820
2820 LET JV=2830
2830 LET JW=2840
2840 LET JX=2850
2850 LET JY=2860
2860 LET JZ=2870
2870 LET KA=2880
2880 LET KB=2890
2890 LET KC=2900
2900 LET KD=2910
2910 LET KE=2920
2920 LET KF=2930
2930 LET KG=2940
2940 LET KH=2950
2950 LET KI=2960
2960 LET KJ=2970
2970 LET KK=2980
2980 LET KL=2990
2990 LET KM=3000
3000 LET KN=3010
3010 LET KO=3020
3020 LET KP=3030
3030 LET KQ=3040
3040 LET KR=3050
3050 LET KS=3060
3060 LET KT=3070
3070 LET KU=3080
3080 LET KV=3090
3090 LET KW=3100
3100 LET KX=3110
3110 LET KY=3120
3120 LET KZ=3130
3130 LET LA=3140
3140 LET LB=3150
3150 LET LC=3160
3160 LET LD=3170
3170 LET LE=3180
3180 LET LF=3190
3190 LET LG=3200
3200 LET LH=3210
3210 LET LI=3220
3220 LET LJ=3230
3230 LET LK=3240
3240 LET LL=3250
3250 LET LM=3260
3260 LET LN=3270
3270 LET LO=3280
3280 LET LP=3290
3290 LET LQ=3300
3300 LET LR=3310
3310 LET LS=3320
3320 LET LT=3330
3330 LET LU=3340
3340 LET LV=3350
3350 LET LW=3360
3360 LET LX=3370
3370 LET LY=3380
3380 LET LZ=3390
3390 LET MA=3400
3400 LET MB=3410
3410 LET MC=3420
3420 LET MD=3430
3430 LET ME=3440
3440 LET MF=3450
3450 LET MG=3460
3460 LET MH=3470
3470 LET MI=3480
3480 LET MJ=3490
3490 LET MK=3500
3500 LET ML=3510
3510 LET MM=3520
3520 LET MN=3530
3530 LET MO=3540
3540 LET MP=3550
3550 LET MQ=3560
3560 LET MR=3570
3570 LET MS=3580
3580 LET MT=3590
3590 LET MU=3600
3600 LET MV=3610
3610 LET MW=3620
3620 LET MX=3630
3630 LET MY=3640
3640 LET MZ=3650
3650 LET NA=3660
3660 LET NB=3670
3670 LET NC=3680
3680 LET ND=3690
3690 LET NE=3700
3700 LET NF=3710
3710 LET NG=3720
3720 LET NH=3730
3730 LET NI=3740
3740 LET NJ=3750
3750 LET NK=3760
3760 LET NL=3770
3770 LET NM=3780
3780 LET NN=3790
3790 LET NO=3800
3800 LET NP=3810
3810 LET NQ=3820
3820 LET NR=3830
3830 LET NS=3840
3840 LET NT=3850
3850 LET NU=3860
3860 LET NV=3870
3870 LET NW=3880
3880 LET NX=3890
3890 LET NY=3900
3900 LET NZ=3910
3910 LET OA=3920
3920 LET OB=3930
3930 LET OC=3940
3940 LET OD=3950
3950 LET OE=3960
3960 LET OF=3970
3970 LET OG=3980
3980 LET OH=3990
3990 LET OI=4000
4000 LET OJ=4010
4010 LET OK=4020
4020 LET OL=4030
4030 LET OM=4040
4040 LET ON=4050
4050 LET OO=4060
4060 LET OP=4070
4070 LET OQ=4080
4080 LET OR=4090
4090 LET OS=4100
4100 LET OT=4110
4110 LET OU=4120
4120 LET OV=4130
4130 LET OW=4140
4140 LET OX=4150
4150 LET OY=4160
4160 LET OZ=4170
4170 LET PA=4180
4180 LET PB=4190
4190 LET PC=4200
4200 LET PD=4210
4210 LET PE=4220
4220 LET PF=4230
4230 LET PG=4240
4240 LET PH=4250
4250 LET PI=4260
4260 LET PJ=4270
4270 LET PK=4280
4280 LET PL=4290
4290 LET PM=4300
4300 LET PN=4310
4310 LET PO=4320
4320 LET PP=4330
4330 LET PQ=4340
4340 LET PR=4350
4350 LET PS=4360
4360 LET PT=4370
4370 LET PU=4380
4380 LET PV=4390
4390 LET PW=4400
4400 LET PX=4410
4410 LET PY=4420
4420 LET PZ=4430
4430 LET QA=4440
4440 LET QB=4450
4450 LET QC=4460
4460 LET QD=4470
4470 LET QE=4480
4480 LET QF=4490
4490 LET QG=4500
4500 LET QH=4510
4510 LET QI=4520
4520 LET QJ=4530
4530 LET QK=4540
4540 LET QL=4550
4550 LET QM=4560
4560 LET QN=4570
4570 LET QO=4580
4580 LET QP=4590
4590 LET QQ=4600
4600 LET QR=4610
4610 LET QS=4620
4620 LET QT=4630
4630 LET QU=4640
4640 LET QV=4650
4650 LET QW=4660
4660 LET QX=4670
4670 LET QY=4680
4680 LET QZ=4690
4690 LET RA=4700
4700 LET RB=4710
4710 LET RC=4720
4720 LET RD=4730
4730 LET RE=4740
4740 LET RF=4750
4750 LET RG=4760
4760 LET RH=4770
4770 LET RI=4780
4780 LET RJ=4790
4790 LET RK=4800
4800 LET RL=4810
4810 LET RM=4820
4820 LET RN=4830
4830 LET RO=4840
4840 LET RP=4850
4850 LET RQ=4860
4860 LET RR=4870
4870 LET RS=4880
4880 LET RT=4890
4890 LET RU=4900
4900 LET RV=4910
4910 LET RW=4920
4920 LET RX=4930
4930 LET RY=4940
4940 LET RZ=4950
4950 LET SA=4960
4960 LET SB=4970
4970 LET SC=4980
4980 LET SD=4990
4990 LET SE=5000
5000 LET SF=5010
5010 LET SG=5020
5020 LET SH=5030
5030 LET SI=5040
5040 LET SJ=5050
5050 LET SK=5060
5060 LET SL=5070
5070 LET SM=5080
5080 LET SN=5090
5090 LET SO=5100
5100 LET SP=5110
5110 LET SQ=5120
5120 LET SR=5130
5130 LET SS=5140
5140 LET ST=5150
5150 LET SU=5160
5160 LET SV=5170
5170 LET SW=5180
5180 LET SX=5190
5190 LET SY=5200
5200 LET SZ=5210
5210 LET TA=5220
5220 LET TB=5230
5230 LET TC=5240
5240 LET TD=5250
5250 LET TE=5260
5260 LET TF=5270
5270 LET TG=5280
5280 LET TH=5290
5290 LET TI=5300
5300 LET TJ=5310
5310 LET TK=5320
5320 LET TL=5330
5330 LET TM=5340
5340 LET TN=5350
5350 LET TO=5360
5360 LET TP=5370
5370 LET TQ=5380
5380 LET TR=5390
5390 LET TS=5400
5400 LET TT=5410
5410 LET TU=5420
5420 LET TV=5430
5430 LET TW=5440
5440 LET TX=5450
5450 LET TY=5460
5460 LET TZ=5470
5470 LET UA=5480
5480 LET UB=5490
5490 LET UC=5500
5500 LET UD=5510
5510 LET UE=5520
5520 LET UF=5530
5530 LET UG=5540
5540 LET UH=5550
5550 LET UI=5560
5560 LET UJ=5570
5570 LET UK=5580
5580 LET UL=5590
5590 LET UM=5600
5600 LET UN=5610
5610 LET UO=5620
5620 LET UP=5630
5630 LET UQ=5640
5640 LET UR=5650
5650 LET US=5660
5660 LET UT=5670
5670 LET UV=5680
5680 LET UW=5690
5690 LET UX=5700
5700 LET UY=5710
5710 LET UZ=5720
5720 LET VA=5730
5730 LET VB=5740
5740 LET VC=5750
5750 LET VD=5760
5760 LET VE=5770
5770 LET VF=5780
5780 LET VG=5790
5790 LET VH=5800
5800 LET VI=5810
5810 LET VJ=5820
5820 LET VK=5830
5830 LET VL=5840
5840 LET VM=5850
5850 LET VN=5860
5860 LET VO=5870
5870 LET VP=5880
5880 LET VQ=5890
5890 LET VR=5900
5900 LET VS=5910
5910 LET VT=5920
5920 LET VU=5930
5930 LET VV=5940
5940 LET VW=5950
5950 LET VX=5960
5960 LET VY=5970
5970 LET VZ=5980
5980 LET WA=5990
5990 LET WB=6000
6000 LET WC=6010
6010 LET WD=6020
6020 LET WE=6030
6030 LET WF=6040
6040 LET WG=6050
6050 LET WH=6060
6060 LET WI=6070
6070 LET WJ=6080
6080 LET WK=6090
6090 LET WL=6100
6100 LET WM=6110
6110 LET WN=6120
6120 LET WO=6130
6130 LET WP=6140
6140 LET WQ=6150
6150 LET WR=6160
6160 LET WS=6170
6170 LET WT=6180
6180 LET WU=6190
6190 LET WV=6200
6200 LET WX=6210
6210 LET WY=6220
6220 LET WZ=6230
6230 LET XA=6240
6240 LET XB=6250
6250 LET XC=6260
6260 LET XD=6270
6270 LET XE=6280
6280 LET XF=6290
6290 LET XG=6300
6300 LET XH=6310
6310 LET XI=6320
6320 LET XJ=6330
6330 LET XK=6340
6340 LET XL=6350
6350 LET XM=6360
6360 LET XN=6370
6370 LET XO=6380
6380 LET XP=6390
6390 LET XQ=6400
6400 LET XR=6410
6410 LET XS=6420
6420 LET XT=6430
6430 LET XU=6440
6440 LET XV=6450
6450 LET XW=6460
6460 LET XX=6470
6470 LET XY=6480
6480 LET XZ=6490
6490 LET YA=6500
6500 LET YB=6510
6510 LET YC=6520
6520 LET YD=6530
6530 LET YE=6540
6540 LET YF=6550
6550 LET YG=6560
6560 LET YH=6570
6570 LET YI=6580
6580 LET YJ=6590
6590 LET YK=6600
6600 LET YL=6610
6610 LET YM=6620
6620 LET YN=6630
6630 LET YO=6640
6640 LET YP=6650
6650 LET YQ=6660
6660 LET YR=6670
6670 LET YS=6680
6680 LET YT=6690
6690 LET YU=6700
6700 LET YV=6710
6710 LET YW=6720
6720 LET YX=6730
6730 LET YY=6740
6740 LET YZ=6750
6750 LET ZA=6760
6760 LET ZB=6770
6770 LET ZC=6780
6780 LET ZD=6790
6790 LET ZE=6800
6800 LET ZF=6810
6810 LET ZG=6820
6820 LET ZH=6830
6830 LET ZI=6840
6840 LET ZJ=6850
6850 LET ZK=6860
6860 LET ZL=6870
6870 LET ZM=6880
6880 LET ZN=6890
6890 LET ZO=6900
6900 LET ZP=6910
6910 LET ZQ=6920
6920 LET ZR=6930
6930 LET ZS=6940
6940 LET ZT=6950
6950 LET ZU=6960
6960 LET ZV=6970
6970 LET ZW=6980
6980 LET ZX=6990
6990 LET ZY=7000
7000 LET ZZ=7010
7010 LET AA=7020
7020 LET AB=7030
7030 LET AC=7040
7040 LET AD=7050
7050 LET AE=7060
7060 LET AF=7070
7070 LET AG=7080
7080 LET AH=7090
7090 LET AI=7100
7100 LET AJ=7110
7110 LET AK=7120
7120 LET AL=7130
7130 LET AM=7140
7140 LET AN=7150
7150 LET AO=7160
7160 LET AP=7170
7170 LET AQ=7180
7180 LET AR=7190
7190 LET AS=7200
7200 LET AT=7210
7210 LET AU=7220
7220 LET AV=7230
7230 LET AW=7240
7240 LET AX=7250
7250 LET AY=7260
7260 LET AZ=7270
7270 LET BA=7280
7280 LET BB=7290
7290 LET BC=7300
7300 LET BD=7310
7310 LET BE=7320
7320 LET BF=7330
7330 LET BG=7340
7340 LET BH=7350
7350 LET BI=7360
7360 LET BJ=7370
7370 LET BK=7380
7380 LET BL=7390
7390 LET BM=7400
7400 LET BN=7410
7410 LET BO=7420
7420 LET BP=7430
7430 LET BQ=7440
7440 LET BR=7450
7450 LET BS=7460
7460 LET BT=7470
7470 LET BU=7480
7480 LET BV=7490
7490 LET BW=7500
7500 LET BX=7510
7510 LET BY=7520
7520 LET BZ=7530
7530 LET CA=7540
7540 LET CB=7550
7550 LET CC=7560
7560 LET CD=7570
7570 LET CE=7580
7580 LET CF=7590
7590 LET CG=7600
7600 LET CH=7610
7610 LET CI=7620
7620 LET CJ=7630
7630 LET CK=7640
7640 LET CL=7650
7650 LET CM=7660
7660 LET CN=7670
7670 LET CO=7680
7680 LET CP=7690
7690 LET CQ=7700
7700 LET CR=7710
7710 LET CS=7720
7720 LET CT=7730
7730 LET CU=7740
7740 LET CV=7750
7750 LET CW=7760
7760 LET CX=7770
7770 LET CY=7780
7780 LET CZ=7790
7790 LET DA=7800
7800 LET DB=7810
7810 LET DC=7820
7820 LET DD=7830
7830 LET DE=7840
7840 LET DF=7850
7850 LET DG=7860
7860 LET DH=7870
7870 LET DI=7880
7880 LET DJ=7890
7890 LET DK=7900
7900 LET DL=7910
7910 LET DM=7920
7920 LET DN=7930
7930 LET DO=7940
7940 LET DP=7950
7950 LET DQ=7960
7960 LET DR=7970
7970 LET DS=7980
7980 LET DT=7990
7990 LET DU=8000
8000 LET DV=8010
8010 LET DW=8020
8020 LET DX=8030
8030 LET DY=8040
8040 LET DZ=8050
8050 LET EA=8060
8060 LET EB=8070
8070 LET EC=8080
8080 LET ED=8090
8090 LET EE=8100
8100 LET EF=8110
8110 LET EG=8120
8120 LET EH=8130
8130 LET EI=8140
8140 LET EJ=8150
8150 LET EK=8160
8160 LET EL=8170
8170 LET EM=8180
8180 LET EN=8190
8190 LET EO=8200
8200 LET EP=8210
8210 LET EQ=8220
8220 LET ER=8230
8230 LET ES=8240
8240 LET ET=8250
8250 LET EU=8260
8260 LET EV=8270
8270 LET EW=8280
8280 LET EX=8290
8290 LET EY=8300
8300 LET EZ=8310
8310 LET FA=8320
8320 LET FB=8330
8330 LET FC=8340
8340 LET FD=8350
8350 LET FE=8360
8360 LET FF=8370
8370 LET FG=8380
8380 LET FH=8390
8390 LET FI=8400
8400 LET FJ=8410
8410 LET FK=8420
8420 LET FL=8430
8430 LET FM=8440
8440 LET FN=8450
8450 LET FO=8460
8460 LET FP=8470
8470 LET FQ=8480
8480 LET FR=8490
8490 LET FS=8500
8500 LET FT=8510
8510 LET FU=8520
8520 LET FV=8530
8530 LET FW=8540
8540 LET FX=8550
8550 LET FY=8560
8560 LET FZ=8570
8570 LET GA=8580
8580 LET GB=8590
8590 LET GC=8600
8600 LET GD=8610
8610 LET GE=8620
8620 LET GF=8630
8630 LET GG=8640
8640 LET GH=8650
8650 LET GI=8660
8660 LET GJ=8670
8670 LET GK=8680
8680 LET GL=8690
8690 LET GM=8700
8700 LET GN=8710
8710 LET GO=8720
8720 LET GP=8730
8730 LET GQ=8740
8740 LET GR=8750
8750 LET GS=8760
8760 LET GT=8770
8770 LET GU=8780
8780 LET GV=8790
8790 LET GW=8800
8800 LET GX=8810
8810 LET GY=8820
8820 LET GZ=8830
8830 LET HA=8840
8840 LET HB=8850
8850 LET HC=8860
8860 LET HD=8870
8870 LET HE=8880
8880 LET HF=8890
8890 LET HG=8900
8900 LET HH=8910
8910 LET HI=8920
8920 LET HJ=8930
8930 LET HK=8940
8940 LET HL=8950
8950 LET HM=8960
8960 LET HN=8970
8970 LET HO=8980
8980 LET HP=8990
8990 LET HQ=9000
9000 LET HR=9010
9010 LET HS=9020
9020 LET HT=9030
9030 LET HU=9040
9040 LET HV=9050
9050 LET HW=9060
9060 LET HX=9070
9070 LET HY=9080
9080 LET HZ=9090
9090 LET IA=9100
9100 LET IB=9110
9110 LET IC=9120
9120 LET ID=9130
9130 LET IE=9140
9140 LET IF=9150
9150 LET IG=9160
9160 LET IH=9170
9170 LET II=9180
9180 LET IJ=9190
9190 LET IK=9200
9200 LET IL=9210
9210 LET IM=9220
9220 LET IN=9230
9230 LET IO=9240
9240 LET IP=9250
9250 LET IQ=9260
9260 LET IR=9270
9270 LET IS=9280
9280 LET IT=9290
9290 LET IU=9300
9300 LET IV=9310
9310 LET IW=9320
9320 LET IX=9330
9330 LET IY=9340
9340 LET IZ=9350
9350 LET JA=9360
9360 LET JB=9370
9370 LET JC=9380
9380 LET JD=9390
9390 LET JE=9400
9400 LET JF=9410
9410 LET JG=9420
9420 LET JH=9430
9430 LET JI=9440
9440 LET JJ=9450
9450 LET JK=9460
9460 LET JL=9470
9470 LET JM=9480
9480 LET JN=9490
9490 LET JO=9500
9500 LET JP=9510
9510 LET JQ=9520
9520 LET JR=9530
9530 LET JS=9540
9540 LET JT=9550
9550 LET JU=9560
9560 LET JV=9570
9570 LET JW=9580
9580 LET JX=9590
9590 LET JY=9600
9600 LET JZ=9610
9610 LET KA=9620
9620 LET KB=9630
9630 LET KC=9640
9640 LET KD=9650
9650 LET KE=9660
9660 LET KF=9670
9670 LET KG=9680
9680 LET KH=9690
9690 LET KI=9700
9700 LET KJ=9710
9710 LET KK=9720
9720 LET KL=9730
9730 LET KM=9740
9740 LET KN=9750
9750 LET KO=9760
9760 LET KP=9770
9770 LET KQ=9780
9780 LET KR=9790
9790 LET KS=9800
9800 LET KT=9810
9810 LET KU=9820
9820 LET KV=9830
9830 LET KW=9840
9840 LET KX=9850
9850 LET KY=9860
9860 LET KZ=9870
9870 LET LA=9880
9880 LET LB=9890
9890 LET LC=9900
9900 LET LD=9910
9910 LET LE=9920
9920 LET LF=9930
9930 LET LG=9940
9940 LET LH=9950
9950 LET LI=9960
9960 LET LJ=9970
9970 LET LK=9980
9980 LET LL=9990
9990 LET LM=10000
10000 LET LN=10010
10010 LET LO=10020
10020 LET LP=10030
10030 LET LQ=10040
10040 LET LR=10050
10050 LET LS=10060
10060 LET LT=10070
10070 LET
```


Programa "GDU"

Con este programa se generan 21 gráficos definidos por usuario (GDU), de una forma aleatoria. Al lado de cada gráfico aparece la letra a la que queda asignado.

Una vez ejecutado pulse cualquier tecla para terminar y aparecerá el mensaje:

OK, 150:1

Pase a modo G y pulsando una tecla de la «A» a la «U», aparecerá en pantalla el gráfico correspondiente.

Programa "DIBUJANDO"

Este programa permite realizar cualquier tipo de dibujo en la pantalla, utilizando los cursores de movimiento situados en las teclas «5», «6», «7» y «8». Al ser ejecutado, lo primero que debemos hacer es pulsar una de las cuatro teclas que indican el color de la «tiza» (V = verde, C = cyan, A = amarillo, B = blanco).

En la parte superior izquierda de la pantalla aparece en

PROGRAMA 5

```

10 REM *****
11 ** CURSOR *****
12 ** BASIC/SINCLAIR *****
13 ** *****
14 ** "BIPBIP" *****
15 *****
16 BORDER 2: PAPER 2: INK 5: C
17
18 DRAW 255,0: DRAW 0,175: DR
19
20 DRAW 0: DRAW 0,175: DR
21
22 PRINT AT 5,7:"PROGRAMA ""BI
23
24 PRINT FLASH 1:AT 12,9:"PARE
25
26 PRINT AT 20,1:"
27
28 RESTORE
29
30 FOR X=1 TO 28
31 READ LOGO
32 PRINT CHR$(LOGO); BEEP 0.05
33
34 DATA 127,93,37,90,67,82,79,
35 72,79,65,36,73,62,65,70,
36 65,69,76,32,80,82,65,69,60
37
38 PRINT AT 12,9:
39
40 GO TO 100
41
42 REM *****
43 ** TEMPORIZACION *****
44 ** *****
45
46 FOR X=1 TO 300: NEXT X: RET
47
48 GO SUB 120: GO TO 400
49
50 REM *****
51 ** TECLADO *****
52 ** *****
53
54 FOR Y=12 TO 14
55 PRINT AT Y,X: PAPER 7: "
56
57 NEXT Y
58
59 INK 0
60
61 DRAW 0,23: DRAW
62
63 DRAW 0,23: TO 14,3 STEP 8
64
65 PLOT X,79: DRAW 0,-22
66
67 LET X=79
68
69 LET X=100: GO SUB 350
70
71 LET X=108: GO SUB 350
72
73 GO TO 100
74
75 REM *****
76 ** TECLADO MUSICAL *****
77 ** *****
78
79 BORDER 1: PAPER 1: INK 6: C
80
81 PRINT " A partir de este
82 momento su Spectrum se ha conve
83 rtido en un instrumento musical.
84 diferente cada vez que pulse u
85 na tecla, o simultáneamente pulse u
86 na combinación de teclas.
87
88 T...1...SYMBOL...PULSE...CAPS SHIP
89
90 0 PRINT 0: PULSE...CAPS SHIP
91
92 640+...SPACE... para terminar."
93
94 GO SUB 120
95
96 IF A$="IN..." THEN GO TO 670
97
98 IF A$="..." THEN GO TO 670
99
100 LET nota=CODE A$/4
101 BEEP duracion,nota
102
103 GO TO 670

```

en la pantalla, utilizando los cursores de movimiento situados en las teclas «5», «6», «7» y «8». Al ser ejecutado, lo primero que debemos hacer es pulsar una de las cuatro teclas que indican el color de la «tiza» (V = verde, C = cyan, A = amarillo, B = blanco).

En la parte superior izquierda de la pantalla aparece en

Para utilizar una variable dentro de una expresión, debe estar asignada previamente por una sentencia LET, es decir, si las variables «dato» y «a» no hubieran sido asignadas no se podría haber utilizado la sentencia:

30 LET resultado = dato + a

ya que hubiera dado el siguiente error:

2 Variable not found

lo mismo ocurre con la sentencia:

30 LET C\$ = A\$ + B\$

Veamos, por pasos, otro ejemplo de utilización de la sentencia LET. Supongamos que se desea implementar la fórmula que calcula el área de un círculo conociendo su diámetro (d = 10).

$$S = \pi r^2, \text{ donde } r = \frac{d}{2}$$

1.º Asignar a la variable «diámetro» el valor 10.

LET diámetro = 10
2.º Calcular el radio (r x d/2).

LET radio = diámetro/2
3.º Asignación del valor «pi» (π). Podríamos hacer:

LET pi = 3.141592

pero como el Spectrum dispone de esta constante numérica, no es necesario asignarla. Pi se encuentra en la tecla «M».

4.º Cálculo del área.

LET superficie = pi * radio ↑ 2
5.º El programa completo sería:

30 LET diámetro=10
40 LET superficie=pi*radio↑2

Después de ejecutar este programa la variable «superficie» tendrá almacenado el valor del área de un círculo de diámetro 10; para visualizar este valor incluya la siguiente instrucción y ejecútelo de nuevo.

40 PRINT superficie

Modificando el valor de la línea 10, obtendrá diversos valores para la variable «superficie».

Representación gráfica

La visualización de caracteres no ocupa la totalidad de la pantalla del televisor o monitor, ésta sólo se realiza en la zona central, dejando un espacio o borde alrededor (BORDER). La zona de representación permite visualizar 24 líneas de 32 caracteres, las líneas o filas están numeradas de la 0 a la 23 y las columnas de la 0 a la 31, esta disposición permite representar 768 caracteres y se denomina de *baja resolución*, ya que no permite hacer gráficos con calidad aceptable de definición.

Las líneas 0 a 21 son las utilizadas por el usuario y la 22 y 23 las utiliza el ordenador para la introducción de comandos directos y edición de programas; estas líneas también pueden ser utilizadas por el usuario accediendo de un modo especial.

Cada carácter de la zona de usuario puede dividirse en una matriz de 8 por 8 puntos, los cuales se denominan *pixel* (picture element o elementos de imagen), con esta nueva división se obtiene una retícula de 176 por 256 puntos, lo que nos da un total de 45.056 pixel,

esta modalidad se denomina *alta resolución* y permite hacer gráficos con una calidad aceptable de definición.

PRINT

Acceso al teclado



Tipo de sentencia

Comando de salida.

Definición

Este comando permite visualizar en la pantalla el valor de las constantes, variables, expresiones o textos indicados en el argumento.

Veamos los distintos tipos de estructura que adopta esta sentencia.

a) Visualizar una constante numérica.

SENTENCIA	ARGUMENTO
PRINT	constante

Ejemplos:

```

10 PRINT 20
30 PRINT 12.372
40 PRINT 0.50

```

b) Visualizar una constante alfanumérica.

SENTENCIA	ARGUMENTO
PRINT	"Cadena"

te de la sentencia REM y por lo tanto no se ejecuta.

— Un argumento largo puede ser incluido en una sola línea.

Ejemplo:

```

10 REM ***** MIHATTAN *****
   @ PABLO OTERO
   Agosto 84
   Edición ... 5
*****

```

Las sentencias REM ocupan parte de la memoria del ordenador, aunque como no son procesadas tienen la ventaja de no retardar la ejecución de un programa.

Funciones de Video



La función que el Spectrum presenta por defecto es la de *video normal* (TRUE VIDEO), es decir, que los caracteres se visualizan en el color de la tinta (INK) y el fondo en el del papel (PAPER). Si desea que estos dos colores se intercambien para destacar un fragmento de programa o algún texto, como, por ejemplo, el argumento de una sentencia REM, es necesario utilizar la función de *video invertido* (INV. VIDEO).

A esta función se accede,

en la edición de instrucciones, después de haber introducido el número de línea, pulsando la tecla CAPS SHIFT simultáneamente con la tecla correspondiente al número 4. A partir de este momento los caracteres tendrán el color del papel y el fondo el color de la tinta.

Para retornar a la visualización en video normal es necesario pulsar CAPS SHIFT y la tecla número 3, simultáneamente.

ADVERTENCIA

Para que este retorno tenga validez es necesario realizarlo dentro de una instrucción, es decir, después de haber introducido el número de línea y antes de pulsar ENTER, bien en la instrucción en la que se insertó la función invertida de video, bien en otra posterior.

Un ejemplo del manejo de estas funciones es el siguiente:

```

10 REM ***** MIHATTAN *****
   @ PABLO OTERO
   Agosto 84
   Edición ... 5
*****

```

Acceso al teclado



Tipo de sentencia

Comando de programación.

Definición

También es conocido como comando de *asignación* ya que a una variable (numérica o de cadena) le asigna un valor; este puede ser una constante o variable (numérica o de cadena), el resultado de una expresión matemática, una operación de cadena o una función VAL, esta última será vista en otro capítulo.

Deben tenerse en cuenta las siguientes consideraciones:

- Una variable o constante numérica y una expresión matemática, sólo pueden ser asignadas a una variable numérica.
- Una variable o constante alfanumérica, una operación de cadena (concatenación, fragmentación) o una función VAL pueden asignarse solamente a las funciones de cadena.

Las estructuras de esta sentencia son:

SENTENCIA	ARGUMENTO
LET	Variable = expresión

Ejemplos:

```

10 LET dato=25+7
30 LET resultado=dato*2
40 LET fin=resultado*7

```

SENTENCIA	ARGUMENTO
LET	Variable \$ = expresión de cadena

Ejemplos:

```

10 LET $="MICROBASIC"
30 LET $=$+"BASIC"
40 LET $=$+"BASIC"
50 LET $=$+"BASIC"

```

CLASIFICACION DE LAS SENTENCIAS SEGUN SU FUNCION

COMANDOS DE CONTROL
RUN
CONT
LIST
LOAD
SAVE
VERIFY
MERGE
CLEAR
CLS
NEW

MANEJO DE CADENAS
VAL
VAL\$
LEN
STR\$
CHR\$
CODE

COMANDOS DE CONTROL DE COLOR E IMPRESION
BORDER
PAPER
INK
BRIGHT
INVERSE
FLASH
OVER

FUNCIONES ARITMETICAS
SGN
ABS
INT
LN
EXP
SQR
FN
RND
SIN
COS
TAN
ASN
ACS
ATN
PI
BIN

COMANDOS DE PROGRAMACION
REM
LET
DIM
DEF EN
PAUSE
STOP
FORTO-STEP
NEXT
IF THEN
GO TO
GO SUB
RETURN
RESTORE
RAND
USR

AUXILIARES
LINE
TAB
AT
POINT
ATTR
SCREEN\$

SONIDO
BEEP

MANEJO IMPRESORA
LLIST
LPRINT
COPY

FUNCIONES LOGICAS
AND
OR
NOT

COMANDOS DE ENTRADA/SALIDA
PRINT
INPUT
INKEY\$
READ
DATA
PEEK
POKE
IN
OUT

MANEJO PERIFERICOS (INTERFACE-1)
FORMAT
CAT
ERASE
OPEN
CLOSE
MOVE

COMANDOS DE DIBUJO
PLOT
DRAW
CIRCLE

tiempo real las coordenadas de la «tiza». La opción «B» borra toda la pantalla y posiciona la «tiza» en las coordenadas iniciales $X = 127$ e $Y = 80$. La opción «F» permite salir del programa, apareciendo el mensaje:

9 STOP statement, 470 : 3

Para que se autoejecute, grábelo de la forma:

SAVE "DIBUJANDO" LINE 10

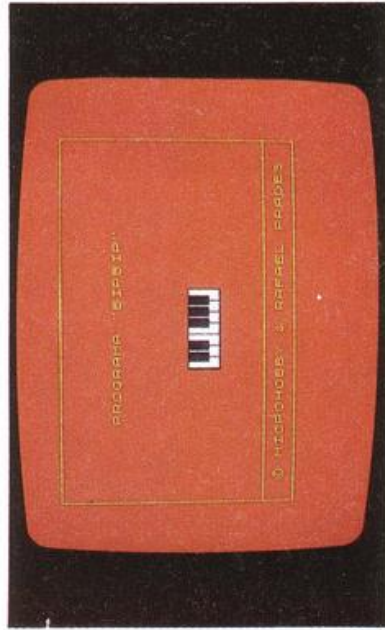
Programa "BIPBIP"

El último programa presentado convierte a su Spectrum en un instrumento musical.

Después de la presentación se genera una melodía conocida por todos. Las intrusiones de manejo se encuentran en el propio programa.

Sávelo de la siguiente manera:

SAVE "BIPBIP" LINE 10



COMANDOS BASICOS

REM

Acceso al teclado



Tipo de sentencia

Comando de programación

Concepto

Si durante la ejecución de un programa el ordenador detecta una sentencia REM, automáticamente analiza la siguiente instrucción y no ejecuta la correspondiente a REM. ¿Para qué sirve entonces una sentencia que no se ejecuta?, simplemente para poder introducir líneas de comentario (REMARK en inglés) dentro de un programa.

Estos comentarios sirven, por ejemplo, para indicar al principio de un programa el título de éste, el nombre del programador, la fecha y la edición, este último dato es bastante importante ya que nos permite comparar a simple vista, cuál es la versión actualizada de un mismo programa.

Dentro del programa, los comentarios sirven para indicar nos las funciones que realizan las distintas rutinas de que se compone o para aclarar el sig-

nificado de alguna de las variables utilizadas.

La estructura de esta sentencia es la siguiente:

SENTENCIA	ARGUMENTO
REM	Cualquier carácter

Ejemplos:

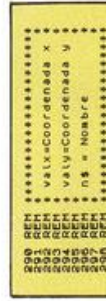
— Título de programas.



— Función de una rutina.

200 REM ** CALCULO VARIABLES **

— Significado de variables.



tas sentencias resalten sobre las demás. Se puede, por ejemplo, enmarcar los mensajes con asteriscos (*) u otro símbolo, también puede utilizarse la función de vídeo invertido (INV. VIDEO), ésta será explicada posteriormente.

— No es necesario encerrar el argumento entre comillas (" "), aunque esté formado por una cadena alfanumérica.

— Pueden ir en líneas independientes o al final de una cadena de sentencias.

Ejemplo:

200 PRINT "MICROHOBBY"; REM
*** ESTO ES UN EJEMPLO ***

ADVERTENCIA

A continuación de este tipo de sentencias, no debe editarse otra instrucción en la misma línea. Como el símbolo utilizado como separador de instrucciones (;) también puede formar parte del argumento de una sentencia REM, la sentencia editada a continuación no sería ejecutada.

Ejemplo:

200 REM; *** ESTO ES UN EJEMPLO ***
*** PRINT "MICROHOBBY";

Consideraciones

En la edición de sentencias REM es necesario tener en cuenta los siguientes puntos:

— Para localizar con facilidad las distintas rutinas de un programa conviene que es-

Si ejecutamos esta instrucción, resulta que la cadena «—MICROHOBBY—» no será visualizada, ya que el ordenador interpreta que forma par-

Otra aplicación

Existe una aplicación de la sentencia «INPUT» no especificada en el manual y que posiblemente sea consecuencia de los «efectos laterales» de la programación. Cuando la sentencia «INPUT» va acompañada de cualquiera de los siguientes argumentos:

```
INPUT cadena alfanumerica
INPUT expresion numerica
```

se borran las dos últimas líneas de la pantalla, la 22 y la 23.

¿Qué utilidad puede tener esto? Cuando se utilizan los canales de comunicación #0 y #1 para enviar mensajes a veces es necesario incluir una instrucción que borre estas líneas para que no se mezclen los mensajes, esto ocurre cuando el segundo es más corto que el primero, como por ejemplo:

```
10 PRINT "¿A qué hora desea un momento para comer?"
20 PAUSE 100
30 PRINT "¿A qué hora desea un momento para comer?"
40 PAUSE 0
```

Para subsanar esta alteración del segundo mensaje, podríamos incluir la instrucción:

```
25 PRINT #0; AT 0, 0
```

o bien

```
25 PRINT #0; AT 0, 22
```

pero resulta más cómodo hacerlo de la forma:

```
25 INPUT 0
```

y además tiene la ventaja de que ocupa menos memoria.

```
190 PRINT AT 16, 3; "TELEFONO:"
200 PRINT AT 19, 3; "POBLACION:"
210 PRINT #0; "Rellene los campos de la ficha"
220 PAUSE 200
230 REM
```

```
*****
* ENTRADA DE DATOS *
*****
```

```
340 PRINT AT 2, 11; FLASH 1; "<"
350 INPUT AT 2, 11; LINE n$
360 PRINT AT 2, 11; FLASH 1; "<"
370 INPUT AT 2, 11; LINE a$
380 PRINT AT 2, 11; FLASH 1; "<"
390 INPUT AT 2, 11; LINE b$
400 PRINT AT 2, 11; FLASH 1; "<"
410 INPUT AT 2, 11; LINE c$
420 PRINT AT 2, 11; FLASH 1; "<"
430 INPUT AT 2, 11; LINE d$
440 PRINT AT 2, 11; FLASH 1; "<"
450 INPUT AT 2, 11; LINE e$
460 PRINT AT 2, 11; FLASH 1; "<"
470 INPUT AT 2, 11; LINE f$
480 PRINT AT 2, 11; FLASH 1; "<"
490 INPUT AT 2, 11; LINE g$
500 PRINT AT 2, 11; FLASH 1; "<"
510 PRINT #0; AT 1, 10; "Fin de edición"
520 PAUSE 200
```

Veamos algunos ejemplos:

```
10 INPUT AT 0, 0; "Nombre:"
20 PRINT AT 0, 0; "Nombre:"
30 INPUT AT 0, 0; "Apellido:"
40 PRINT AT 0, 0; "Apellido:"
50 INPUT AT 0, 0; "Edad:"
60 PRINT AT 0, 0; "Edad:"
70 INPUT AT 0, 0; "Sexo:"
80 PRINT AT 0, 0; "Sexo:"
90 INPUT AT 0, 0; "Profesion:"
100 PRINT AT 0, 0; "Profesion:"
110 INPUT AT 0, 0; "Domicilio:"
120 PRINT AT 0, 0; "Domicilio:"
130 INPUT AT 0, 0; "Teléfono:"
140 PRINT AT 0, 0; "Teléfono:"
150 INPUT AT 0, 0; "Ocupación:"
160 PRINT AT 0, 0; "Ocupación:"
170 INPUT AT 0, 0; "Estado civil:"
180 PRINT AT 0, 0; "Estado civil:"
190 INPUT AT 0, 0; "Fecha de nacimiento:"
200 PRINT AT 0, 0; "Fecha de nacimiento:"
210 INPUT AT 0, 0; "Fecha de ingreso:"
220 PRINT AT 0, 0; "Fecha de ingreso:"
230 INPUT AT 0, 0; "Fecha de salida:"
240 PRINT AT 0, 0; "Fecha de salida:"
250 INPUT AT 0, 0; "Fecha de fallecimiento:"
260 PRINT AT 0, 0; "Fecha de fallecimiento:"
270 INPUT AT 0, 0; "Fecha de entierro:"
280 PRINT AT 0, 0; "Fecha de entierro:"
290 INPUT AT 0, 0; "Fecha de sepelio:"
300 PRINT AT 0, 0; "Fecha de sepelio:"
310 INPUT AT 0, 0; "Fecha de exhumación:"
320 PRINT AT 0, 0; "Fecha de exhumación:"
330 INPUT AT 0, 0; "Fecha de inhumación:"
340 PRINT AT 0, 0; "Fecha de inhumación:"
350 INPUT AT 0, 0; "Fecha de cremación:"
360 PRINT AT 0, 0; "Fecha de cremación:"
370 INPUT AT 0, 0; "Fecha de incineración:"
380 PRINT AT 0, 0; "Fecha de incineración:"
390 INPUT AT 0, 0; "Fecha de enterramiento:"
400 PRINT AT 0, 0; "Fecha de enterramiento:"
410 INPUT AT 0, 0; "Fecha de sepultamiento:"
420 PRINT AT 0, 0; "Fecha de sepultamiento:"
430 INPUT AT 0, 0; "Fecha de exhumación:"
440 PRINT AT 0, 0; "Fecha de exhumación:"
450 INPUT AT 0, 0; "Fecha de inhumación:"
460 PRINT AT 0, 0; "Fecha de inhumación:"
470 INPUT AT 0, 0; "Fecha de cremación:"
480 PRINT AT 0, 0; "Fecha de cremación:"
490 INPUT AT 0, 0; "Fecha de incineración:"
500 PRINT AT 0, 0; "Fecha de incineración:"
510 INPUT AT 0, 0; "Fecha de enterramiento:"
520 PRINT AT 0, 0; "Fecha de enterramiento:"
530 INPUT AT 0, 0; "Fecha de sepultamiento:"
540 PRINT AT 0, 0; "Fecha de sepultamiento:"
550 INPUT AT 0, 0; "Fecha de exhumación:"
560 PRINT AT 0, 0; "Fecha de exhumación:"
570 INPUT AT 0, 0; "Fecha de inhumación:"
580 PRINT AT 0, 0; "Fecha de inhumación:"
590 INPUT AT 0, 0; "Fecha de cremación:"
600 PRINT AT 0, 0; "Fecha de cremación:"
610 INPUT AT 0, 0; "Fecha de incineración:"
620 PRINT AT 0, 0; "Fecha de incineración:"
630 INPUT AT 0, 0; "Fecha de enterramiento:"
640 PRINT AT 0, 0; "Fecha de enterramiento:"
650 INPUT AT 0, 0; "Fecha de sepultamiento:"
660 PRINT AT 0, 0; "Fecha de sepultamiento:"
670 INPUT AT 0, 0; "Fecha de exhumación:"
680 PRINT AT 0, 0; "Fecha de exhumación:"
690 INPUT AT 0, 0; "Fecha de inhumación:"
700 PRINT AT 0, 0; "Fecha de inhumación:"
710 INPUT AT 0, 0; "Fecha de cremación:"
720 PRINT AT 0, 0; "Fecha de cremación:"
730 INPUT AT 0, 0; "Fecha de incineración:"
740 PRINT AT 0, 0; "Fecha de incineración:"
750 INPUT AT 0, 0; "Fecha de enterramiento:"
760 PRINT AT 0, 0; "Fecha de enterramiento:"
770 INPUT AT 0, 0; "Fecha de sepultamiento:"
780 PRINT AT 0, 0; "Fecha de sepultamiento:"
790 INPUT AT 0, 0; "Fecha de exhumación:"
800 PRINT AT 0, 0; "Fecha de exhumación:"
810 INPUT AT 0, 0; "Fecha de inhumación:"
820 PRINT AT 0, 0; "Fecha de inhumación:"
830 INPUT AT 0, 0; "Fecha de cremación:"
840 PRINT AT 0, 0; "Fecha de cremación:"
850 INPUT AT 0, 0; "Fecha de incineración:"
860 PRINT AT 0, 0; "Fecha de incineración:"
870 INPUT AT 0, 0; "Fecha de enterramiento:"
880 PRINT AT 0, 0; "Fecha de enterramiento:"
890 INPUT AT 0, 0; "Fecha de sepultamiento:"
900 PRINT AT 0, 0; "Fecha de sepultamiento:"
910 INPUT AT 0, 0; "Fecha de exhumación:"
920 PRINT AT 0, 0; "Fecha de exhumación:"
930 INPUT AT 0, 0; "Fecha de inhumación:"
940 PRINT AT 0, 0; "Fecha de inhumación:"
950 INPUT AT 0, 0; "Fecha de cremación:"
960 PRINT AT 0, 0; "Fecha de cremación:"
970 INPUT AT 0, 0; "Fecha de incineración:"
980 PRINT AT 0, 0; "Fecha de incineración:"
990 INPUT AT 0, 0; "Fecha de enterramiento:"
1000 PRINT AT 0, 0; "Fecha de enterramiento:"
```

INPUT LINE

La sentencia «INPUT» también puede combinarse con la palabra clave «LINE». Debe utilizarse únicamente con las variables de cadena y el resultado obtenido es el que el ordenador no visualiza las comillas durante la introducción de datos alfanuméricos.

La estructura es:

SENTENCIA	ARGUMENTO
INPUT	LINE var. cadena

Ejemplos:

```
10 INPUT LINE a$
20 PRINT AT 0, 0; "Nombre:"
30 INPUT LINE b$
40 PRINT AT 0, 0; "Apellido:"
50 INPUT LINE c$
60 PRINT AT 0, 0; "Edad:"
70 INPUT LINE d$
80 PRINT AT 0, 0; "Sexo:"
90 INPUT LINE e$
100 PRINT AT 0, 0; "Profesion:"
110 INPUT LINE f$
120 PRINT AT 0, 0; "Domicilio:"
130 INPUT LINE g$
140 PRINT AT 0, 0; "Teléfono:"
150 INPUT LINE h$
160 PRINT AT 0, 0; "Ocupación:"
170 INPUT LINE i$
180 PRINT AT 0, 0; "Estado civil:"
190 INPUT LINE j$
200 PRINT AT 0, 0; "Fecha de nacimiento:"
210 INPUT LINE k$
220 PRINT AT 0, 0; "Fecha de ingreso:"
230 INPUT LINE l$
240 PRINT AT 0, 0; "Fecha de salida:"
250 INPUT LINE m$
260 PRINT AT 0, 0; "Fecha de fallecimiento:"
270 INPUT LINE n$
280 PRINT AT 0, 0; "Fecha de entierro:"
290 INPUT LINE o$
300 PRINT AT 0, 0; "Fecha de sepelio:"
310 INPUT LINE p$
320 PRINT AT 0, 0; "Fecha de exhumación:"
330 INPUT LINE q$
340 PRINT AT 0, 0; "Fecha de inhumación:"
350 INPUT LINE r$
360 PRINT AT 0, 0; "Fecha de cremación:"
370 INPUT LINE s$
380 PRINT AT 0, 0; "Fecha de incineración:"
390 INPUT LINE t$
400 PRINT AT 0, 0; "Fecha de enterramiento:"
410 INPUT LINE u$
420 PRINT AT 0, 0; "Fecha de sepultamiento:"
430 INPUT LINE v$
440 PRINT AT 0, 0; "Fecha de exhumación:"
450 INPUT LINE w$
460 PRINT AT 0, 0; "Fecha de inhumación:"
470 INPUT LINE x$
480 PRINT AT 0, 0; "Fecha de cremación:"
490 INPUT LINE y$
500 PRINT AT 0, 0; "Fecha de incineración:"
510 INPUT LINE z$
520 PRINT AT 0, 0; "Fecha de enterramiento:"
530 INPUT LINE aa$
540 PRINT AT 0, 0; "Fecha de sepultamiento:"
550 INPUT LINE ab$
560 PRINT AT 0, 0; "Fecha de exhumación:"
570 INPUT LINE ac$
580 PRINT AT 0, 0; "Fecha de inhumación:"
590 INPUT LINE ad$
600 PRINT AT 0, 0; "Fecha de cremación:"
610 INPUT LINE ae$
620 PRINT AT 0, 0; "Fecha de incineración:"
630 INPUT LINE af$
640 PRINT AT 0, 0; "Fecha de enterramiento:"
650 INPUT LINE ag$
660 PRINT AT 0, 0; "Fecha de sepultamiento:"
670 INPUT LINE ah$
680 PRINT AT 0, 0; "Fecha de exhumación:"
690 INPUT LINE ai$
700 PRINT AT 0, 0; "Fecha de inhumación:"
710 INPUT LINE aj$
720 PRINT AT 0, 0; "Fecha de cremación:"
730 INPUT LINE ak$
740 PRINT AT 0, 0; "Fecha de incineración:"
750 INPUT LINE al$
760 PRINT AT 0, 0; "Fecha de enterramiento:"
770 INPUT LINE am$
780 PRINT AT 0, 0; "Fecha de sepultamiento:"
790 INPUT LINE an$
800 PRINT AT 0, 0; "Fecha de exhumación:"
810 INPUT LINE ao$
820 PRINT AT 0, 0; "Fecha de inhumación:"
830 INPUT LINE ap$
840 PRINT AT 0, 0; "Fecha de cremación:"
850 INPUT LINE aq$
860 PRINT AT 0, 0; "Fecha de incineración:"
870 INPUT LINE ar$
880 PRINT AT 0, 0; "Fecha de enterramiento:"
890 INPUT LINE as$
900 PRINT AT 0, 0; "Fecha de sepultamiento:"
910 INPUT LINE at$
920 PRINT AT 0, 0; "Fecha de exhumación:"
930 INPUT LINE au$
940 PRINT AT 0, 0; "Fecha de inhumación:"
950 INPUT LINE av$
960 PRINT AT 0, 0; "Fecha de cremación:"
970 INPUT LINE aw$
980 PRINT AT 0, 0; "Fecha de incineración:"
990 INPUT LINE ax$
1000 PRINT AT 0, 0; "Fecha de enterramiento:"
```

Tipo de sentencia

Auxiliar.

Concepto

Esta palabra clave se utiliza conjuntamente con PRINT y tabula la salida de datos al valor deseado. Su estructura es la siguiente:

```
PRINT TAB expresión; valor
```

una vez evaluada la expresión, nos indica ésta el número de columna a partir del cual debe visualizarse el valor (numérico o de cadena).

Ejemplos:

```
10 PRINT TAB 10; "Directorio:"
20 PRINT TAB 10; "Nombre:"
30 PRINT TAB 10; "Apellido:"
40 PRINT TAB 10; "Edad:"
50 PRINT TAB 10; "Sexo:"
60 PRINT TAB 10; "Profesion:"
70 PRINT TAB 10; "Domicilio:"
80 PRINT TAB 10; "Teléfono:"
90 PRINT TAB 10; "Ocupación:"
100 PRINT TAB 10; "Estado civil:"
110 PRINT TAB 10; "Fecha de nacimiento:"
120 PRINT TAB 10; "Fecha de ingreso:"
130 PRINT TAB 10; "Fecha de salida:"
140 PRINT TAB 10; "Fecha de fallecimiento:"
150 PRINT TAB 10; "Fecha de entierro:"
160 PRINT TAB 10; "Fecha de sepelio:"
170 PRINT TAB 10; "Fecha de exhumación:"
180 PRINT TAB 10; "Fecha de inhumación:"
190 PRINT TAB 10; "Fecha de cremación:"
200 PRINT TAB 10; "Fecha de incineración:"
210 PRINT TAB 10; "Fecha de enterramiento:"
220 PRINT TAB 10; "Fecha de sepultamiento:"
230 PRINT TAB 10; "Fecha de exhumación:"
240 PRINT TAB 10; "Fecha de inhumación:"
250 PRINT TAB 10; "Fecha de cremación:"
260 PRINT TAB 10; "Fecha de incineración:"
270 PRINT TAB 10; "Fecha de enterramiento:"
280 PRINT TAB 10; "Fecha de sepultamiento:"
290 PRINT TAB 10; "Fecha de exhumación:"
300 PRINT TAB 10; "Fecha de inhumación:"
310 PRINT TAB 10; "Fecha de cremación:"
320 PRINT TAB 10; "Fecha de incineración:"
330 PRINT TAB 10; "Fecha de enterramiento:"
340 PRINT TAB 10; "Fecha de sepultamiento:"
350 PRINT TAB 10; "Fecha de exhumación:"
360 PRINT TAB 10; "Fecha de inhumación:"
370 PRINT TAB 10; "Fecha de cremación:"
380 PRINT TAB 10; "Fecha de incineración:"
390 PRINT TAB 10; "Fecha de enterramiento:"
400 PRINT TAB 10; "Fecha de sepultamiento:"
410 PRINT TAB 10; "Fecha de exhumación:"
420 PRINT TAB 10; "Fecha de inhumación:"
430 PRINT TAB 10; "Fecha de cremación:"
440 PRINT TAB 10; "Fecha de incineración:"
450 PRINT TAB 10; "Fecha de enterramiento:"
460 PRINT TAB 10; "Fecha de sepultamiento:"
470 PRINT TAB 10; "Fecha de exhumación:"
480 PRINT TAB 10; "Fecha de inhumación:"
490 PRINT TAB 10; "Fecha de cremación:"
500 PRINT TAB 10; "Fecha de incineración:"
510 PRINT TAB 10; "Fecha de enterramiento:"
520 PRINT TAB 10; "Fecha de sepultamiento:"
530 PRINT TAB 10; "Fecha de exhumación:"
540 PRINT TAB 10; "Fecha de inhumación:"
550 PRINT TAB 10; "Fecha de cremación:"
560 PRINT TAB 10; "Fecha de incineración:"
570 PRINT TAB 10; "Fecha de enterramiento:"
580 PRINT TAB 10; "Fecha de sepultamiento:"
590 PRINT TAB 10; "Fecha de exhumación:"
600 PRINT TAB 10; "Fecha de inhumación:"
610 PRINT TAB 10; "Fecha de cremación:"
620 PRINT TAB 10; "Fecha de incineración:"
630 PRINT TAB 10; "Fecha de enterramiento:"
640 PRINT TAB 10; "Fecha de sepultamiento:"
650 PRINT TAB 10; "Fecha de exhumación:"
660 PRINT TAB 10; "Fecha de inhumación:"
670 PRINT TAB 10; "Fecha de cremación:"
680 PRINT TAB 10; "Fecha de incineración:"
690 PRINT TAB 10; "Fecha de enterramiento:"
700 PRINT TAB 10; "Fecha de sepultamiento:"
710 PRINT TAB 10; "Fecha de exhumación:"
720 PRINT TAB 10; "Fecha de inhumación:"
730 PRINT TAB 10; "Fecha de cremación:"
740 PRINT TAB 10; "Fecha de incineración:"
750 PRINT TAB 10; "Fecha de enterramiento:"
760 PRINT TAB 10; "Fecha de sepultamiento:"
770 PRINT TAB 10; "Fecha de exhumación:"
780 PRINT TAB 10; "Fecha de inhumación:"
790 PRINT TAB 10; "Fecha de cremación:"
800 PRINT TAB 10; "Fecha de incineración:"
810 PRINT TAB 10; "Fecha de enterramiento:"
820 PRINT TAB 10; "Fecha de sepultamiento:"
830 PRINT TAB 10; "Fecha de exhumación:"
840 PRINT TAB 10; "Fecha de inhumación:"
850 PRINT TAB 10; "Fecha de cremación:"
860 PRINT TAB 10; "Fecha de incineración:"
870 PRINT TAB 10; "Fecha de enterramiento:"
880 PRINT TAB 10; "Fecha de sepultamiento:"
890 PRINT TAB 10; "Fecha de exhumación:"
900 PRINT TAB 10; "Fecha de inhumación:"
910 PRINT TAB 10; "Fecha de cremación:"
920 PRINT TAB 10; "Fecha de incineración:"
930 PRINT TAB 10; "Fecha de enterramiento:"
940 PRINT TAB 10; "Fecha de sepultamiento:"
950 PRINT TAB 10; "Fecha de exhumación:"
960 PRINT TAB 10; "Fecha de inhumación:"
970 PRINT TAB 10; "Fecha de cremación:"
980 PRINT TAB 10; "Fecha de incineración:"
990 PRINT TAB 10; "Fecha de enterramiento:"
1000 PRINT TAB 10; "Fecha de sepultamiento:"
```

Cuando el resultado de la expresión es un número negativo, el ordenador presenta el mensaje:

```
B Integer out of range
```

Los números decimales son redondeados de manera que TAB 7.5 es igual que TAB 8 y TAB 7.4 es equivalente a TAB 7.

Podríamos suponer en un principio que los posibles valores de la expresión tendrían que estar comprendidos entre 0 y 31, ya que éstos son los números de columna existentes, pues bien, esto no es así, ya que podemos introducir cualquier número comprendido entre 0 y 65535. ¿Cómo tabula el ordenador una sentencia del tipo TAB 750? la respuesta es: reduciendo a «módulo 32», es decir, divide la expresión entre 32 sin obtener decimales y el resto de la división lo interpreta como número de columna.

Observe cómo las siguientes instrucciones, que simulan la forma en que el ordenador calcula el número de columna, realizan la misma función que

```
PRINT TAB 750; "MICROHOBBY"
```

o

```
PRINT TAB 14; "MICROHOBBY"
```

```
10 REM *** Módulo 32 ***
20 LET numero=750
30 LET cociente=INT (numero/32)
40 LET resto=numero-cociente*32
50 PRINT TAB resto;"MICROHOBBY"
```

esta sentencia auxiliar puede combinarse con los signos ortográficos (" , " ; " , ") que también determinan el formato de salida de datos.

Ejemplos:

```
10 PRINT TAB 10; "Nombre:"
20 PRINT TAB 10; "Apellido:"
30 PRINT TAB 10; "Edad:"
40 PRINT TAB 10; "Sexo:"
50 PRINT TAB 10; "Profesion:"
60 PRINT TAB 10; "Domicilio:"
70 PRINT TAB 10; "Teléfono:"
80 PRINT TAB 10; "Ocupación:"
90 PRINT TAB 10; "Estado civil:"
100 PRINT TAB 10; "Fecha de nacimiento:"
110 PRINT TAB 10; "Fecha de ingreso:"
120 PRINT TAB 10; "Fecha de salida:"
130 PRINT TAB 10; "Fecha de fallecimiento:"
140 PRINT TAB 10; "Fecha de entierro:"
150 PRINT TAB 10; "Fecha de sepelio:"
160 PRINT TAB 10; "Fecha de exhumación:"
170 PRINT TAB 10; "Fecha de inhumación:"
180 PRINT TAB 10; "Fecha de cremación:"
190 PRINT TAB 10; "Fecha de incineración:"
200 PRINT TAB 10; "Fecha de enterramiento:"
210 PRINT TAB 10; "Fecha de sepultamiento:"
220 PRINT TAB 10; "Fecha de exhumación:"
230 PRINT TAB 10; "Fecha de inhumación:"
240 PRINT TAB 10; "Fecha de cremación:"
250 PRINT TAB 10; "Fecha de incineración:"
260 PRINT TAB 10; "Fecha de enterramiento:"
270 PRINT TAB 10; "Fecha de sepultamiento:"
280 PRINT TAB 10; "Fecha de exhumación:"
290 PRINT TAB 10; "Fecha de inhumación:"
300 PRINT TAB 10; "Fecha de cremación:"
310 PRINT TAB 10; "Fecha de incineración:"
320 PRINT TAB 10; "Fecha de enterramiento:"
330 PRINT TAB 10; "Fecha de sepultamiento:"
340 PRINT TAB 10; "Fecha de exhumación:"
350 PRINT TAB 10; "Fecha de inhumación:"
360 PRINT TAB 10; "Fecha de cremación:"
370 PRINT TAB 10; "Fecha de incineración:"
380 PRINT TAB 10; "Fecha de enterramiento:"
390 PRINT TAB 10; "Fecha de sepultamiento:"
400 PRINT TAB 10; "Fecha de exhumación:"
410 PRINT TAB 10; "Fecha de inhumación:"
420 PRINT TAB 10; "Fecha de cremación:"
430 PRINT TAB 10; "Fecha de incineración:"
440 PRINT TAB 10; "Fecha de enterramiento:"
450 PRINT TAB 10; "Fecha de sepultamiento:"
460 PRINT TAB 10; "Fecha de exhumación:"
470 PRINT TAB 10; "Fecha de inhumación:"
480 PRINT TAB 10; "Fecha de cremación:"
490 PRINT TAB 10; "Fecha de incineración:"
500 PRINT TAB 10; "Fecha de enterramiento:"
510 PRINT TAB 10; "Fecha de sepultamiento:"
520 PRINT TAB 10; "Fecha de exhumación:"
530 PRINT TAB 10; "Fecha de inhumación:"
540 PRINT TAB 10; "Fecha de cremación:"
550 PRINT TAB 10; "Fecha de incineración:"
560 PRINT TAB 10; "Fecha de enterramiento:"
570 PRINT TAB 10; "Fecha de sepultamiento:"
580 PRINT TAB 10; "Fecha de exhumación:"
590 PRINT TAB 10; "Fecha de inhumación:"
600 PRINT TAB 10; "Fecha de cremación:"
610 PRINT TAB 10; "Fecha de incineración:"
620 PRINT TAB 10; "Fecha de enterramiento:"
630 PRINT TAB 10; "Fecha de sepultamiento:"
640 PRINT TAB 10; "Fecha de exhumación:"
650 PRINT TAB 10; "Fecha de inhumación:"
660 PRINT TAB 10; "Fecha de cremación:"
670 PRINT TAB 10; "Fecha de incineración:"
680 PRINT TAB 10; "Fecha de enterramiento:"
690 PRINT TAB 10; "Fecha de sepultamiento:"
700 PRINT TAB 10; "Fecha de exhumación:"
710 PRINT TAB 10; "Fecha de inhumación:"
720 PRINT TAB 10; "Fecha de cremación:"
730 PRINT TAB 10; "Fecha de incineración:"
740 PRINT TAB 10; "Fecha de enterramiento:"
750 PRINT TAB 10; "Fecha de sepultamiento:"
760 PRINT TAB 10; "Fecha de exhumación:"
770 PRINT TAB 10; "Fecha de inhumación:"
780 PRINT TAB 10; "Fecha de cremación:"
790 PRINT TAB 10; "Fecha de incineración:"
800 PRINT TAB 10; "Fecha de enterramiento:"
810 PRINT TAB 10; "Fecha de sepultamiento:"
820 PRINT TAB 10; "Fecha de exhumación:"
830 PRINT TAB 10; "Fecha de inhumación:"
840 PRINT TAB 10; "Fecha de cremación:"
850 PRINT TAB 10; "Fecha de incineración:"
860 PRINT TAB 10; "Fecha de enterramiento:"
870 PRINT TAB 10; "Fecha de sepultamiento:"
880 PRINT TAB 10; "Fecha de exhumación:"
890 PRINT TAB 10; "Fecha de inhumación:"
900 PRINT TAB 10; "Fecha de cremación:"
910 PRINT TAB 10; "Fecha de incineración:"
920 PRINT TAB 10; "Fecha de enterramiento:"
930 PRINT TAB 10; "Fecha de sepultamiento:"
940 PRINT TAB 10; "Fecha de exhumación:"
950 PRINT TAB 10; "Fecha de inhumación:"
960 PRINT TAB 10; "Fecha de cremación:"
970 PRINT TAB 10; "Fecha de incineración:"
980 PRINT TAB 10; "Fecha de enterramiento:"
990 PRINT TAB 10; "Fecha de sepultamiento:"
1000 PRINT TAB 10; "Fecha de exhumación:"
```

sustituya el punto y coma del final por una coma y observe el nuevo resultado. Las cinco sentencias anteriores podrían editarse en una sola:

```
10 PRINT TAB 10; "Nombre:"
20 PRINT TAB 10; "Apellido:"
30 PRINT TAB 10; "Edad:"
40 PRINT TAB 10; "Sexo:"
50 PRINT TAB 10; "Profesion:"
60 PRINT TAB 10; "Domicilio:"
70 PRINT TAB 10; "Teléfono:"
80 PRINT TAB 10; "Ocupación:"
90 PRINT TAB 10; "Estado civil:"
100 PRINT TAB 10; "Fecha de nacimiento:"
110 PRINT TAB 10; "Fecha de ingreso:"
120 PRINT TAB 10; "Fecha de salida:"
130 PRINT TAB 10; "Fecha de fallecimiento:"
140 PRINT TAB 10; "Fecha de entierro:"
150 PRINT TAB 10; "Fecha de sepelio:"
160 PRINT TAB 10; "Fecha de exhumación:"
170 PRINT TAB 10; "Fecha de inhumación:"
180 PRINT TAB 10; "Fecha de cremación:"
190 PRINT TAB 10; "Fecha de incineración:"
200 PRINT TAB 10; "Fecha de enterramiento:"
210 PRINT TAB 10; "Fecha de sepultamiento:"
220 PRINT TAB 10; "Fecha de exhumación:"
230 PRINT TAB 10; "Fecha de inhumación:"
240 PRINT TAB 10; "Fecha de cremación:"
250 PRINT TAB 10; "Fecha de incineración:"
260 PRINT TAB 10; "Fecha de enterramiento:"
270 PRINT TAB 10; "Fecha de sepultamiento:"
280 PRINT TAB 10; "Fecha de exhumación:"
290 PRINT TAB 10; "Fecha de inhumación:"
300 PRINT TAB 10; "Fecha de cremación:"
310 PRINT TAB 10; "Fecha de incineración:"
320 PRINT TAB 10; "Fecha de enterramiento:"
330 PRINT TAB 10; "Fecha de sepultamiento:"
340 PRINT TAB 10; "Fecha de exhumación:"
350 PRINT TAB 10; "Fecha de inhumación:"
360 PRINT TAB 10; "Fecha de cremación:"
370 PRINT TAB 10; "Fecha de incineración:"
380 PRINT TAB 10; "Fecha de enterramiento:"
390 PRINT TAB 10; "Fecha de sepultamiento:"
400 PRINT TAB 10; "Fecha de exhumación:"
410 PRINT TAB 10; "Fecha de inhumación:"
420 PRINT TAB 10; "Fecha de cremación:"
430 PRINT TAB 10; "Fecha de incineración:"
440 PRINT TAB 10; "Fecha de enterramiento:"
450 PRINT TAB 10; "Fecha de sepultamiento:"
460 PRINT TAB 10; "Fecha de exhumación:"
470 PRINT TAB 10; "Fecha de inhumación:"
480 PRINT TAB 10; "Fecha de cremación:"
490 PRINT TAB 10; "Fecha de incineración:"
500 PRINT TAB 10; "Fecha de enterramiento:"
510 PRINT TAB 10; "Fecha de sepultamiento:"
520 PRINT TAB 10; "Fecha de exhumación:"
530 PRINT TAB 10; "Fecha de inhumación:"
540 PRINT TAB 10; "Fecha de cremación:"
550 PRINT TAB 10; "Fecha de incineración:"
560 PRINT TAB 10; "Fecha de enterramiento:"
570 PRINT TAB 10; "Fecha de sepultamiento:"
580 PRINT TAB 10; "Fecha de exhumación:"
590 PRINT TAB 10; "Fecha de inhumación:"
600 PRINT TAB 10; "Fecha de cremación:"
610 PRINT TAB 10; "Fecha de incineración:"
620 PRINT TAB 10; "Fecha de enterramiento:"
630 PRINT TAB 10; "Fecha de sepultamiento:"
640 PRINT TAB 10; "Fecha de exhumación:"
650 PRINT TAB 10; "Fecha de inhumación:"
660 PRINT TAB 10; "Fecha de cremación:"
670 PRINT TAB 10; "Fecha de incineración:"
680 PRINT TAB 10; "Fecha de enterramiento:"
690 PRINT TAB 10; "Fecha de sepultamiento:"
700 PRINT TAB 10; "Fecha de exhumación:"
710 PRINT TAB 10; "Fecha de inhumación:"
720 PRINT TAB 10; "Fecha de cremación:"
730 PRINT TAB 10; "Fecha de incineración:"
740 PRINT TAB 10; "Fecha de enterramiento:"
750 PRINT TAB 10; "Fecha de sepultamiento:"
760 PRINT TAB 10; "Fecha de exhumación:"
770 PRINT TAB 10; "Fecha de inhumación:"
780 PRINT TAB 10; "Fecha de cremación:"
790 PRINT TAB 10; "Fecha de incineración:"
800 PRINT TAB 10; "Fecha de enterramiento:"
810 PRINT TAB 10; "Fecha de sepultamiento:"
820 PRINT TAB 10; "Fecha de exhumación:"
830 PRINT TAB 10; "Fecha de inhumación:"
840 PRINT TAB 10; "Fecha de cremación:"
850 PRINT TAB 10; "Fecha de incineración:"
860 PRINT TAB 10; "Fecha de enterramiento:"
870 PRINT TAB 10; "Fecha de sepultamiento:"
880 PRINT TAB 10; "Fecha de exhumación:"
890 PRINT TAB 10; "Fecha de inhumación:"
900 PRINT TAB 10; "Fecha de cremación:"
910 PRINT TAB 10; "Fecha de incineración:"
920 PRINT TAB 10; "Fecha de enterramiento:"
930 PRINT TAB 10; "Fecha de sepultamiento:"
940 PRINT TAB 10; "Fecha de exhumación:"
950 PRINT TAB 10; "Fecha de inhumación:"
960 PRINT TAB 10; "Fecha de cremación:"
970 PRINT TAB 10; "Fecha de incineración:"
980 PRINT TAB 10; "Fecha de enterramiento:"
990 PRINT TAB 10; "Fecha de sepultamiento:"
1000 PRINT TAB 10; "Fecha de exhumación:"
```

AT

Acceso al teclado

CODE



IN



Auxiliar.

Concepto

Esta sentencia visualiza a partir del número de línea y columna especificado, un valor. Su estructura es la siguiente:

PRINT AT linea. columna: valor

Ejemplos:

[illegible]

Cuando el valor de *línea* es superior a 22 ó 31 en el caso de *columna*, el ordenador presenta el mensaje:

B Integer out of range

En cambio, cuando se quiere imprimir en la primera línea de la zona destinada a los mensajes (22), presenta otro distinto:

5 Out of screen

Los números negativos los interpreta como si fueran positivos, por tanto dará igual editar:

10 PRINT AT 10, 10: "HOLA"

○

10 PRINT AT -10, -10: "Hola"

aunque la primera forma es la más correcta.

74 MICROBASIC

PROGRAMA 1

```

10 REM *****
** CURSO BASIC **
*****
** "GRANJA" **
*****
L5 20 BORDER 1; PAPER 1: INK 7: C
30 REM *****
** INTRODUCCION **
*****
40 PRINT AT 3,3;"EL programa A
NIMAL calcula el numero total
de animales que con viven en
una granja, asi como la suma
de sus patas."
50 PRINT AT 12,3;"Introduzca l
os datos a medida que el orde
nador se los vaya pidiendo."
60 PRINT #0;AT 0,2;"Pulse una
tecla para seguir"
70 PAUSE 0
80 CLS
90 REM *****
** ENTRADA DE DATOS **
*****
100 INPUT "Numero de patos? ";P
110 INPUT "Numero de gallinas?";gallina
120 INPUT "Numero de conejos?";conejo
130 INPUT "Numero de palomas?";paloma
140 INPUT "Numero de cerdos?";cerdo
150 REM *****
** PRINT DATOS **
*****
160 PRINT AT 2,0;"Patos ...."
170 PRINT AT 4,0;"Gallinas .."
180 PRINT AT 6,0;"Conejos ..."
190 PRINT AT 8,0;"Palomas ..."
200 PRINT AT 10,0;"Cerdos ...."
210 REM *****
** CALCULO TOTAL **
*****

```

PROGRAMA 5

```

10 REM *****  

** CURSO BASIC  

*****  

** FICHA  

*****  

**  

*****  

L$= "PAPER 1: INK 6: C"  

20 BORDER 1:  

30 REM *****  

** DIBUJO FICHA **  

*****  

50 LET a$=""  

60 LET b$=""  

70 PRINT ""+a$+" "  

80 FOR n = 1 TO 20  

90 PRINT AT n,0,"";AT n,31;" "  

100 NEXT n  

110 PRINT ""+b$+" "  

120 REM *****  

** ROTULOS FICHA *  

*****  

130 PRINT AT 2,3:"NOMBRE:"  

140 PRINT AT 5,11;"APELLIDOS"  

150 PRINT AT 6,1;  

160 PRINT AT 8,3;"1:";AT 8,16;" "  

170 PRINT AT 11,3;"CALLE:";AT 1  

1,23;"N."  

180 PRINT AT 13,3;"PISO :";AT 1  

3,18;"PUERTA:"

```

Ejemplos:

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
00	01	02	03	04																																																																																															

Cuando entre los comentarios del mensaje deba figurar el contenido de una variable, esta deberá ir encerrada entre paréntesis, bien ella sola, bien todo el mensaje, ya que de lo contrario, el intérprete BASIC la tomará como variable a introducir.

Ejemplos:

```
10 INPUT "Nombre? "; N$
20 INPUT "HOLA "; N$;" en que
   calle vives? "; C$
30 INPUT "En que numero de la
   calle "; N$;" vives? "; N$
40 PRINT "Te llamas "; N$;" y v
   ves en el numero "; N$;" de la ca
   lle "; C$;
```

INPUT TAB y AT

Las sentencias auxiliares «TAB» y «AT» también pueden utilizarse conjuntamente con «INPUT». La palabra clave «TAB» se utiliza de forma similar que cuando acompaña a la sentencia «PRINT».

Ejemplos:

```

10 INPUT TAB 25: a
20 INPUT TAB 27: "Nombre: ", b
30 INPUT TAB 2: "Tabulation: ",
40 INPUT TAB (a): "Mémoria en K
tes: ", w

```

«AT» tiene un tratamiento ligeramente distinto. Todos los «INPUT» a utilizar con «A» deben estar incluidos en la misma instrucción, separados por «,». Independientemente de las coordenadas del primer «AT», la entrada de datos del primer «INPUT» se realiza en la zona inferior de la pantalla. Los siguientes se van a introducir en las coordenadas indicadas en los «AT», pero tomando como línea de referencia la especificada en el primer «AT».

Canales de Comunicación

El Spectrum dispone de una serie de canales de comunicación o «streams», por los que el ordenador mantiene el intercambio de información con sus periféricos. Estos canales están numerados del «0» al «15» y precedidos por el signo del sostenido (#).

Los canales « Φ » al «3» tienen una asignación fija de pines y el resto puede ser utilizado por cualquiera de ellos, previa definición por el usuario.

El canal # es el asignado al periférico conocido como *impresora*, por tanto será de salida. Una instrucción del tipo:

PRINT # 3: "MICROHOB"™

imprimirá la misma cadena que la sentencia específica de la impresora (LPRINT):

I PRINT "MICROHOBBY"

Los canales # ϕ y # 1 son de entrada/salida y están relacionados con las dos últimas líneas de la pantalla, la 22 y 23. Estas líneas están destinadas para la introducción de sentencias en la edición de pro-

MICROBASIC 75

Canales de Comunicación

Pueden ser de *entrada* (INPUT) si los datos llegan al ordenador procedentes de alguno de sus periféricos, de *salida* (OUTPUT), si el ordenador es el que los envía y de *entrada/salida* (I/O), si el canal sirve tanto para enviar como para recibir datos.

A través del canal # 2 se pueden visualizar valores numéricos o cadenas en la zona de pantalla destinada al usuario, por tanto es indiferente utilizar el comando «PRINT # 2» o «PRINT», ya que este último lleva asignado, implícitamente el canal de comunicación # 2.

El canal # es el asignado al periférico conocido como *impresora*, por tanto será de salida. Una instrucción del tipo:

PRINT # 3: "MICROHOB"™

imprimirá la misma cadena que la sentencia específica de la impresora (LPRINT):

I PRINT "MICROHOBBY"

Los canales # ϕ y # 1 son de entrada/salida y están relacionados con las dos últimas líneas de la pantalla, la 22 y 23. Estas líneas están destinadas para la introducción de sentencias en la edición de pro-

MICROBASIC 75

INPUT

Acceso al teclado

CODE



MODO K

IN

Tipo de sentencia

Comando de entrada.

Concepto

Esta sentencia permite introducir por teclado, durante la ejecución de un programa, datos, tanto numéricos como de cadena y se asignan a la variable indicada en el argumento. Los datos introducidos se visualizan en las dos líneas inferiores de la pantalla, si cometemos algún error, podemos corregirlo con la función «DELETE». Para que los datos sean aceptados por el ordenador y continúe, por tanto, la ejecución del programa, debe pulsarse la tecla «ENTER».

Las estructuras básicas de esta sentencia son:

SENTENCIA	ARGUMENTO
INPUT	Var. numérica

Ejemplos:



Si se introduce un valor no numérico, el intérprete del

```
180 PAUSE 0
190 CLS
200 REM
```

```
*****
***** ENTRADA DE DATOS *****
*****
```

```
210 INPUT AT 0,0;"Valor de 'a'"
220 INPUT AT 1,0;"Valor de 'b'"
230 INPUT AT 2,0;"Valor de 'c'"
240 REM
```

```
***** VISUALIZACION *****
*****
```

```
250 PRINT "a=";"a
260 PRINT "b=";"b
270 PRINT "c=";"c
280 REM
```

```
***** CALCULO DE RAICES *****
*****
```

```
290 LET raiz=SQR (b*b-(4*a*c))
300 LET divisor=2*a
310 LET raiz1=(-b+raiz)/divisor
320 LET raiz2=(-b-raiz)/divisor
330 REM
```

```
***** VISUALIZACION DE *****
*****
```

```
410 PRINT "Primera raiz ..."
420 PRINT "Segunda raiz ..."
430 REM
```

gramas, a la de datos cuando se utiliza la sentencia «INPUT» y para la visualización de los informes del ordenador.

Como ya vimos, con la sentencia PRINT no se podía visualizar ningún texto en estas líneas, sin embargo utilizando cualquiera de estos dos canales (0 ó 1) esto es posible. Ejecute las siguientes instrucciones y lo comprobará:



Al pulsar cualquier tecla, observará que las líneas 22 y

23 desaparecen, ¿por qué ocurre esto? Como hemos dicho anteriormente, estas líneas también las utiliza el ordenador para enviarnos sus mensajes, por tanto al visualizar el informe de fin de programa:

OK, 00,1

estas desaparecen.

Para aprovechar al máximo la capacidad de la zona de visualización destinada al usuario, los mensajes que tengan que enviarse en un programa, pueden hacerse a través de estos canales.

Ejemplos:

```
10 PRINT "Pulse una tecla p"
20 CONTINUE
30 PAUSE 0.05,20
```

```
10 PRINT "Para grabar pulse"
20 PAUSE 0
30 PAUSE 0.05,20
```

La instrucción «PAUSE 0» detiene la ejecución de un programa hasta que se pulsa una tecla, y la sentencia «BEEP 0,05,20» hace que suene el altavoz interno del Spectrum durante 0,05 sg. y con un tono de valor 20.

La impresión a través de los canales #0 y #1 puede combinarse con las sentencias auxiliares TAB y AT, teniendo en cuenta que para esta última, la línea 22 se convierte en la 0 y la 23 en la 1.

Ejemplos:

```
10 PRINT "Tab 14.";"ola"
20 PAUSE 0
30 BEEP 0.05,5
```

```
10 PRINT "At 0,0;"Conecte la"
20 PRINT "impresora"
30 PRINT "a tecla para seguir"
40 PAUSE 0.05,30
```

Cuando se utiliza un canal que no está activado, aparece el siguiente mensaje:

O invalid stream

Al especificar una operación de entrada en un canal destinado a salida el informe de error es:

J Invalid I/O device

PROGRAMA 3

```
10 REM
*****
***** CURSO BASIC *****
*****
***** INTERES SIMPLE *****
*****
```

```
11 BORDER 1: PAPER 1: INK 7: C
12 REM
```

```
*****
***** INTRODUCCION *****
*****
```

```
13 PRINT AT 2,3;"Este programa calcula el interés simple de acuerdo con la fórmula:"
14 PRINT AT 4,0;"res simple de"
15 PRINT AT 6,0;"mula:"
16 PRINT AT 10,11;"C * R * T"
17 PRINT AT 11,7;"I ="
18 PRINT AT 13,14;100: PAUSE 3
19 CLS
```

```
***** ENTRADA DE *****
***** DATOS *****
*****
```

```
20 INPUT "CAPITAL: ";capital
30 PRINT "CAPITAL: ...";capital
40 INPUT "RENTAS en %: ";rentas
50 PRINT "RENTAS ...";rentas
60 INPUT "TIEMPO: ";tiempo
70 PRINT "TIEMPO: ...";tiempo
80 REM
```

```
***** CALCULO *****
*****
```

```
90 LET interes=capital*rentas*tiempo/100
100 REM
```

```
***** RESULTADO *****
*****
```

```
110 PRINT "INTERESES ..";interes;" pesetas"
```


entre los márgenes 1 y 20. Los bucles «FOR»...«NEXT» serán estudiados con mayor detalle en otro capítulo.

La sentencia «FLASH 1» antepuesta al símbolo «<» (menor que) hace que este parpadee en la pantalla para llamar la atención sobre el dato a introducir.

Y por último la sentencia «PAUSE 20» temporiza, aproximadamente, la ejecución del programa durante cuatro segundos.

La estructura del programa es la siguiente:

10 : Comentario con el nombre del programa.

20 : Asignación de los colores de la pantalla, azul, para el borde y papel, y amarillo para los caracteres.

50 — 110 : Dibujo del contorno con los gráficos predefinidos.

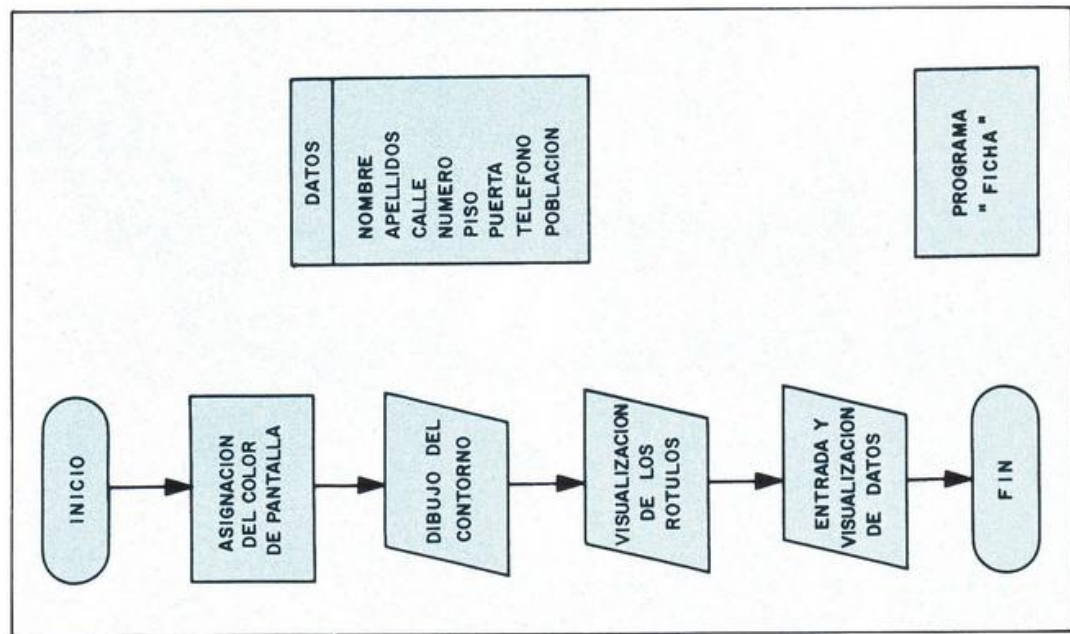
130 — 210 : Visualización de los campos de la ficha.

220 : Temporización.

240 — 500 : Entrada de los datos y visualización de estos en los campos correspondientes.

510 : Utilización del canal de comunicación para visualizar el informe «Fin de edición».

520 : Temporización.



Programa "Ficha".

Este programa simula una ficha con diversos campos (nombre, apellidos, etc...) que el usuario debe rellenar. Este programa puede formar parte de otro mayor que almacene, por ejemplo, los datos de los empleados de una empresa, los datos de los clientes o los suministradores.

Al editar el programa, debe poner atención en las líneas

50 a 110, ya que van incluidos algunos de los gráficos predefinidos que incorpora el Spectrum. Debe pasar a modo G (gráficos).

Las líneas 80 a 100 forman lo que en programación se llama **bucle**, por tanto la instrucción 90 se repetirá 20 veces pero con diversos valores de la variable «n», ya que el índice del bucle está comprendido

Programas de repaso

Como colofón al estudio de las sentencias básicas de programación:

REM
LET
PRINT
INPUT

se analizan en este capítulo cinco programas realizados con este tipo de instrucciones.

Estos programas, numerados del uno al cinco, son los siguientes:

1. GRANJA
2. ECUACION
3. INTERES
4. GRADOS
5. FICHA

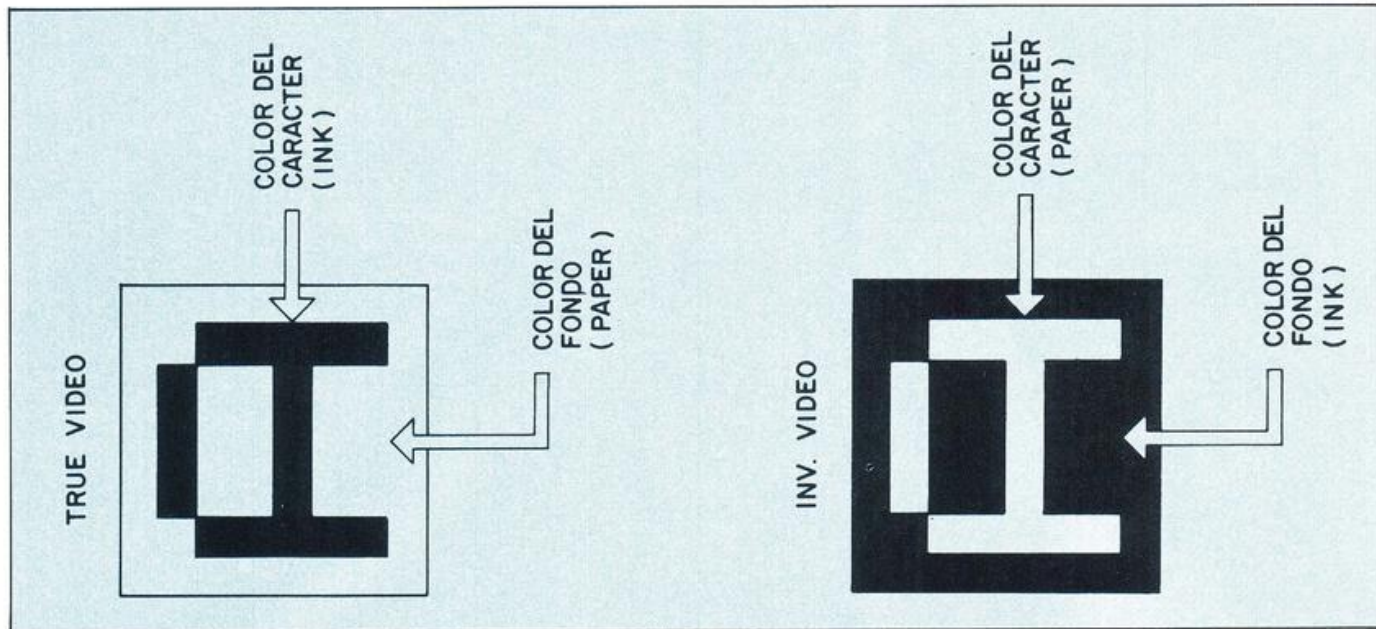
Programa «GRANJA»

Para almacenarlo en cinta una vez editado hágalo, por ejemplo, de la forma:

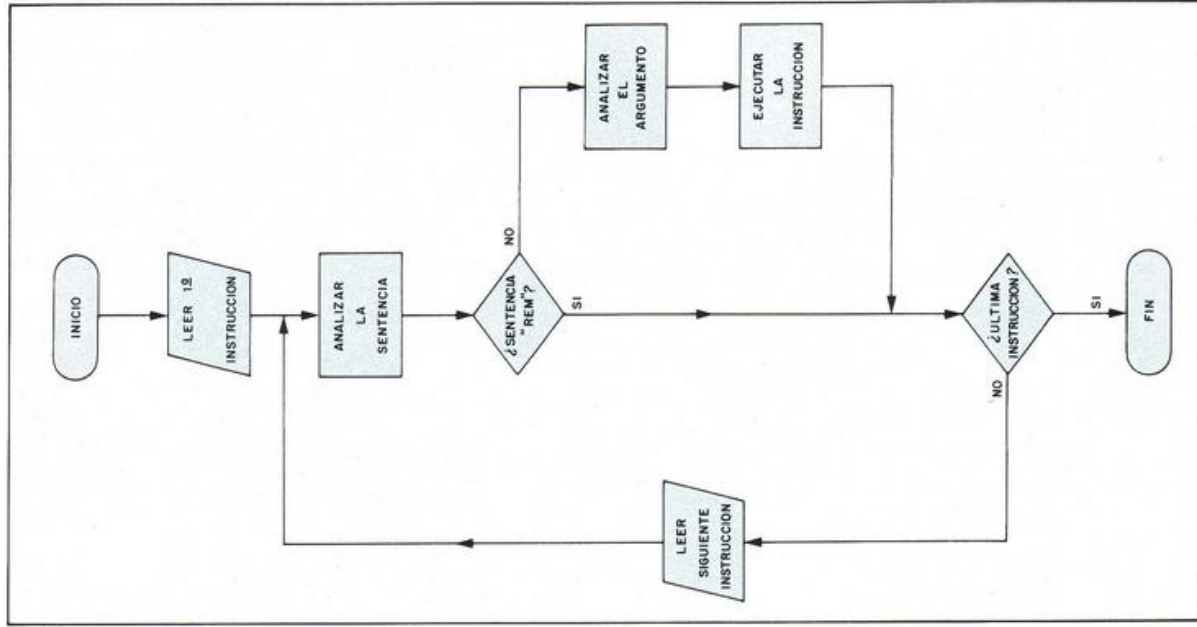
SAVE "granja"

Este programa es bastante sencillo, ya que calcula el número total de animales que hay en una granja, a partir de los datos que le son introducidos por teclado; también calcula el número total de pautas.

La sentencia 10 es la pre-sentación del programa. En la 20 hay cuatro sentencias, esto es posible ya que se utiliza el signo separador «:»; éstas no se han explicado todavía, pero vamos a ver unas pequeñas nociones sobre su funcionamiento. En conjunto realizan la tarea de colorear la pantalla. La sentencia «BORDER 1» asigna el color azul al borde de la pantalla, «PAPER 1» asigna el color azul al fondo, e «INK 7» tiene la misión de que el color de los caracteres sea



Funciones de video.



Análisis sentencia "REM".

blanco. Por último, «CLS» es una sentencia que borra la pantalla y asigna de inmediato los colores especificados en los comandos anteriores.

Las sentencias 4φ y 5φ son canal de comunicación φ

del tipo «PRINT AT» y sirven para dar al usuario una pequeña información sobre la finalidad del programa.

La instrucción 6φ utiliza el canal de comunicación φ

para visualizar el mensaje de espera.

«PAUSE φ» situada en la línea 7φ es una instrucción que detiene la ejecución del programa hasta que se pulsa una tecla.

La sentencia 8φ borra la información visualizada en la pantalla.

De la línea 10φ a la 14φ se encuentran los «INPUT» necesarios para la entrada de datos. La variable numérica utilizada para almacenar el número total de patos es «pato», para los restantes animales se han utilizado: «gallina», «conejo», «paloma» y «cerdo».

Después de la introducción de datos, se realiza la visualización detallada de éstos, los «PRINT AT» de las líneas 16φ a 20φ, se encargan de ello.

El cálculo del número total de animales se realiza en la línea 22φ, se asigna a la variable «total» la suma de las variables «pato», «gallina», «conejo», «paloma» y «cerdo». La línea 23φ se encarga de visualizar este valor.

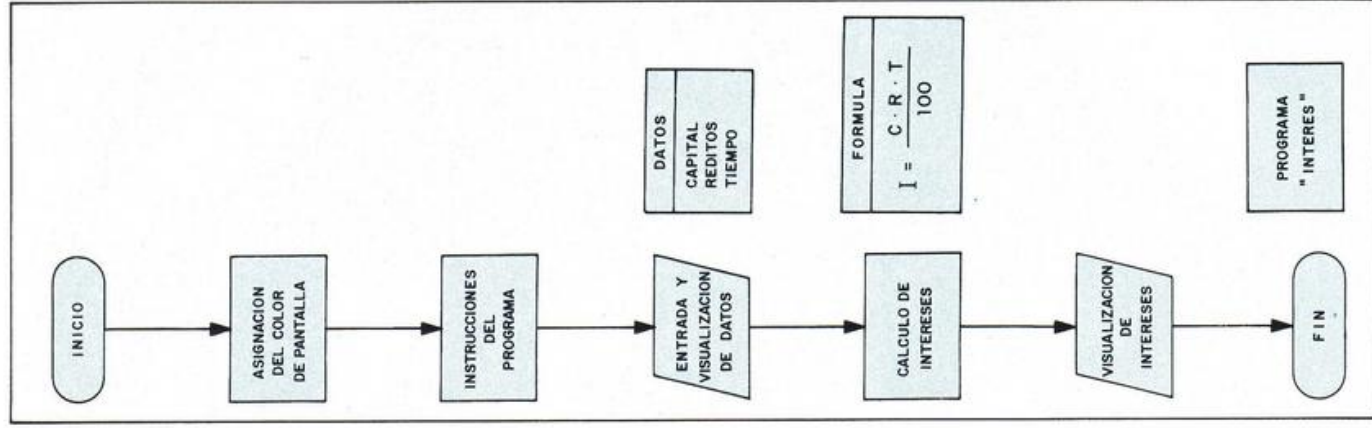
Las líneas 25φ a 30φ se encargan de calcular el número total de patas. Primero se asignan a las variables «pat», «gat», «co», «pal» y «ce» los valores totales por especie.

Ejemplo: como los conejos tiene cuatro patas, será necesario multiplicar este número por el número total de conejos, valor especificado en la variable «conejo»:

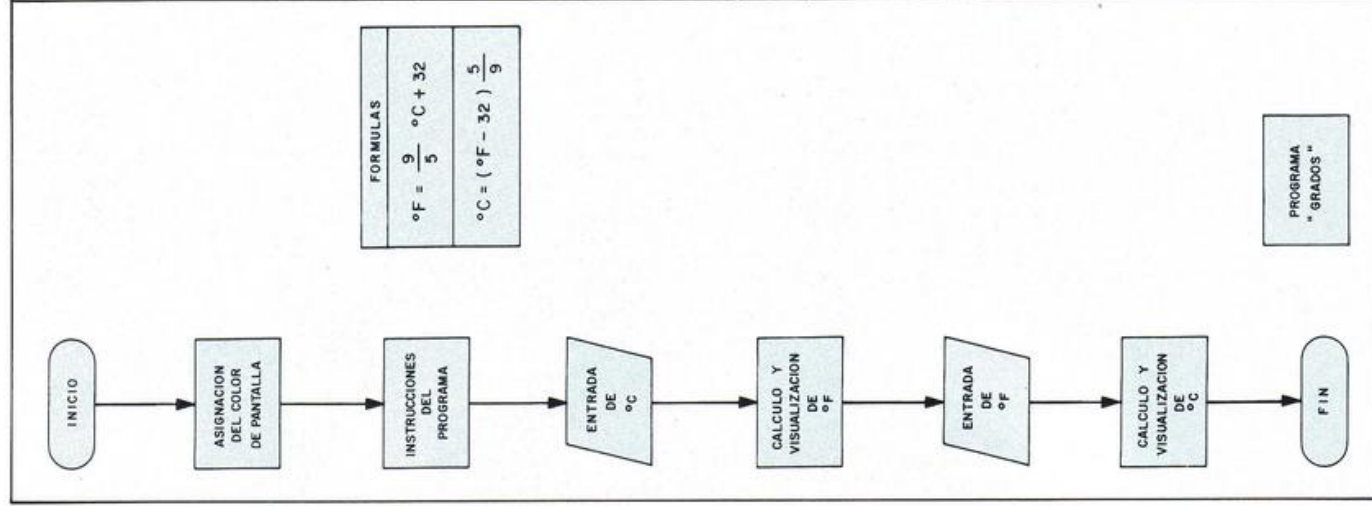
LET co = conejo * 4

Posteriormente se asigna a la variable «patas» la suma de las variables «pat», «ga», «co», «pal» y «ce».

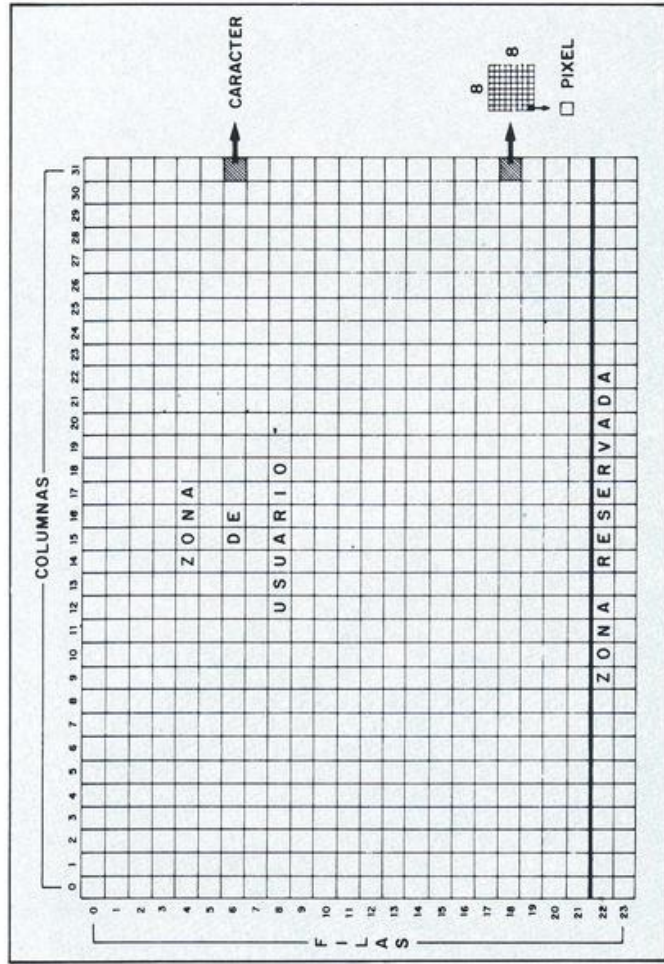
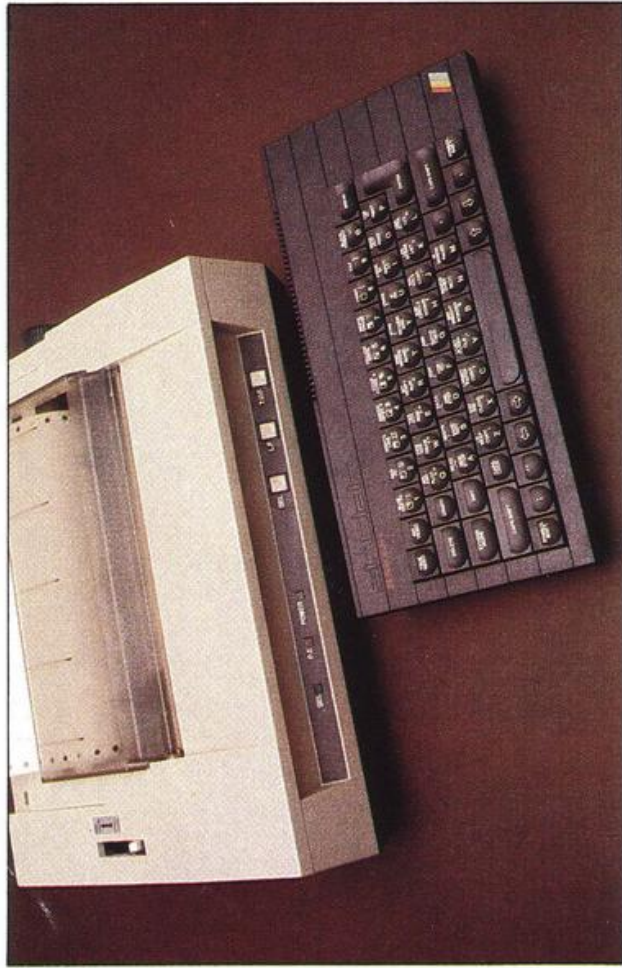
Por último el valor de la variable «patas» se visualiza con el «PRINT AT» de la línea 40φ.



Programa "Interés".



Programa "Grados".



- 2φ — 7φ : Entrada y visualización de los contenidos de las variables «capitales», «réditos» y «tiempo».
- 9φ : Cálculo de los intereses. A la variable «interés» se le asigna el resultado de la fórmula.
- 11φ : Visualización de los resultados.

Programa «GRADOS»

Almacenarlo en cinta, por ejemplo, de la forma:

SAVE "grados"

El programa «GRADOS» consta de dos partes, en la primera transforma un valor de grados centígrados (°C), introducido por teclado, en grados Fahrenheit (°F) de acuerdo con la fórmula:

para los caracteres.

La variable «C» contiene los grados centígrados a transformar y la variable «fahrenheit» el resultado.

En la segunda parte hace la transformación inversa, es decir, transforma un valor de grados Fahrenheit en centígrados, la fórmula implementada en este caso es:

$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) \cdot \frac{5}{9}$$

La variable «f» contiene los grados Fahrenheit a transformar y la variable «centígrados» el resultado.

El programa ha sido estructurado de la siguiente manera:

- 1φ : Comentario con el nombre del programa
- 2φ : Asignación del color azul para el borde y el fondo y blanco

- 4φ — 13φ : Breve descripción del programa. En la línea 11φ se utiliza el canal de comunicación φ.
- 15φ — 16φ : Entrada y visualización de la variable «C».
- 18φ — 19φ : Cálculo y visualización del resultado en grados Fahrenheit.

- 21φ — 22φ : Entrada y visualización de la variable «f».
- 24φ — 25φ : Cálculo y visualización del resultado en grados centígrados.

Programa «FICHA»

Salvarlo de la forma habitual:

SAVE "ficha"

Zonas de visualización.

Programa «ECUACION»

Salvar el programa de la forma:

SAVE "ecuación"

Este programa calcula las dos raíces de una ecuación de segundo grado del tipo:

$$ax^2 + bx + c = 0$$

Los dos valores de «X» que cumplen esta ecuación se calculan con la fórmula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Los valores que deben introducirse para que el programa calcule dichas ecuaciones son los correspondientes a las variables «a», «b» y «c».

Las funciones de las sentencias que componen dicho programa son las siguientes:

- 1φ : Comentario con el nombre del programa.
- 2φ : Asignación de los colores de borde, papel y tinta.
- 4φ — 14φ : Primera pantalla de información.
- 15φ — 19φ : Segunda pantalla de información.
- 21φ : Introducción de los valores de las variables «a», «b» y «c».
- 23φ — 25φ : Visualización detallada de las variables.
- 27φ — 30φ : Cálculo de las dos raíces. En este programa se ha utilizado la sentencia «SQ» que calcula la raíz cua-

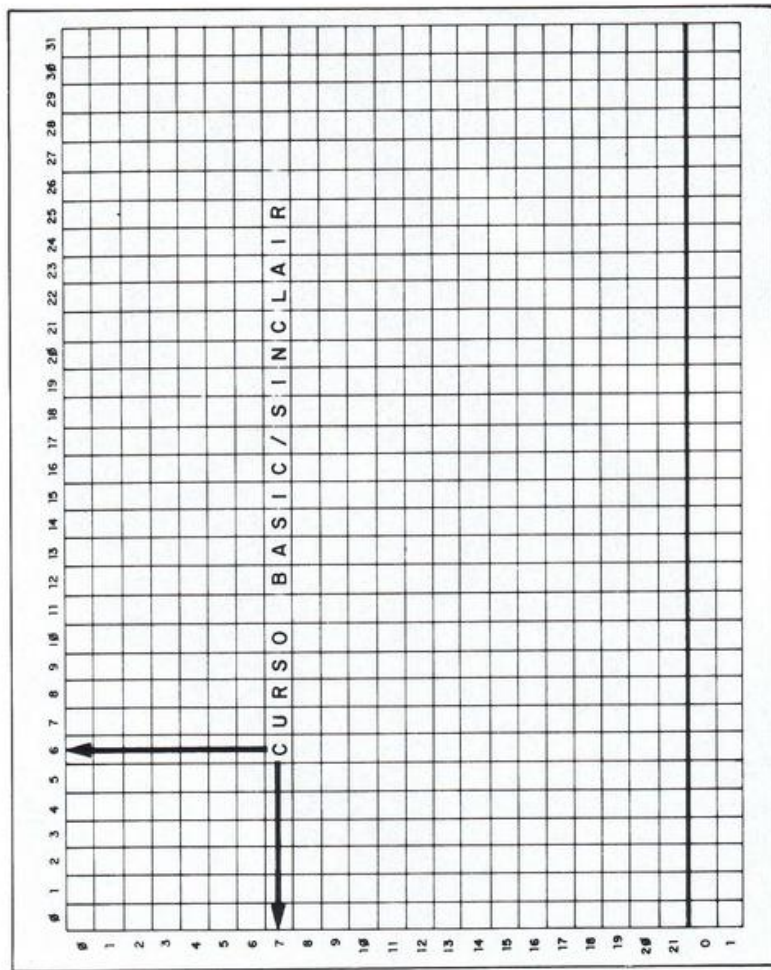
drada del argumento que va entre paréntesis. El cálculo se ha realizado en varias etapas, primeramente se han evaluado las partes comunes; a la variable «raíz» se le ha asignado el resultado de:

$$\sqrt{b^2 - 4ac}$$

y a la variable «divisor»

2a

posteriormente y a partir de estas dos variables se han obtenido las dos



Print AT 7.6.

raíces, «raíz 1» y «raíz 2».

41φ — 42φ : Visualización de los resultados.

Programa «INTERES»

Grábelo en cinta de la forma:

SAVE "Interés"

Este programa calcula el interés simple de un capital colocado en un banco durante cierto número de años. La fórmula del interés simple implementada en el programa es:

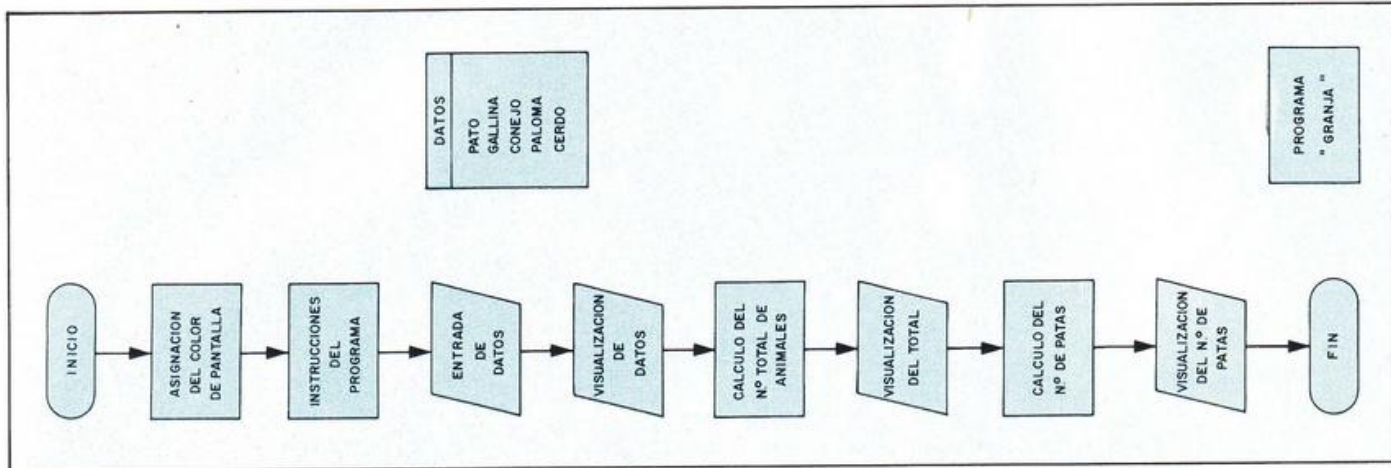
$$I = \frac{C \cdot R \cdot T}{100}$$

El valor asignado a la variable «capital» debe estar expresado en pesetas, el de la variable «reditos» en %, es decir, si el banco proporciona unos intereses al 3%, el valor a introducir deberá ser «3», y por último el asignado a la variable «tiempo» deberá ser expresado en años.

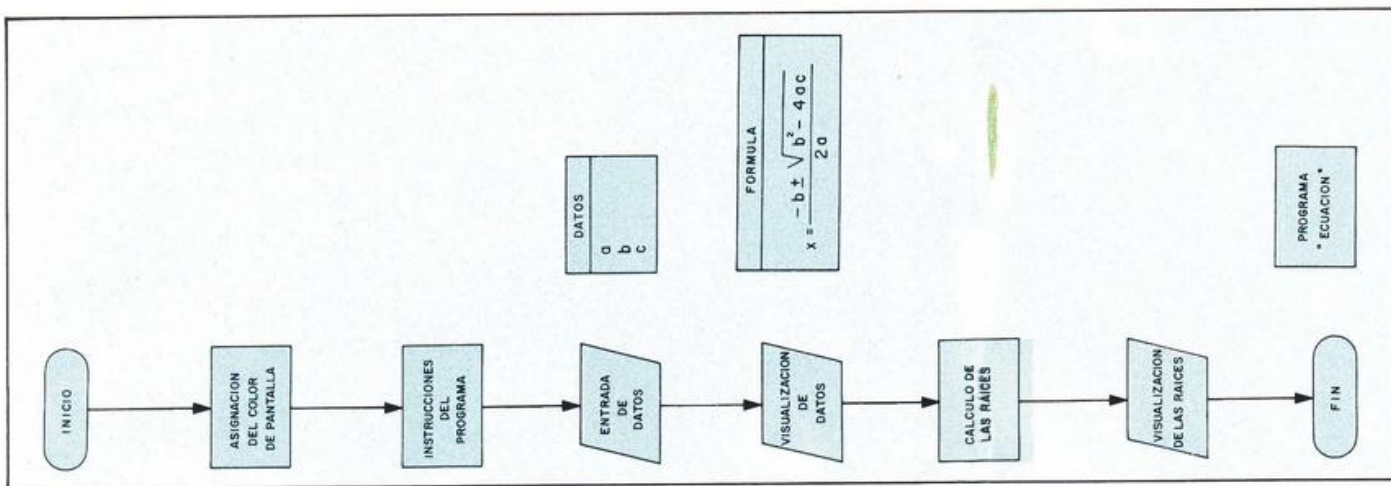
La estructura del programa es la siguiente:

- 1φ : Comentario con el nombre del programa.
- 11 : Asignación del color azul para el borde y el fondo y blanco para los caracteres.
- 13 — 18 : Visualización de una breve descripción del programa.

La segunda sentencia de la línea 18 es «PAUSE 30φ», ésta proporciona una temporización de aproximadamente seis segundos, desde que aparece la información hasta que se borra con la siguiente sentencia (CLS). Si durante la temporización se pulsa una tecla, ésta termina y se ejecuta la siguiente instrucción.



Programa «Granja».



Programa «Ecuación».

en estos dos ejemplos, el programa comienza a ejecutarse de nuevo al llegar a la última sentencia, de esta forma se crea un bucle sin fin.

El argumento también puede ser una variable numérica previamente definida.

Ejemplo:

```

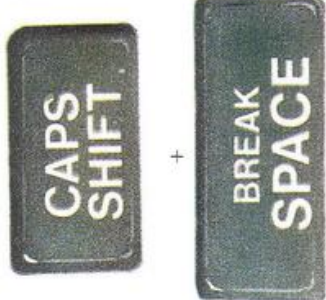
1 REM *****
2 INPUT "En que linea quieres"
3 "5 lineas de ejecucion del programa"
4 "5 lineas de ejecucion del programa"
5 PRINT "Ejecuto la 1a"
6 PRINT "Ejecuto la 2a"
7 PRINT "Ejecuto la 3a"
8 PRINT "Ejecuto la 4a"
9 PRINT "Ejecuto la 5a"
10 PRINT "Ejecuto la 6a"
11 GOTO 2

```

Una vez ejecutado el programa, éste pide que le introduzcamos el n.º de línea de la nueva ejecución, valor asignado a la variable «línea», la sentencia «RUN línea» lo hace a partir de este valor.

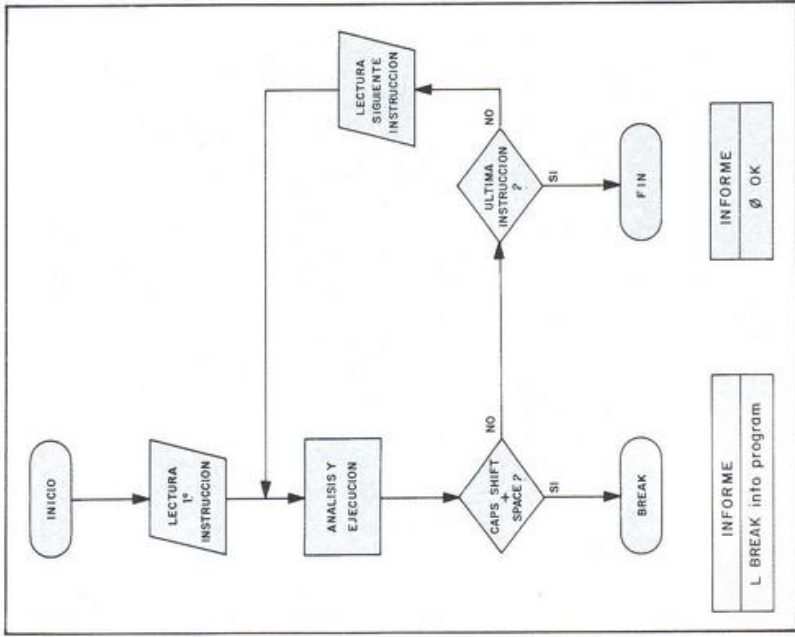
BREAK

Acceso al teclado



Definición

La función «BREAK» provoca una ruptura en la ejecución de un programa, en el acceso a los periféricos «impresora» y «cassette» y en los listados de más de 22 líneas, es decir, en



Análisis sentencia «Break».

aquellos en que aparece el mensaje:

scroll?

Esta ruptura sólo provoca una interrupción en la ejecución del programa, es decir, que no borra el contenido de la memoria. En la mayoría de los casos se podrá continuar con ella, utilizando el comando «CONTINUE» (CONT).

ADVERTENCIA

Si se está ejecutando una sentencia del tipo «INPUT», no se puede provocar la ruptura del programa, esto se consigue utilizando otra técnica que posteriormente será descrita.

```

1Ø BEEP 2, 1Ø
2Ø BEEP 3, 2Ø
3Ø BEEP 1, 15
4Ø BEEP 3, 5

```

TIPO DE "INPUT"	METODO
NUMERICO	STOP
ALFANUMERICO	ELIMINAR COMILLAS DELETE EDIT
LINE	+ STOP CAPS SHIFT + 6

Ruptura en «Input».

«CONTINUE» repite la ejecución en la misma línea donde se provocó la interrupción.

Ejemplo:

Introduzca las siguientes líneas

```

1Ø LOAD " "
2Ø PRINT "cargado"

```

cuando está en la fase de «carga» pulse la tecla «SPACE». Una vez interrumpido teclee «CONT», el programa volverá a ejecutar la sentencia «LOAD».

Si el programa se ha interrumpido debido a un error, podemos subsanar el problema momentáneamente y continuar con la ejecución.

Ejemplo:

```

10 REM *****
20 INPUT "Coordenada x? ", x
30 PRINT x
40 INPUT "Coordenada y? ", y
50 PRINT y
60 GOTO 20

```

Este programa una vez ejecutado e introducido el valor de la «coordenada X» presenta un fallo en la línea 30, ya que no está definida la variable «Z». Defínala con un comando directo como por ejemplo:

```
LET Z = 5
```

y tecleando «CONTINUE» volverá a ejecutarse el programa a partir de la línea 30

«CONTINUE» no se puede emplear con comandos directos. Se pueden distinguir tres casos:

a) Cuando se interrumpe en la primera sentencia.

Ejemplo:

```
LOAD " " : LET a = 2 : PRINT a
```

si se pulsa «SPACE» (BREAK) y se pretende continuar con la ejecución de las restantes sentencias, el programa pierde el control y se queda un bucle sin fin, para salir de él pulse la tecla «CAPS SHIFT» + «SPACE».

b) Cuando se interrumpe en la segunda sentencia.

Para borrar las comillas, existen dos métodos. El primero es utilizando la función «DELETE», de esta forma desaparece la comilla situada a la izquierda del cursor. A partir de este instante ya se puede introducir la sentencia «STOP»; el mensaje que aparece es también:

H STOP in INPUT

En el segundo método se utiliza la función «EDIT», de esta manera desaparecen ambas comillas y al igual que en el caso anterior, se puede introducir «STOP».

Para cortar un «INPUT LINE», la filosofía es totalmente distinta, ya que ni se permite introducir «STOP», ni eliminar las comillas, simplemente por que estas no existen.

Ejemplo:

```
10 INPUT " "; LINE AS
40 PRINT AS
30 GO TO 10
```

La única manera de salir del programa anterior, una vez ejecutado, es utilizando el cursor de desplazamiento inferior («CAPS SHIFT» + «6»). El mensaje presentado, al igual que en las anteriores situaciones es:

H STOP in INPUT

CONTINUE

Acceso al teclado

L PRINT



PAPER

```
100 PRINT AT 3,0;"Este programa
se autodestruirá al llegar
a cuenta 00:00"
110 REM
*****
* CUENTA ATRAS *
*****
*****
*****
120 FOR m=9 TO 0 STEP -1
130 PRINT AT 12,13;"0";m
140 FOR s=59 TO 0 STEP -1
142 IF s<10 THEN PRINT AT 12,15
;0 s: GO TO 155
150 PRINT AT 12,15;s
155 BEEP 0.02,20
160 NEXT s
160 REM
*****
* !!! BOOM !!! *
*****
*****
*****
190 FOR n=0 TO 7
200 BORDER n
210 BEEP 0.03,n
220 NEXT n
230 CLS
235 FOR n=1 TO 50
240 PRINT OVER 1;AT 10,8;"!!!!
BOOM !!!"
245 NEXT n
250 REM
*****
* !!! NEW !!! *
*****
*****
*****
250 NEW
```

Definición

El comando directo «CONTINUE» se reconoce en el teclado por su forma abreviada «CONT». La utilidad de este comando es continuar con la ejecución de un programa que, debido a un informe de error o a un «BREAK» se ha interrumpido. Este comando no precisa argumento.

Cuando la interrupción se ha debido a un «BREAK» con informe:

L BREAK into program

o a una sentencia «STOP»:

9 STOP statement

el comando «CONTINUE» comienza a ejecutar el programa a partir de la siguiente instrucción, en que se produjo la ruptura.

Sin embargo, cuando el informe presentado por «BREAK» es:

D BREAK — CONT repeats

o se visualiza:

H STOP in INPUT

Ejecute estas sentencias y utilice la función «BREAK».

Dependiendo de la situación en que se utilice «BREAK», existen dos tipos de informes. Cuando se utiliza para interrumpir un programa, el informe visualizado en pantalla es:

L BREAK into program

En los restantes casos, con sólo mantener pulsada la tecla «SPACE» (BREAK) se consigue la interrupción, y el mensaje presentado por el ordenador es:

D BREAK — CONT repeats

La diferencia entre estos dos mensajes será explicada con detalle al tratar la sentencia «CONTINUE» (CONT).

Al final del informe aparece la línea y el número de sentencia, dentro de la línea donde se produjo la interrupción.

STOP

Acceso al teclado



READ



Tipo de sentencia

Comando de programación.

Ejemplo sentencia «Stop».

Definición

A pesar de ser un comando de programación, «STOP» es tratado en este capítulo, ya que su función es de control, cuando se ejecuta se interrumpe el programa.

«STOP» no tiene sentido como comando directo, por lo que debe ser editado como línea de programa sin argumento.

Ejemplo:

400

STOP

cualquier el intérprete BASIC analiza esta sentencia, se detiene en la línea 400 y presenta el mensaje:

9 STOP statement, 400

Esta sentencia puede ser utilizada en diversas ocasiones, pero principalmente:

a) Para separar diversas rutinas independientes dentro de un solo programa.

Ejemplo:

Edite el programa «1» que simula una calculadora básica.

Las sentencias «STOP», separan las rutinas de «suma», «resta», «multiplicación» y «división». Para acceder a las diversas rutinas se utiliza la sentencia «RUN» y como argumento la variable «código», el valor de esta se asigna con el «INPUT» de la línea «8».

b) Para separar un programa principal de las *subrutinas*. Estas se estudiarán en otro capítulo.

c) Cuando se desea que se interrumpa la ejecución de un programa en función del resultado de una comparación. Se utiliza conjuntamente con el par de sentencias «IF-THEN».

Ejemplo:

```
10 REM *****
11 INPUT "Numero 1?";a1
12 INPUT "Numero 2?";a2
13 IF a1=a2 THEN STOP
14 GO TO 10
```

se producirá la interrupción del programa, cuando las variables «a1» y «a2» sean iguales.

d) En técnicas especiales de depuración de programas.

e) También se utiliza para provocar la ruptura de un programa, en una sentencia del tipo «INPUT».

Ruptura del «INPUT»

Para interrumpir la ejecución de un programa en una sentencia «INPUT», es necesario utilizar una serie de técnicas, dependiendo estas del tipo de «INPUT»:

- INPUT numérico.
- INPUT de cadena.
- INPUT LINE.

PROGRAMA 1

```
10 REM *****
11 CURSO BASIC *****
12 *****
13 CALCULADORA *****
14 *****
15 BORDER 4: PAPER 4: INK 1: C
16 30 REM *****
17 OPCIONES *****
18 *****
19 PRINT AT 3,10;"CALCULADORA"
20 PRINT AT 7,4;"CODIGO" OPE
21 PRINT AT 8,4;"
22 PRINT AT 10,6;"14" SUMA
23 PRINT AT 12,6;"22" REST
24 PRINT AT 14,6;"30" MULT
25 PRINT AT 16,6;"38" DIVI
26 INPUT "Introduzca código de
27 operación ";codigo
28 RUN codigo*10
29 *****
30 SUMA *****
31 *****
32 PRINT AT 3,13;"SUMA"
33 INPUT "Sumando 1?";suma1
34 INPUT "Sumando 2?";suma2
35 LET suma=suma1+suma2
36 PRINT suma;" + ";suma2;" =
37 suma
38 STOP
39 REM *****
40 RESTA *****
41 *****
42 PRINT AT 3,12;"RESTA"
43 INPUT "Minuendo?";min
44 INPUT "Sustraendo?";sus
45 LET resta=min-sus
46 PRINT min;" - ";sus;" = ";r
47 esta
```

```
280 STOP
290 REM *****
300 PRINT AT 3,8;"MULTIPLICACION"
310 INPUT "Multiplicando?";mul
320 INPUT "Multiplicador?";mul
330 LET multiplicacion=mul*mul
340 CLS
350 PRINT multiplicacion
360 STOP
370 REM *****
380 PRINT AT 3,12;"DIVISION"
390 INPUT "Dividendo?";divi
400 INPUT "Divisor?";divi
410 LET division=divi/divi
420 CLS
430 PRINT division;" / ";divi;" = ";
440 STOP
```

PROGRAMA 2

```
10 REM *****
11 CURSO BASIC *****
12 *****
13 !!! NEW !!! *****
14 *****
15 BORDER 5: PAPER 5: INK 1: C
16 30 REM *****
17 DIBUJO CRONO *****
18 *****
19 PRINT AT 10,11;"PAPER 2: IN
20 FLASH 1;
21 FOR n=11 TO 13
22 PRINT AT n,11; PAPER 2: INK
23 FLASH 1;
24 NEXT n
25 PRINT AT 14,11;"PAPER 2: IN
26 FLASH 1;
27 PRINT AT 12,13;"10:00"
```

Cuando son del tipo numérico basta simplemente con teclear la sentencia «STOP» (SYMBOL SHIFT + A) y «ENTER», inmediatamente se provoca la ruptura del programa y aparece el mensaje:

H STOP in INPUT

Ejemplo:

```
10 INPUT ">";a
20 PRINT a
30 GO TO 10
```

Si intenta utilizar la función «BREAK», observará que no sirve en estos casos.

Puede obtener la ruptura también, de una forma menos elegante, tecleando letras aleatoriamente, de esta manera el intérprete BASIC al analizar la entrada de datos y comprobar que no corresponde con un valor numérico o con una variable previamente definida, visualizará el mensaje:

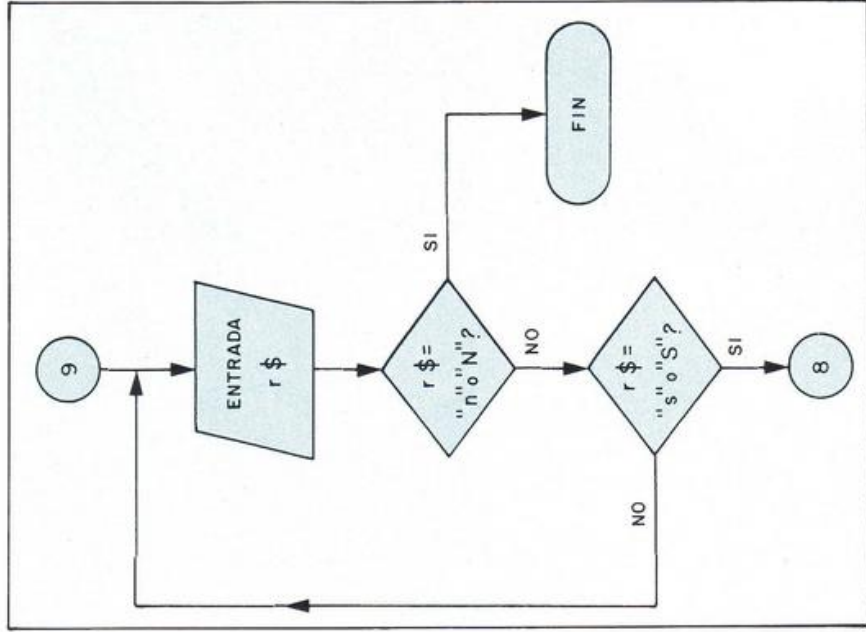
2 Variable not found

Cuando el «INPUT» es del tipo alfanumérico, la técnica es ligeramente distinta.

Ejemplo:

```
10 INPUT "? ";a$
20 PRINT a$
30 GO TO 10
```

Intente introducir «STOP», observará que el programa no se interrumpe, ya que la variable alfanumérica «a\$» asume el código correspondiente al token «STOP», y se ejecuta la siguiente instrucción, y así sucesivamente. La única manera de introducir «STOP», sin que lo tome como valor alfanumérico, es eliminando las comillas ("").



Programa "Areas" Rutina "Fin".

teriormente, por parejas hasta que sólo quede una condición, si el resultado es « ϕ » el ordenador pasará a ejecutar la siguiente instrucción; si es «1», ejecutará antes las instrucciones contenidas en el «THEN...».

Ejemplo:

```
IF (a > b OR c = 7) AND (t < >
 $\phi$  AND p = a * 5) THEN ...
```

para los valores:

```
a = 4
b = 2
c = 6
t = 5 $\phi$ 
p = 2 $\phi$ 
```

observamos que el resultado es «1», ya que se cumple una de las condiciones. (Repasar el capítulo dedicado a «OPERADORES LÓGICOS»).

```
1 OR  $\phi$ 
```

Resolviendo por pasos:

a) (a > b) OR (c = 7)
(a > b) es verdadero (1), ya que «4» es mayor que «2», (c = 7) es falso (ϕ), ya que «6» no es igual a «7».

Realizando la operación lógica «OR» de los valores anteriores

b) (t < > ϕ) AND (p = a * 5)
(t < > ϕ) es verdadero (1), ya que «5 ϕ » es distinto de « ϕ », (p = a * 5) es verdadero (1), por que «2 ϕ » es igual a «4» por «5». Realizando la operación lógica «AND» de los valores anteriores «1» y «1»

```
1 AND 1
```

tenemos un resultado igual a «1» ya que se cumplen las dos condiciones.

c) (condición 1) AND (condición 2)
Como las dos condiciones se cumplen, el resultado global es también verdadero «1».

Asigne otros valores a las variables e intente resolver el resultado; si tiene algún problema o desea comparar los resultados ejecute el siguiente programa:

```
10 REM *****
20 INPUT "a" ; a
30 INPUT "b" ; b
40 INPUT "c" ; c
50 INPUT "t" ; t
60 INPUT "p" ; p
70 IF (a > b) OR (c = 7) AND (t < >
AND p = a * 5) THEN PRINT "VERDAD"
80 GO TO 20
90 GO TO 30
```

El valor de una condición también puede ser asignado a una variable, de la forma:

```
LET resultado = f$ = "FIN"
```

la variable «resultado» tendrá el valor «1» cuando f\$ sea igual a la cadena «FIN», y « ϕ » cuando f\$ tenga otro valor. Podríamos, por tanto, editar una sentencia del tipo:

```
IF resultado THEN ...
```

NEW

Acceso al teclado



MODO K

Definición

Generalmente este comando se utiliza de forma directa y no precisa de ningún argumento para poderse ejecutar. «NEW» borra el programa o programas almacenados en memoria, también borra el valor de las variables definidas. Hay una serie de *variables de sistema* que no se ven afectadas por este comando, entre ellas los GDU o gráficos definidos por el usuario.

Debe utilizarse con mucho cuidado ya que de lo contrario, podríamos borrar un programa que aún no ha sido salvado, cosa no muy agradable por cierto.

Cuando se ejecuta da la im-

presión de haber conectado el aparato de nuevo, ya que nos presenta el famoso mensaje inicial:

© 1982 Sinclair Research Ltd.

Puede incluirse con precaución dentro de un programa para dar por finalizada su ejecución y borrado.

El programa n.º «2» incorpora esta sentencia, «sálvelo antes de ejecutar».

CLS

Acceso al teclado



MODO K

Definición

El comando «CLS» puede ser utilizado tanto en modo di-

recto, como formando parte de un programa; no precisa de ningún argumento.

La función de este comando es borrar la pantalla de caracteres y gráficos, asumiendo ésta el color especificado en la última sentencia «PAPER» ejecutada con anterioridad. El color del borde de la pantalla no se ve afectado por esta sentencia.

Ejemplos:

— Introduzca el siguiente programa:

```
10 REM *****
20 FOR i=1 TO 704
30 PRINT "a" ; i
40 NEXT i
50 REM *****
60 BORDER 2
```

este programa llena la zona de visualización con el símbolo «a», teclee el comando directo «CLS» y observe la pantalla.

En el siguiente programa la sentencia «CLS» se utiliza para borrar la pantalla y asumir los colores «magenta» para el fondo y «amarillo» el de los caracteres; el color del borde se asigna directamente con la sentencia «BORDER».

```
10 REM *****
20 ASIGNAR COLORES *****
30 BORDER "verde" **
40 BORDER FONCO "magenta" **
50 BORDER TINTA "amarillo" **
60 CLS
70 PRINT AT 10,10:"MICROHOBBY"
```

Ejemplo de listado.

```
100 IF HT > 7 THEN LET C=0: GO
125 INPUT "¿BERCON?"; X#; GO TO
130
25 NEXT Q: GO TO 25
28 LET B=B+7*(B=C
GO SUB 2510: INPUT "RADIAL?";
RD:
30 INPUT "HEADING?"; HD: LE
HD=HD*PI/180:
40 LET S=SF#SIN HD+W$#COS
LET W=W+SF#SIN HD+W$#COS
50 LET HD=HD+RL/(SF#7E3): IF
D>CR THEN LET HD=HD-CR
55 IF HD < 0 THEN LET HD=HD+CR
60 TO EX
65 LET G=G-(BB+20)/M1: IF G <
0 THEN GO SUB 2514-M1
```




MODO K

Definición

«LIST» se utiliza normalmente como comando directo y permite obtener un listado del programa almacenado en memoria. La estructura general de esta sentencia es:

SENTENCIA	ARGUMENTO
LIST	N.º DE LINEA

Ejemplos:

- LIST 10
- LIST 30
- LIST

Cuando el argumento se omite, el intérprete BASIC ejecuta este comando a partir de la línea 1.

El listado del programa se visualiza en páginas de 22 líneas presentando en la parte inferior de la pantalla el mensaje:

scroll?

este mensaje, como ya recordará el lector de lo explicado anteriormente con la sentencia «PRINT», sirve para preguntarnos si queremos visualizar la siguiente página. Pulsando las teclas «N», «SPACE» o «STOP» («SYMBOL SHIFT» + «a») el listado se interrumpirá

y se nos presentará el mensaje:

D BREAK — CONT repeats

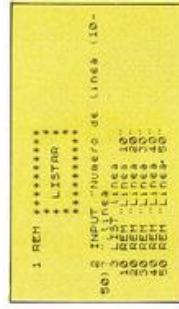
pulsando cualquier otra tecla se visualizará la siguiente página y así sucesivamente hasta que se termine el listado (mensaje «OK»).

Cuando como argumento se introduce, por error, un número decimal, el intérprete BASIC redondea este valor hasta el número entero más próximo: si tiene un programa almacenado en la memoria, ejecuta estos dos comandos directos y observe los resultados:

LIST 10.2
LIST 10.5

Si se especifica un número de línea inexistente, el comando «LIST» empezará a ejecutarse a partir de la siguiente.

El argumento también puede ser una variable numérica previamente definida. En el siguiente ejemplo, la instrucción «LIST linea» visualiza el listado a partir del valor asignado a la variable «linea».



LIST Y EDIT

En el capítulo «2» dedicado a la edición de programas y

corrección de los posibles errores, se estudió un método para corregir líneas de programa una vez editadas. Este método consistía en desplazar con los cursores (y) el prompt «>» hasta situarlo en la línea que queríamos corregir; pero ¿qué pasa si tenemos el cursor en la línea 4000 y queremos corregir la 200? ¿cómo vemos este método no es efectivo ya que perderíamos mucho tiempo desplazando cursores; en estos casos resulta más interesante utilizar el comando «LIST».

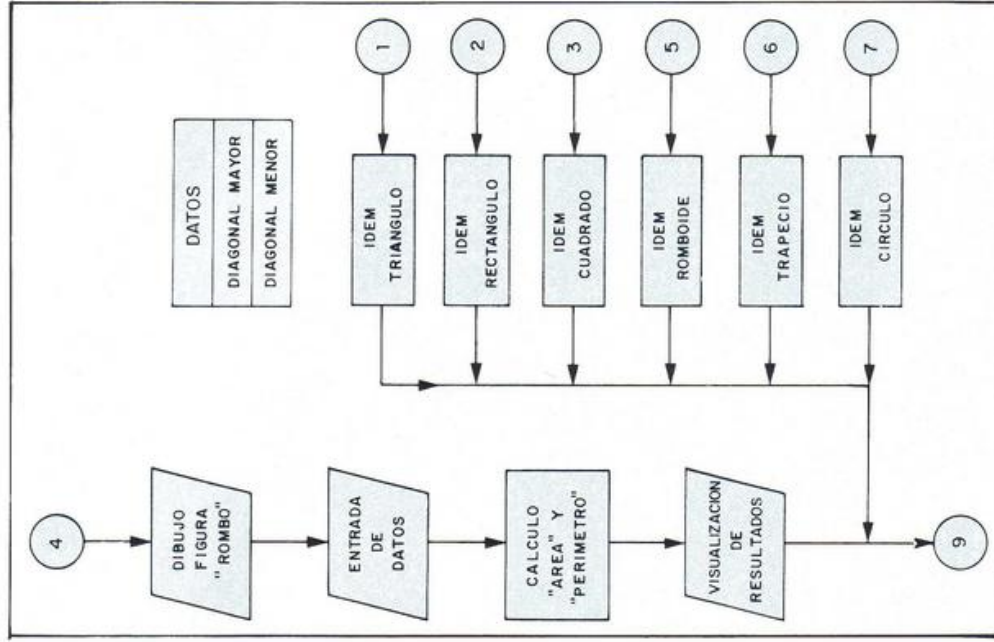
Para situar el prompt «>» en la línea que deseamos corregir basta simplemente con pedir un listado a partir de dicha línea, por ejemplo si deseamos corregir la línea 200, introduciremos el comando directo:

LIST 200

lógicamente si el listado es largo aparecerá el mensaje:

scroll?

pulsando la tecla «N», «SPACE» o «STOP» el listado se interrumpirá. El prompt ya lo tendremos situado en la línea 200; para corregirla utilizaremos la Función «EDIT», con lo que nos pasará a la parte inferior de la pantalla. A partir de ese momento podremos corregirla utilizando los cursores (→ y) y la función «DELETE». Una vez terminada la modificación, pulsando «ENTER» volverá a la parte superior. ■



Programa «Áreas» desarrollo opción «Rombo».

Los operadores lógicos «AND» y «OR» son utilizados cuando hay una combinación de condiciones dentro de una sentencia «IF ... THEN ...». El operador «AND» implica que deben cumplirse todas y cada una de las condiciones.

Ejemplo:

```
10 IF n >= 0 AND n <= 9 THEN ...
```

solamente se cumplen las dos condiciones, cuando la variable

«n» tiene un valor comprendido entre «0» y «9».

Utilizando el operador «OR» basta solamente con que se cumpla una de las condiciones previstas en la comparación: Ejemplo:

```
10 IF A$ = "A" OR B$ < "C" OR n >= 30 THEN ...
```

en este ejemplo, se cumple la condición general en cualquiera de los siguientes casos:

- Cuando la variable A\$ tenga el valor «A».
- Cuando el contenido de la variable B\$ sea distinto de «C».
- Cuando «n» sea superior a «30».
- O con cualquier combinación de los anteriores casos.

También pueden combinarse los operadores «AND» y «OR». Ejemplo:

```
10 IF (A$ = "NO" AND n < 10) OR (A$ = "SI" AND (p = 10 OR t = 7)) THEN ...
```

primero, se evalúan individualmente las condiciones encerradas entre paréntesis, y posteriormente, se evalúan entre sí los resultados parciales, por tanto, será necesario que se cumplan cualquiera de las siguientes condiciones:

- Cuando JS = «NO» y n < 10.
- Cuando JS = «SI» y p = 10.
- Cuando JS = «SI» y t = 7.
- Combinaciones de a, b y c.

El programa número «1» es un ejemplo de aplicación de las sentencias «IF ... THEN ...»; acaba su ejecución cuando se teclean dieciocho nombres con sus correspondientes teléfonos o cuando se introduce la palabra «FIN» o «fin» en el instante que el ordenador espera un nombre.

Evaluación de las condiciones

Cuando una condición se cumple, es decir, que es verdadera, se le asigna el valor «1» (distinto de 0) y cuando es falsa, el valor 0 (igual a 0). Para evaluar una condición completa, primero se evalúan una a una, asignando los valores «0» o «1» según corresponda; pos-

SALTOS INCONDICIONALES Y CONDICIONALES

Hay ocasiones en que por razones de estructura de un programa, interesa que las instrucciones no se ejecuten de forma secuencial, es decir, una detrás de otra, sino que, por el contrario realicen saltos; éstos pueden clasificarse dependiendo de su función en:

- INCONDICIONALES.
- CONDICIONALES.

Como su propio nombre indica, un salto *incondicional* es aquel que salta directamente al número de línea especificado en el argumento, sin embargo, los condicionales necesitan que se cumpla previamente la condición prevista en la instrucción.

GO TO

Acceso al teclado



Tipo de sentencia

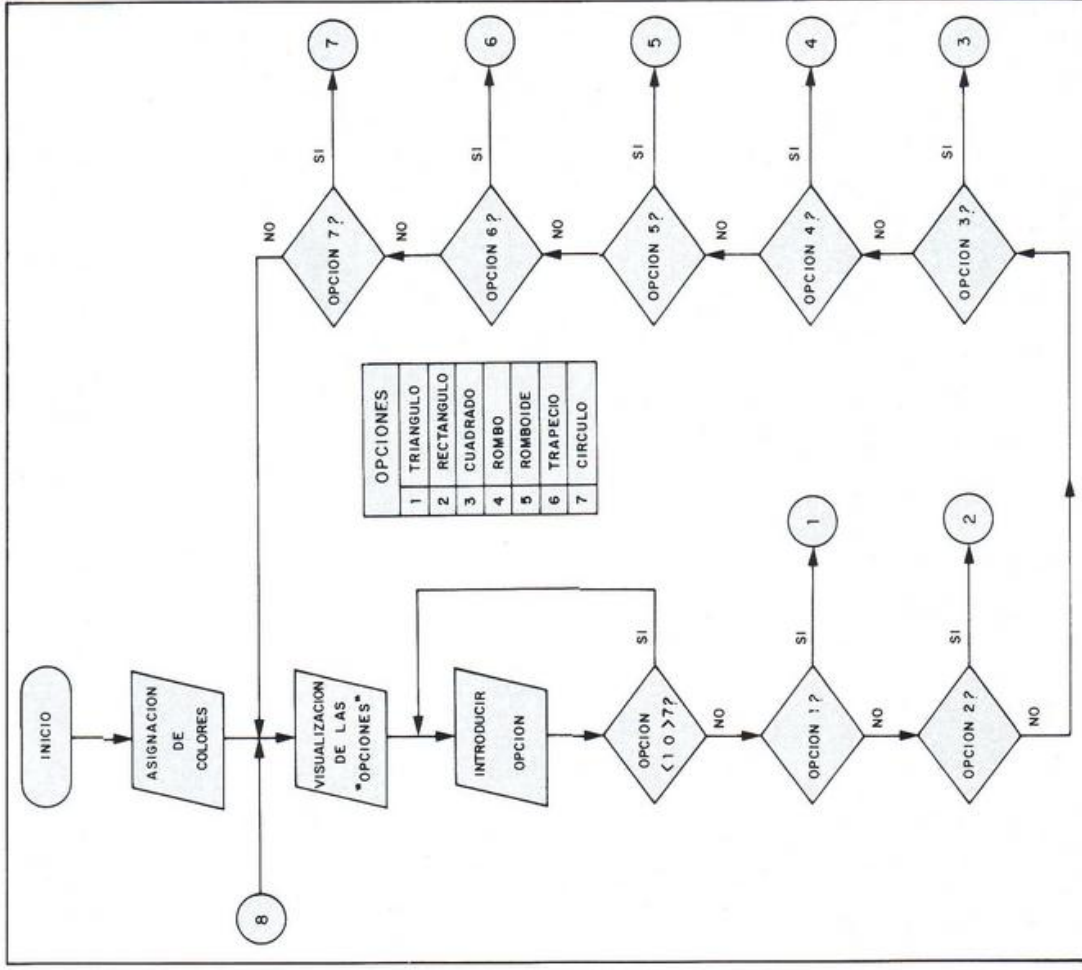
Comando de programación.

Definición

La sentencia «GO TO» realiza los saltos incondicionales dentro de un programa, su estructura general es:

SENTENCIA	ARGUMENTO
GO TO	N.º de línea

Ejemplo sentencia "GO TO".



Programa "Areas" menú de opciones.

PROGRAMA 1

```

10 REM *****
   * AGENDA *
   * *****
20 PRINT AT 0,4; "NOMBRE"
25 PRINT " "
27 LET indice=1
30 INPUT "Nombre >>> "; LINE n
$
40 IF n$="FIN" OR n$="fin" THEN
N GO TO 1000
50 PRINT n$, "telefono >>> "; telef
on 70 PRINT telefono
50 IF indice=10 THEN GO TO 100
90 LET indice=indice+1
100 GO TO 30
1000 PRINT #0; AT 1,2; "<<<< Fin
de edicion >>>>";
1010 PAUSE 0

```


Ejemplos:

- GO TO 30
- GO TO 70

Una sentencia de este tipo transfiere la ejecución del programa a la línea especificada en su argumento.

En el siguiente programa, al analizar el intérprete BASIC la instrucción «40», la siguiente que ejecutará será la «10».

```
10 REM *****
20 INPUT "Escriba lo que quiere"
30 PRINT "ENTER: >>>>"
40 GO TO 10
```

En este otro se ha incluido un índice asignado a la variable «potencia» que se incrementa en uno cada vez que se ejecuta la línea «50», este índice sirve tanto para indicar el número de veces menos uno, que se realiza la operación matemática (2ⁿ) como para utilizarse como potencia de la misma.

```
10 REM *****
20 POTENCIA = 0
30 PRINT "Potencia: "
50 LET POTENCIA = POTENCIA + 1
60 GO TO 30
```

Debe poner atención al calcular el número de línea donde desea que se realice el salto, ya que podrían no ejecutarse ciertas líneas intermedias.

Ejemplo:

```
10 REM *****
20 LET A$ = "MICRO"
30 LET B$ = "MOBY"
50 PRINT A$
70 PRINT B$
80 GO TO 60
```

en este ejemplo, solamente la primera vez se ejecuta el programa correctamente; en las siguientes, la variable «a\$» no se visualiza, por tanto la línea 80 debería ser:

GO TO 50

«GO TO» también puede ser utilizado como comando directo, esta aplicación es bastante interesante, ya que permite ejecutar un programa sin alterar el contenido de las variables hasta ese momento definidas. En la depuración de programas se utiliza frecuentemente en sustitución del comando «RUN» que lo borra todo.

El argumento puede ser una variable de tipo numérico:

```
1 REM *****
2 VARIABLE *****
3 *****
4 INPUT "R que línea hago el"
5 LINE 10, 20, 30, 40, 50, 60
6 GO TO LINE 30
7 PRINT "Salto a la 30"
8 PRINT "Salto a la 40"
9 PRINT "Salto a la 50"
10 PRINT "Salto a la 60"
11 PRINT "Salto a la 70"
12 GO TO 4
```

una vez introducida la variable «línea» el control de ejecución salta hasta el valor especificado en ella.

IF ... THEN ...

Acceso al teclado



Tipo de sentencia

Comando de programación.

Definición

El grupo de sentencias «IF» y «THEN» permiten realizar los saltos o ejecutar una serie de instrucciones de una manera condicional, es decir, en función del resultado de una comparación.

Las estructuras básicas son:

a) Salto condicional.

SENTENCIA	ARGUMENTO
IF condición	THEN GO TO...

Ejemplo:

```
10 REM *****
20 *****
30 *****
40 INPUT "Número a="
50 IF a = 0 THEN GO TO 70
60 LET b = a/2
70 PRINT "Valor a="
80 PRINT "Fin"
90 FIN
```

cuando las variables «a» y «b» sean iguales, la ejecución del programa continuará en la línea 70, si no, se asignará a la variable «a» el valor de «a + b/2» y posteriormente, ejecuta un salto incondicional a la línea 30.

b) Ejecución condicional.

SENTENCIA	ARGUMENTO
IF condición	THEN instrucciones...

Ejemplo:

```
10 REM *****
20 *****
30 PRINT "MICROHOB"
40 INPUT "Quiere continuar?"
50 IF "SI" = "NO" THEN PRINT AT 10
60 "Fin"
70 LET a = 0
80 GO TO 20
```

si no se cumple que la variable de cadena «f\$» sea igual a «NO» se ejecuta la instrucción siguiente; si por el contrario lo son, se ejecutan las sentencias que acompañan al «THEN».

De lo explicado hasta este momento, se desprende que la sentencia «IF ... THEN ...» es como una encrucijada con dos caminos, donde el ordenador tiene que elegir uno de ellos.

OBSERVACIÓN

Si se utiliza la sentencia «GO TO», dentro de la lista de instrucciones que deben ejecutarse si se cumple la condición impuesta en el «IF», ésta deberá ser colocada la última, ya que de lo contrario, quedarían sin ejecutar algunas instrucciones.

Ejemplo:

```
10 IF mes = 8 THEN PRINT
20 "AGOSTO": GO TO 120
30 LET mes = 0
```

la asignación del valor «0» a la variable «mes» no se realiza, ya que antes se ejecuta una instrucción de salto incondicional a la línea 120.

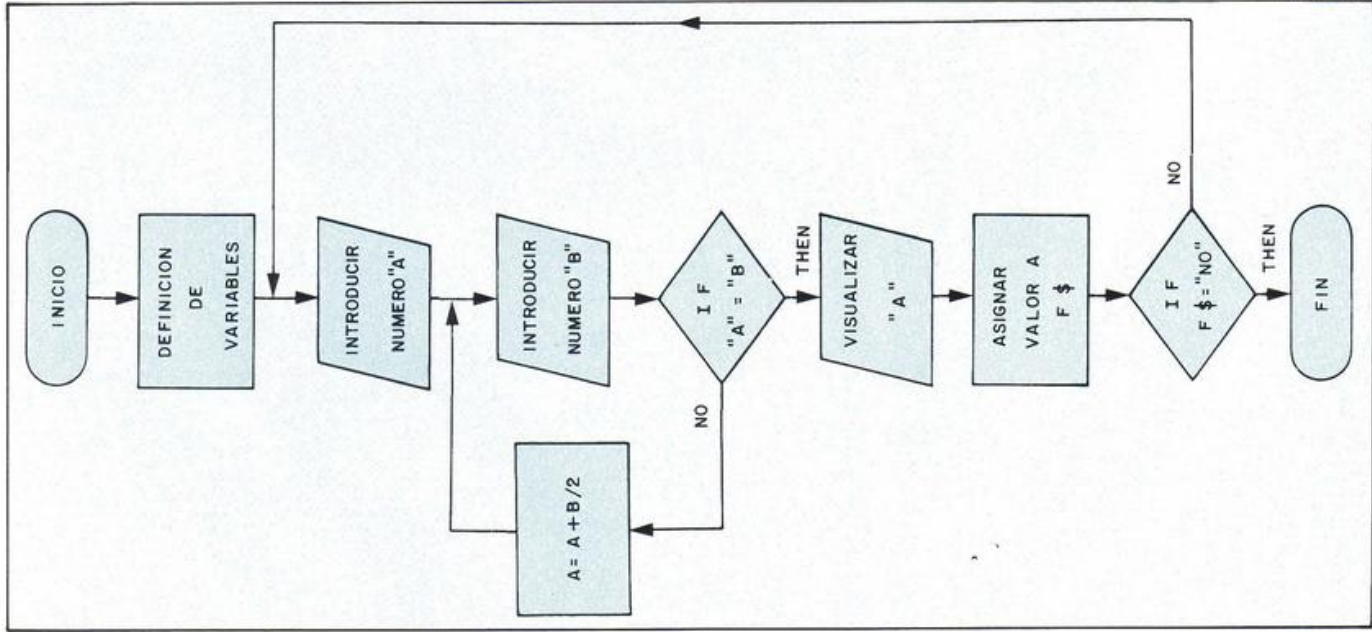
Para realizar las comparaciones, puede utilizarse cualquiera de los operadores relacionales:

>	Mayor que ...
<	Menor que ...
>=	Mayor o igual
<=	Menor o igual
<>	Distinto

Ejemplos:

- IF a > b THEN ...
- IF JS < SS THEN ...
- IF n >= k * t THEN ...
- IF kilo <= 7 THEN ...
- IF PS < > "SI" THEN ...

Ejemplo salto y ejecución condicional (IF... THEN...).



PROGRAMA 2

```

10 REM
** CURSO BASIC
** AREAS Y
** PERIMETROS
*****
20 BORDER 1: PAPER 1: INK 7: C
30 REM
** OPCIONES
*****
40 PRINT AT 1,0: "CALCULO DE AR
50 PRINT AT 4,9: "MENU DE OPCIO
60 PRINT AT 5,7: "
70 PRINT AT 8,8: "1 - TRIANGULO
80 PRINT AT 10,8: "2 - RECTANGU
90 PRINT AT 12,8: "3 - CUADRADO
100 PRINT AT 14,8: "4 - ROMBO"
110 PRINT AT 16,8: "5 - ROMBOIDE
120 PRINT AT 18,8: "6 - TRAPECIO
130 PRINT AT 20,8: "7 - CIRCUNFE
RENCIA"
140 INPUT "Introduzca la opcion
deseada >>> ",opcion
150 REM
*****
** VERIFICACION
*****
160 IF opcion<1 OR opcion>7 THE
N GO TO 140
170 CLS
180 IF opcion=1 THEN GO TO 250
190 IF opcion=2 THEN GO TO 450
200 IF opcion=3 THEN GO TO 970
210 IF opcion=4 THEN GO TO 1170
220 IF opcion=5 THEN GO TO 1380
230 IF opcion=6 THEN GO TO 1580
240 IF opcion=7 THEN GO TO 1630
250 REM
*****
** TRIANGULO
*****
260 PRINT AT 2,11: "TRIANGULO"
270 PLOT INK 4: 16,47
280 DRAW INK 4: 16,0
290 DRAW INK 4: 16,63
300 DRAW INK 4: 16,0
310 PRINT AT 17,5: "h"
320 PRINT AT 12,5: "b"
*****
** ENTRADA DATOS
*****
330 INPUT "Base >>> ",base
340 INPUT "Altura >>> ",altura
350 PRINT AT 14,13: "AREA
360 PRINT AT 14,13: "PERIMETRO:
*****
** CALCULOS
*****
370 REM
*****

```

```

380 LET area=base*altura/2
390 PRINT AT 12,11: "AREA
400 LET lado=sqr (altura^2+(bas
e/2)^2)
410 LET perimetro=lado+base
420 PRINT AT 14,11: "PERIMETRO:
430 LET perimetro=2*base+2*
lado
440 GO TO 9000
450 REM
*****
** RECTANGULO
*****
460 PRINT AT 2,11: "RECTANGULO"
470 PLOT INK 4: 16,47
480 DRAW INK 4: 16,0
490 DRAW INK 4: 16,63
500 DRAW INK 4: 16,0
510 PRINT AT 17,5: "h"
520 PRINT AT 12,5: "b"
530 PRINT AT 10,0: "h"
540 REM
*****
** ENTRADA DATOS
*****
550 INPUT "Base >>> ",base
560 INPUT "Altura >>> ",altura
570 PRINT AT 14,13: "AREA
580 PRINT AT 14,13: "PERIMETRO:
*****
** CALCULOS
*****
590 REM
*****
610 LET area=base*altura
620 PRINT AT 12,13: "AREA
630 LET perimetro=base+2*altura
640 PRINT AT 14,13: "PERIMETRO:
650 GO TO 9000
660 REM
*****
** CUADRADO
*****
670 PRINT AT 2,13: "CUADRADO"
680 PLOT INK 4: 16,47
690 DRAW INK 4: 16,0
700 DRAW INK 4: 16,63
710 PRINT AT 17,5: "h"
720 PRINT AT 12,5: "b"
*****
** ENTRADA DATOS
*****
730 INPUT "Lado >>> ",lado
740 PRINT AT 14,13: "AREA
750 PRINT AT 14,13: "PERIMETRO:
*****
** CALCULOS
*****
760 REM
*****
920 LET area=lado^2
930 PRINT AT 12,13: "AREA
940 LET perimetro=lado*4
950 PRINT AT 14,13: "PERIMETRO:
960 GO TO 9000
970 REM
*****

```

INCORRECTO

CORRECTO

Anidamiento de bucles.

Cuando «STEP» se omite, el intérprete BASIC toma por defecto el valor «1», como se ha visto en los ejemplos anteriores, que la variable de control asuma inicialmente el valor del límite inferior y se incrementaba en uno hasta alcanzar el del límite superior. Su estructura es la siguiente:

SENTENCIA	ARGUMENTO
FOR	... TO STEP expresión

Ejemplo:

```
FOR j = 8 TO 32 STEP 4
```

la variable de control «j» tomará el valor «8» y se incrementará de cuatro en cuatro hasta alcanzar el valor «32», es decir, 8-12-16-20 ... 32. El bucle se realiza siete veces.

Ejemplo:

```

10 REM
** STEP
**
30 INPUT "Paso >>> ",paso
30 FOR n=8 TO 100 STEP paso
40 PRINT "Variable de control:
50 NEXT n

```

ERROR si el valor inicial es inferior al final o si el paso es mayor o igual a cero.

Bucles anidados

Se denomina bucle anidado, aquel que contiene otro en su interior; no existe límite en cuanto a la cantidad de bucles que se pueden anidar.

Los bucles deben estar completamente encajados, por lo que no deben solaparse.

Ejemplo:

```

10 REM
** ANIDADO
**
20 FOR a=1 TO 5
30 FOR b=1 TO 5
40 PRINT "Bucle 1: ",a
50 PRINT "Bucle 2: ",b
60 PRINT " "
70 NEXT b
80 NEXT a

```

El bucle interior «b» se ejecuta 50 veces aunque sus límites van del 1 al 10, ya se multiplica por los del exterior «a». El programa visualiza los valores de las dos variables de

este programa sólo se ejecuta con bucles decrecientes, ya que presenta un mensaje de


```

*****
** ROMBO **
*****
900 PRINT AT 2,13;"ROMBO"
990 PRINT
1000 PLOT INK 4,13,40
1010 DRAW INK 4,13,40
1020 DRAW INK 4,13,48
1030 DRAW INK 4,13,48
1040 DRAW INK 4,13,48
1050 REM
*****
** ENTRADA DATOS **
*****
1060 INPUT "Diagonal mayor >>> "
1070 PRINT AT 7,13;"D ";mayor
1080 INPUT "Diagonal menor >>> "
1090 PRINT AT 9,13;"d ";menor
1100 REM
*****
** CALCULOS **
*****
1110 LET area=mayor*menor/2
1120 PRINT AT 12,12;"AREA "
1130 LET lado=sqr ((mayor/2)^2+(menor/2)^2)
1140 PRINT AT 14,12;"PERIMETRO: "
1150 LET perimetro=lado*4
1160 GO TO 9000
1170 REM
*****
** ROMBOIDE **
*****
1180 PRINT AT 3,13;"ROMBOIDE"
1190 PRINT
1200 PLOT INK 4,13,47
1210 DRAW INK 4,13,47
1220 DRAW INK 4,15,40
1230 DRAW INK 4,15,40
1240 DRAW INK 4,15,40
1250 PRINT AT 17,13;"b ";lado
1260 PRINT AT 19,13;"h ";lado
1270 REM
*****
** ENTRADA DATOS **
*****
1280 INPUT "Base >>> "
1290 PRINT AT 7,13;"B ";base
1300 INPUT "Altura >>> "
1310 PRINT AT 9,13;"h ";altura
1320 INPUT "Lado >>> "
1330 PRINT AT 11,13;"l ";lado
1340 REM
*****
** CALCULOS **
*****
1350 LET area=base*altura
1360 PRINT AT 14,13;"AREA "
1370 LET perimetro=lado*2+base*2
1380 PRINT AT 16,13;"PERIMETRO: "
1390 GO TO 9000
1400 REM
*****
** TRAPECIO **
*****
1410 PRINT AT 3,13;"TRAPECIO"
1420 PLOT INK 4,13,47
1430 DRAW INK 4,13,47
1440 DRAW INK 4,15,64
1450 DRAW INK 4,15,64
1460 DRAW INK 4,15,64
1470 PRINT AT 17,13;"B ";base
1480 PRINT AT 19,13;"b ";base
1490 REM
*****
** ENTRADA DATOS **
*****
1500 INPUT "Base mayor >>> "
1510 PRINT AT 7,13;"B ";mayor
1520 INPUT "Base menor >>> "
1530 PRINT AT 9,13;"b ";menor
1540 INPUT "Altura >>> "
1550 PRINT AT 11,13;"h ";altura
1560 REM
*****
** CALCULOS **
*****
1570 LET area=(mayor+menor)*altura/2
1580 PRINT AT 14,13;"AREA "
1590 LET lado=sqr (((mayor-menor)/2)^2+altura^2)
1600 LET perimetro=mayor+menor+2*lado
1610 PRINT AT 16,13;"PERIMETRO: "
1620 GO TO 9000
1630 REM
*****
** CIRCUNFERENCIA **
*****
1640 PRINT AT 2,9;"CIRCUNFERENCIA"
1650 PRINT AT 3,9;" "
1660 PRINT
1670 CIRCLE INK 4,40,88,32
1680 PRINT AT 9,13;"r ";radio
1690 REM
*****
** ENTRADA DATOS **
*****
1700 INPUT "Radio >>> "
1710 PRINT AT 7,13;"r ";radio
1720 REM
*****
** CALCULOS **
*****
1730 LET area=pi*r^2
1740 PRINT AT 10,12;"AREA "
1750 LET perimetro=2*pi*radio
1760 PRINT AT 12,12;"PERIMETRO: "
1770 GO TO 9000
1780 REM
*****
** MAS CALCULOS? **
*****
9010 INPUT "Quiere continuar (S/N) >>> "
9020 IF r$="n" OR r$="N" THEN STOP
9030 IF r$="s" OR r$="S" THEN CLS
9040 GO TO 9010

```

```

Ejemplo:
10 REM *****
      MOD IF 2
      *****
20 LET I=1
   DO WHILE I<=10
      LET S=I+2
      LET I=I+1
   LOOP
   PRINT S
   PRINT "Total: ",I,"Llaves"

```

A pesar de las líneas 50 y 60 que modifican los límites, la variable «f» asume los valores iniciales 1 a 20.

Dentro de un bucle «FOR/NEXT», se puede incluir una instrucción del tipo «IF ... THEN ...» que si se cumple la condición, se produzca una ruptura en su ejecución, aumentando la variable de control no haya alcanzado su valor final.

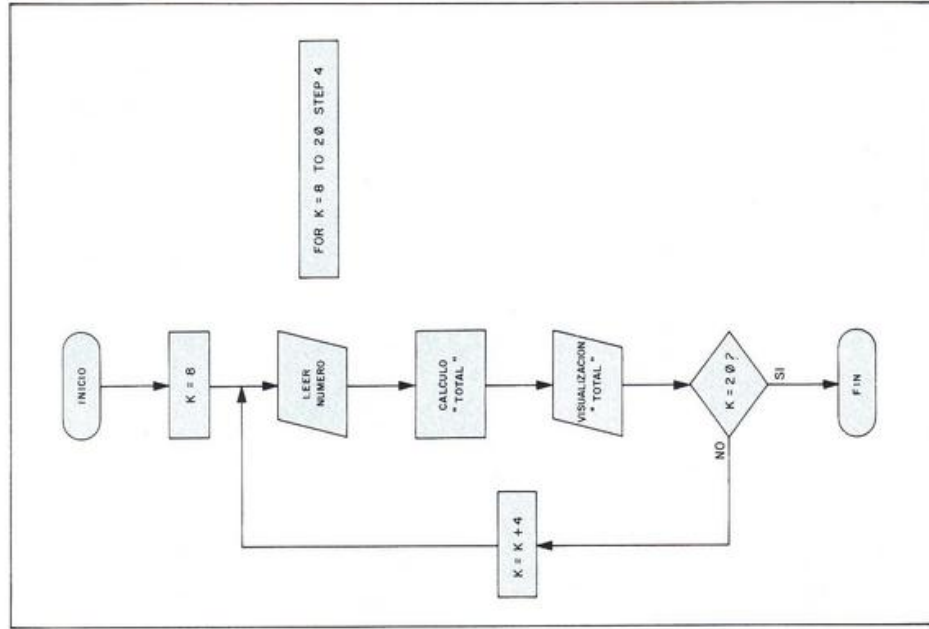
Este tipo de estructura de bucle es similar a la que, en otros lenguajes de más alto nivel, se denomina «DO WHILE/BREAK».

```

10 REM *****
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****

```

si se introducen valores de radio distintos de « ϕ », el bucle se ejecuta un número de veces que coincide con los parámetros especificados (15 veces); si por el contrario, se introduce el *código de ruptura*, que en este caso es igual a « ϕ »; la ejecución pasa a la línea 10 sin haber concluido el bucle.



Desarrollo de un bucle FOR-NEXT.

Tipo de sentencia

Acceso al teclado

Comando de programación.

Definición

La palabra clave «STEP» se maneja conjuntamente con la sentencia «FOR ... TO...», formando parte de su argumento. Se utiliza para modificar el paso de un bucle, es decir, el incremento que se realiza a la variable de control cuando se ejecuta la sentencia «NEXT» correspondiente.



za la palabra clave «TO», cuyo acceso al teclado es:



SGN



La estructura de «NEXT» tiene el siguiente formato:

SENTENCIA	ARGUMENTO
NEXT	variable

Ejemplos:

```
10 FOR I = 2 TO 10
```

```
50 NEXT I
```

En palabras sencillas, la explicación del ejemplo anterior es:

Ejecutar las instrucciones siguientes al «FOR ... TO ...» hasta que la variable «I», pasando de «2», tome el valor «10»; la sentencia «NEXT» se encarga de incrementar el valor de esta variable; el bucle se repite por tanto, nueve veces.

La variable de control del bucle, sólo puede estar formada por una letra, si existe otra variable con el mismo nombre, ésta es borrada y asume el nuevo valor. La expresión anterior al «TO» es el valor inicial que debe tomar la variable de control, y la expresión posterior el valor final. A diferencia de otros lenguajes BASIC, la

sentencia «NEXT» debe incluir como argumento el nombre de la variable de control; por tanto *no puede omitirse*.

Ejemplo:

```
10 REM *****
   : FOR...NEXT *****
   :
20 FOR N=1 TO 61
40 PRINT "MICROHOBBY ";
50 NEXT N
```

En el programa anterior la cadena alfanumérica «MICROHOBBY» es visualizada 61 veces; se utiliza para formatear, el signo ortográfico «;».

La variable de control puede ser incluida en el grupo de sentencias que forman el bucle.

Ejemplo:

```
10 REM *****
   : CUADRADOS *****
   :
30 FOR C=100 TO 199
40 PRINT "Numero: ";C
50 LET A=C*CUADRADO: "A
60 NEXT C
```

la variable de control «C» que varía entre 100 y 199 es elevada al cuadrado, asignando este valor a la variable «A»; ambas variables son visualizadas. Los límites, al ser expresiones de tipo numérico, pueden estar constituidos por variables previamente asignadas.

Ejemplo:

```
10 REM *****
   : PARAMETROS *****
   :
20 INPUT "Limite inferior >>>";
30 INPUT "Limite superior >>>";
```

Programa

Como programa de repaso de las sentencias «GO TO» e «IF ... THEN ...», se propone el programa número «2». Sálvelo, por ejemplo, de la forma:

SAVE "AREAS" LINE 10

Este programa calcula el área y el *perímetro* de las siguientes figuras geométricas.

- TRIANGULO.
- RECTANGULO.
- CUADRADO.
- ROMBO.
- ROMBOIDE.
- TRAPECIO.
- CIRCULO.

El programa, al autoejecutarse, presenta en pantalla un menú con las diversas opciones; seleccionando una de ellas, pasaremos a una pantalla en la que aparecerá dibujada la figura geométrica correspondiente, ésta es realizada con ayuda de

las sentencias «PLOT», «DRAW» o «CIRCLE». Una vez introducidos los datos (lado, altura,...) que nos pide el ordenador, los resultados correspondientes al «área» y al «perímetro» serán visualizados.

Para retornar al menú principal debe pulsar «S» o «S», pulsando «N» o «N» el programa se detendrá presentando el mensaje:

9 STOP statement, 90 20 2

La estructura del programa es:

- 10: Comentario con el nombre del programa.
- 20: Asignación de los colores de la pantalla.
- 30-130: Menú de opciones.
- 140: Entrada de «opción».
- 150-240: Verificación y selección.
- 250-314: Dibujo de triángulo.
- 320-360: Entrada de datos, base y altura).
- 370-430: Cálculo y visualización.

- 440: Salto de la rutina «FIN».
- 450-540: Dibujo del rectángulo.
- 550-590: Entrada de datos, (base y altura).
- 600-640: Cálculo y visualización.
- 650: Salto a la rutina «FIN».
- 700-780: Dibujo del cuadrado.
- 790-900: Entrada de datos, (lado).
- 910-950: Cálculo y visualización.
- 960: Salto a la rutina «FIN».
- 970-1040: Dibujo del rombo.
- 1050-1090: Entrada de datos, (diagonal mayor y diagonal menor).
- 1100-1150: Cálculo y visualización.
- 1160: Salto a la rutina «FIN».
- 1170-1265: Dibujo del romboide.
- 1270-1314: Entrada de datos, (base, altura y lado).
- 1320-1360: Cálculo y visualización.
- 1370: Salto a la rutina «FIN».
- 1380-1480: Dibujo del trapecio.
- 1490-1550: Entrada de datos, (base mayor, base menor y altura).
- 1560-1610: Cálculo y visualización.
- 1620: Salto a la rutina «FIN».
- 1630-1670: Dibujo del círculo.
- 1680-1700: Entrada de datos, (radio).
- 1710-1750: Cálculo y visualización.
- 1760: Salto a la rutina «FIN».
- 9000-9040: Rutina «FIN».

BUCLAS

Al analizar un programa, sucede con bastante frecuencia, que deba repetirse un cálculo o realizar una misma tarea con distintos datos. Sería una manera ilógica, en principio, editar las instrucciones de cálculo tantas veces como datos tengamos, ya que se utilizaría más memoria del ordenador; una forma algo más lógica sería utilizar un *bucle* (loop en inglés), que repitiera las mismas instrucciones tantas veces como quisieramos.

En programación, la terminología inglesa de los tipos de bucle más utilizada es:

- DO WHILE
- REPEAT UNTIL
- FOR ... NEXT

DO WHILE

La estructura de bucle DO WHILE (Hacer mientras...) permite realizar una tarea varias veces, siempre y cuando se cumpla la condición impuesta.

Ejemplo:

```
10 REM *****
11 DO WHILE *****
12 LET total=0
13 LET numero=1
14 WHILE numero>0
15 PRINT "Introduce un número"
16 INPUT numero
17 IF numero<0 THEN
18 PRINT "Número no válido"
19 GOTO 14
20 LET total=total+numero
21 LET numero=numero+1
22 WEND
23 PRINT "Total: ",total
24 GOTO 20
```

mientras se cumpla que la variable «numero» sea igual o inferior a 10000, se realizarán indefinidamente las tareas de cálculo y visualización de la variable «total».

tico tiene esto? Que con una estructura DO WHILE, si al entrar en el bucle no se cumple la condición prevista, se sale de él sin haber ejecutado ninguna tarea; sin embargo, con la estructura REPEAT UNTIL, al menos una vez se ejecutan las instrucciones contenidas en él.

FOR/NEXT

Acceso al teclado

SGN

F TO FOR MODO K

IN KEY \$

N NEXT MODO K

OVER

Tipo de sentencia

Comando de programación.

Definición

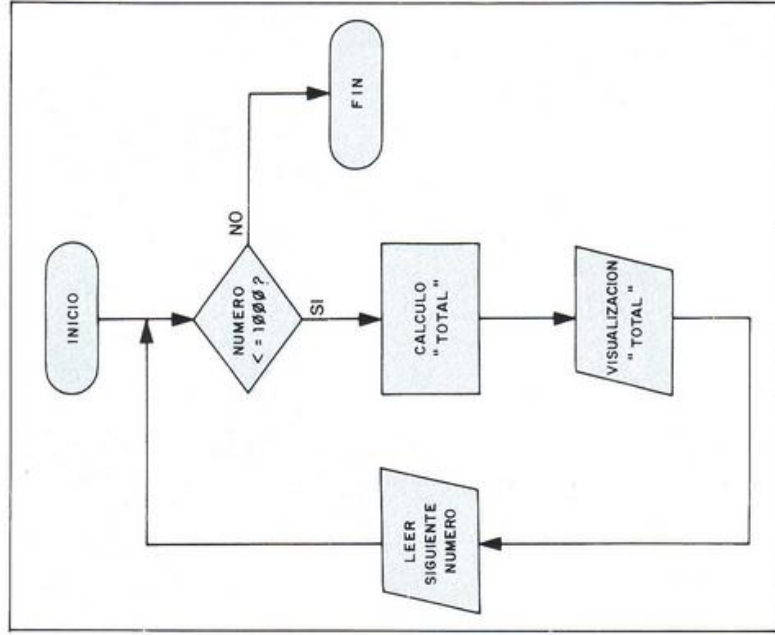
Los bucles «FOR/NEXT» permiten realizar las instrucciones contenidas en él tantas veces como se indica en los límites del argumento. El comienzo del bucle está definido por la sentencia «FOR», y el final por «NEXT». La estructura de la sentencia «FOR» es:

SENTENCIA	ARGUMENTO
FOR	var.= exp:TO exp.

var. = variable.

exp. = expresión.

dentro del argumento se utiliza.



Estructura «DO WHILE».

REPEAT UNTIL

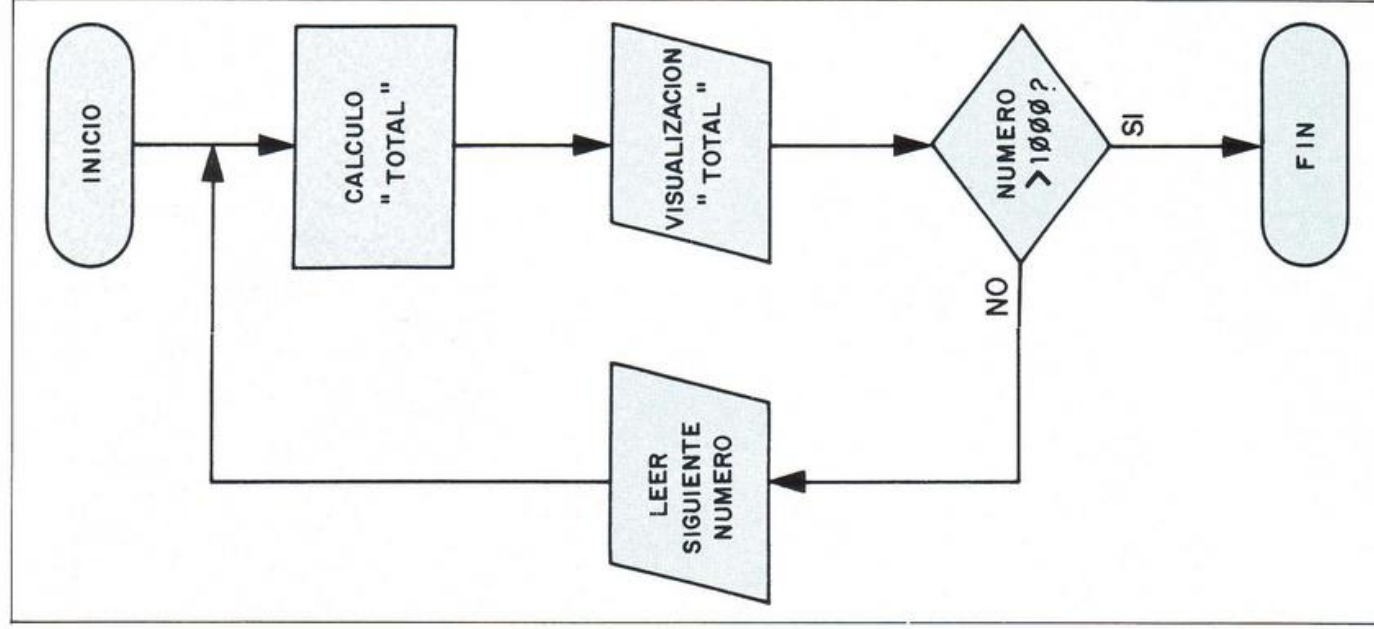
Con este tipo de bucle, (Repetir hasta..., es la traducción) el ordenador realiza una y otra vez la tarea hasta que se da una condición.

Ejemplo:

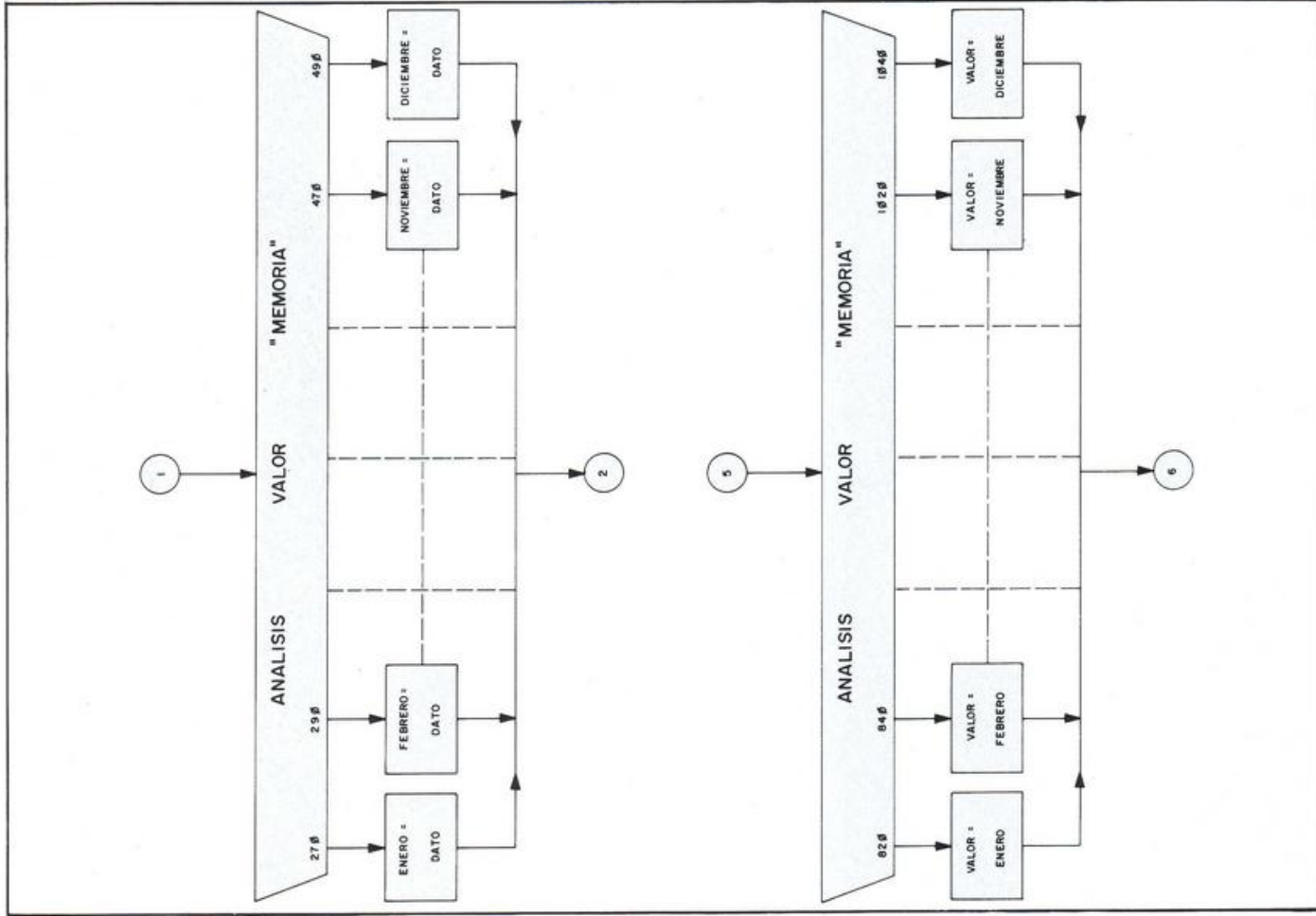
```
10 REM *****
11 DO UNTIL *****
12 LET total=0
13 LET numero=1
14 WHILE numero>0
15 PRINT "Introduce un número"
16 INPUT numero
17 IF numero<0 THEN
18 PRINT "Número no válido"
19 GOTO 14
20 LET total=total+numero
21 LET numero=numero+1
22 WEND
23 PRINT "Total: ",total
24 GOTO 20
```

Diferencias

La diferencia entre estos dos tipos de estructura estriba, en que en el primer caso (DO WHILE) la salida del bucle se encuentra antes de realizar la tarea, y en el segundo (REPEAT UNTIL) se encuentra al final, ¿Qué significado prác-



Estructura «REPEAT UNTIL».



Programa «HISTOGRAMA» almacenamiento y extracción de datos.

control, la variable «a» se incrementa en 1 cada vez que «b» completa su ciclo.

El siguiente programa calcula las potencias segunda, tercera, cuarta y quinta correspondientes a los veinte primeros números:

```

10 REM *****
  : POTENCIACION *****
  : *****
20 FOR J=1 TO 20
30 LET S=J^2
40 LET S=S^2
50 PRINT S;J^3;J^4;J^5
60 NEXT J

```

en la primera columna, aparecen los números del 1 al 20 y en las siguientes, por orden, la correspondiente potenciación.

Errores

Hay una serie de mensajes de error relacionados con los bucles «FOR ... NEXT ...»:

- Sentencia «NEXT» sin «FOR».
- El mensaje:

```

1 NEXT without FOR

```

aparece cuando el ordenador encuentra una sentencia «NEXT» sin haber ejecutado con anterioridad su sentencia «FOR» correspondiente y, además, existe definida una variable con el mismo nombre que el argumento de «NEXT».

Ejemplo:

```

10 REM *****
  : ERROR 1 *****
  : *****
20 LET J=30
30 LET S=J^2
40 NEXT J

```

- Variable no encontrada.

```

2 Variable not found

```

similar al error anterior, pero sin estar definida ninguna variable con el mismo nombre.

Ejemplo:

```

10 REM *****
  : ERROR 2 *****
  : *****
30 LET S=30
40 PRINT S;J^2
50 NEXT J

```

- Sentencia «FOR» sin «NEXT».

```

1 FOR with out NEXT

```

Este mensaje lo presenta el ordenador cuando se encuentra con una sentencia «FOR» en la que los límites o el paso están incorrectos y además no encuentra la sentencia «NEXT» correspondiente.

Ejemplo:

```

10 REM *****
  : ERROR 3 *****
  : *****
20 FOR J=4 TO 2
30 PRINT J;J

```

- STEP ϕ .
- Cuando por error se edita una sentencia «FOR» con paso ϕ , la variable de control no se incrementa al ejecutarse la sentencia «NEXT» correspondiente.

Ejemplo:

```

10 REM *****
  : ERROR 4 *****
  : *****
20 FOR J=30
30 LET S=J^2
40 NEXT S;J

```

```

20 FOR J=10 TO 200 STEP 0
30 LET S=J^2
40 PRINT S;J
50 NEXT J

```

a pesar de estar comprendidos los límites entre 10 y 200, la variable «j» asume el valor «10» una y otra vez y, por tanto, no alcanza el valor final.

- Cuando por error, se omite la palabra clave «STEP» en un bucle decreciente, éste no se ejecuta y, por lo tanto, continúa en la instrucción siguiente al «NEXT». El ordenador no presenta en este caso mensaje de error.
- Ejemplo:

```

10 REM *****
  : ERROR 5 *****
  : *****
20 FOR I=200 TO 10
30 LET C=I^2
40 PRINT C;I
50 NEXT I
60 PRINT ***** FIN *****

```

Programas

Los dos programas que se muestran a continuación, son aplicaciones de bucles «FOR ... NEXT ...».

El programa número «1» es de utilidad en matemáticas. Según se introducen los datos, se calcula su suma aritmética, su media aritmética y la suma de sus cuadrados.

La estructura es la siguiente:

- 1 ϕ : Comentario con el nombre del programa.
- 2 ϕ : Asignación de los colores de pantalla.
- 3 ϕ -5 ϕ : Entrada del número de datos a calcular,

[illegible]

60-68	: Definición de variables.
72-74	: Visualización del nombre del programa.
80-160	: Bucle central para la entrada de datos, salto a la rutina de «cálculo» y visualización de los resultados. El dato tiene que estar comprendido entre los valores « ϕ_0 » y « $100\phi_0$ ».
170-190	: Fin de los cálculos.
200-230	: Rutina de cálculo.
250-270	: Bucle para el borrado de los resultados anteriores.
280	: Salto al bucle central.

El programa «2» dibuja una gráfica de barras o «histograma» con los datos de ventas mensuales.

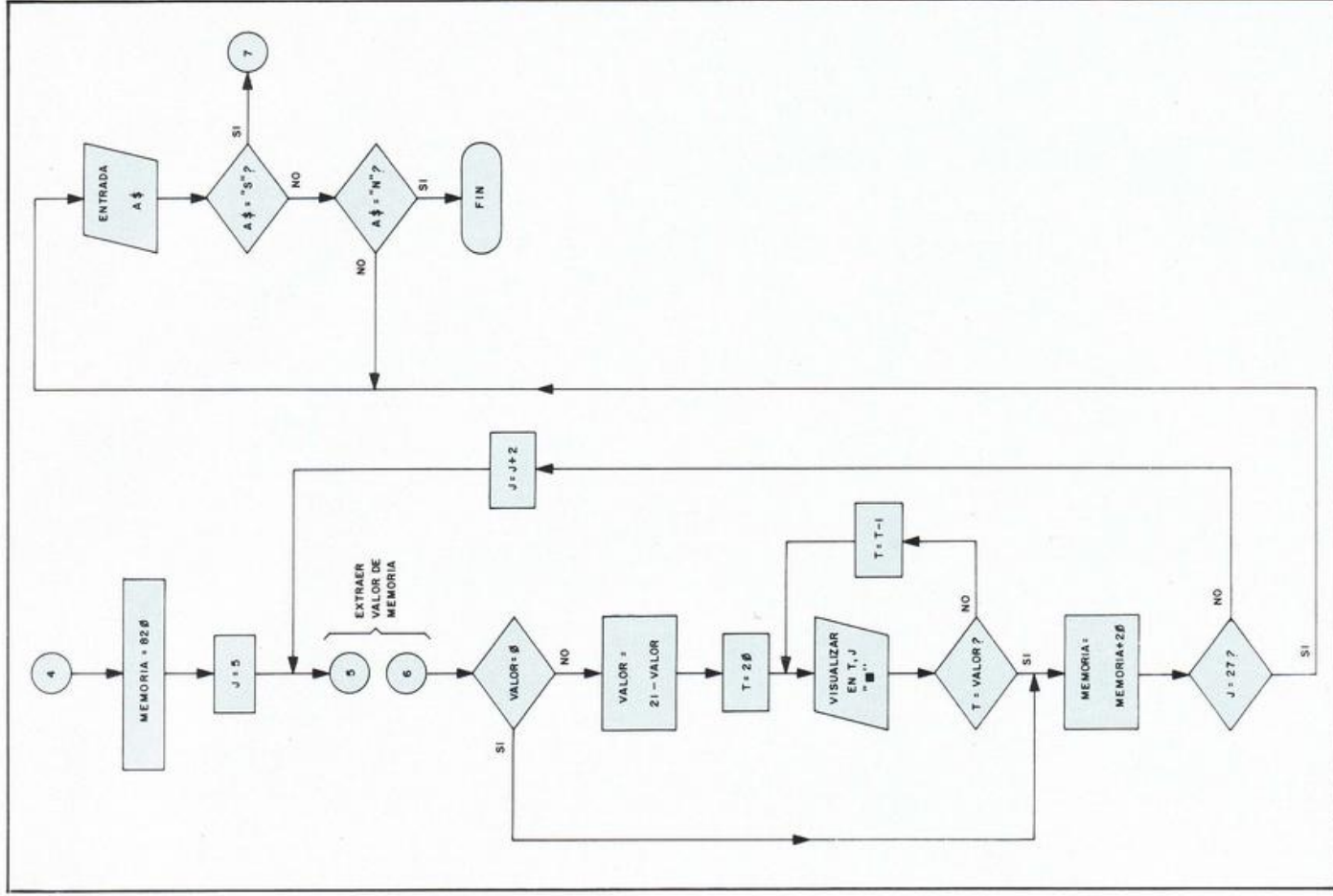
10 : Comentario con el nombre del programa.

20 : Asignación del color verde para fondo y borde, y negro para tinta.

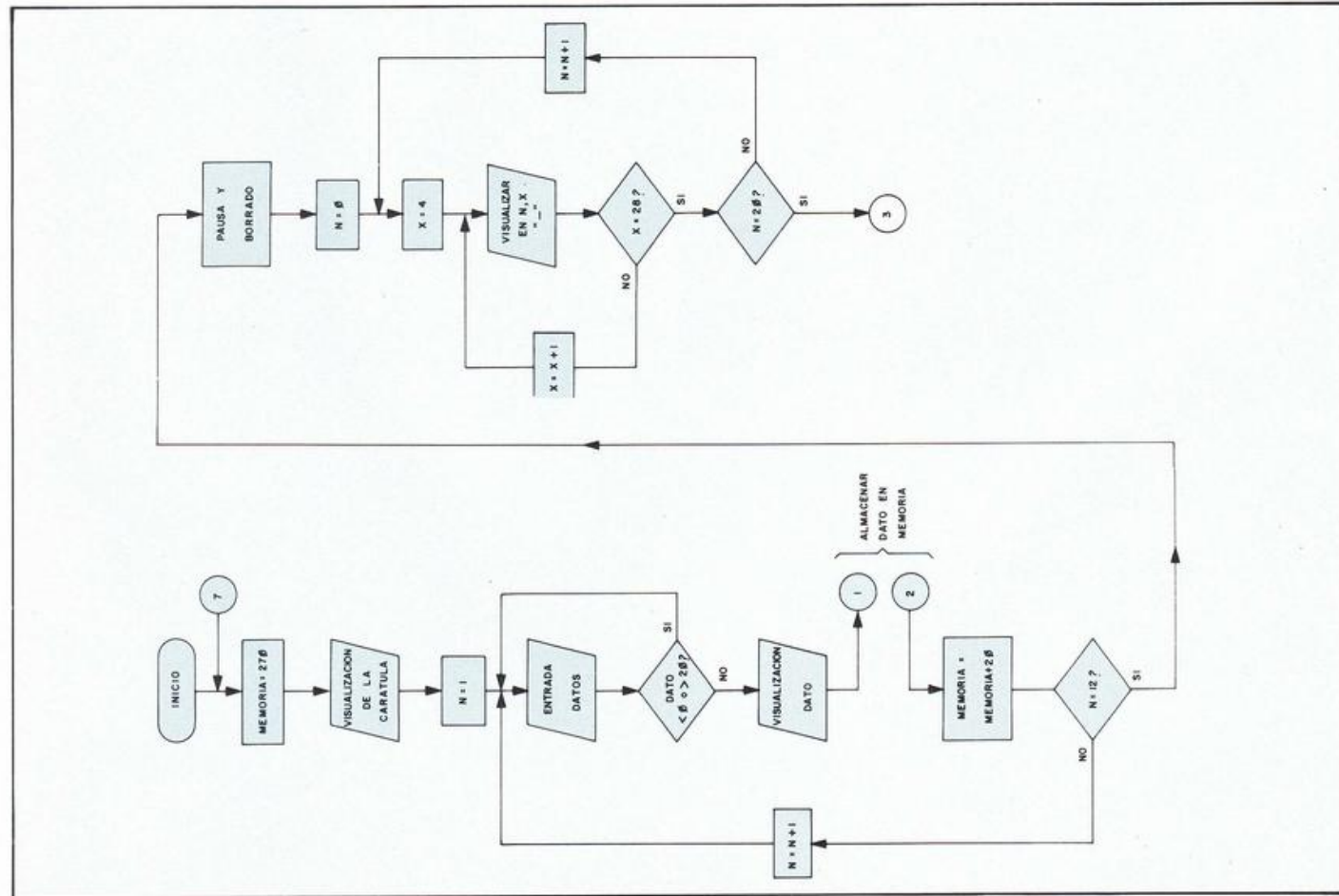
22 : Definición de la variable, que asume el valor de la línea donde se comienza la rutina de almacenamiento de ventas.

3φ-17φ : Presentación de la carátula donde se visualizarán posteriormente los datos de ventas.

180-250 : Bucle para la entrada de los datos mensuales.



Programa «HISTOGRAMA» dibujo, barras y rutina «FIN».



Programa «HISTOGRAMA» entrada de datos y dibujo líneas horizontales.

```

*****
* CALCULO *
*****
210 LET suma=suma+dato
220 LET cuadrados=cuadrados+dato
230 LET media=suma/n
240 REM
*****
* BORRADO *
*****
250 FOR a=7 TO 13 STEP 2
260 PRINT AT a,19; "
270 NEXT a
280 GO TO 120

```

verificación de los mismos, tienen que estar comprendidos entre «0» y «20» (millones), visualización y salto a la rutina de almacenamiento.

252-256 : Pausa y borrado de la pantalla.
 260 : Salto a la rutina de dibujo.
 262-500 : Almacenamiento de datos y retorno al bucle principal.
 510-560 : Bucle anidado

para dibujar las rayas horizontales del «histograma» con el símbolo del subrayado «_».

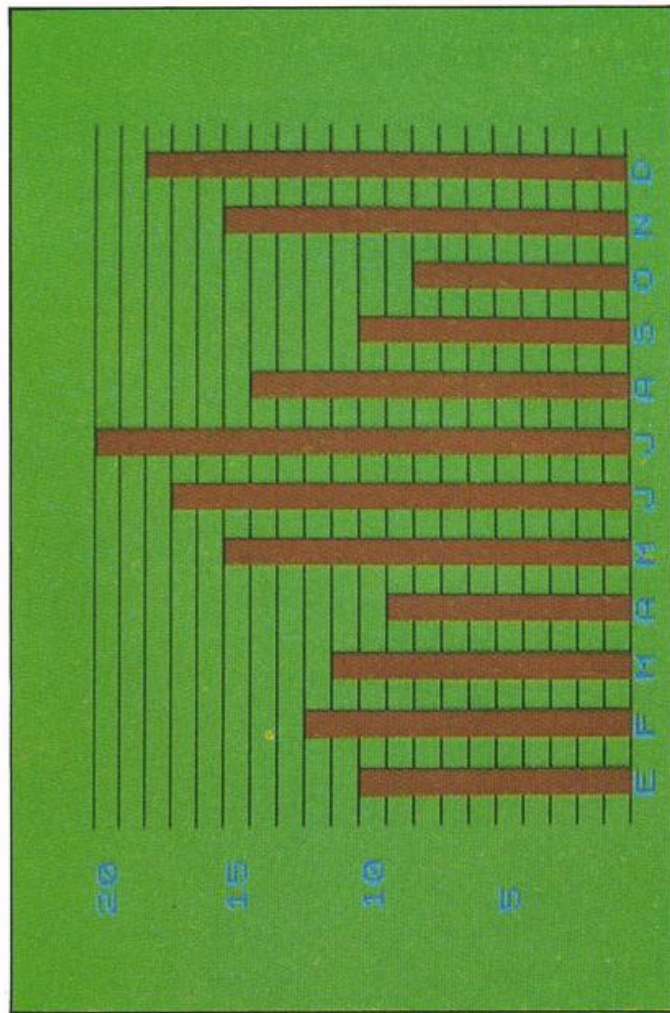
570-620 : Visualización de los valores en millones en el eje vertical (tinta azul).

630-680 : Visualización de las iniciales de los meses en el eje horizontal (tinta azul).

690-790 : Bucle para el salto a la rutina de extracción de datos de memoria y su representación en gráfico de barras (tinta roja).

800 : Salto de la rutina «fin».

810-1050 : Rutina de extracción de datos y

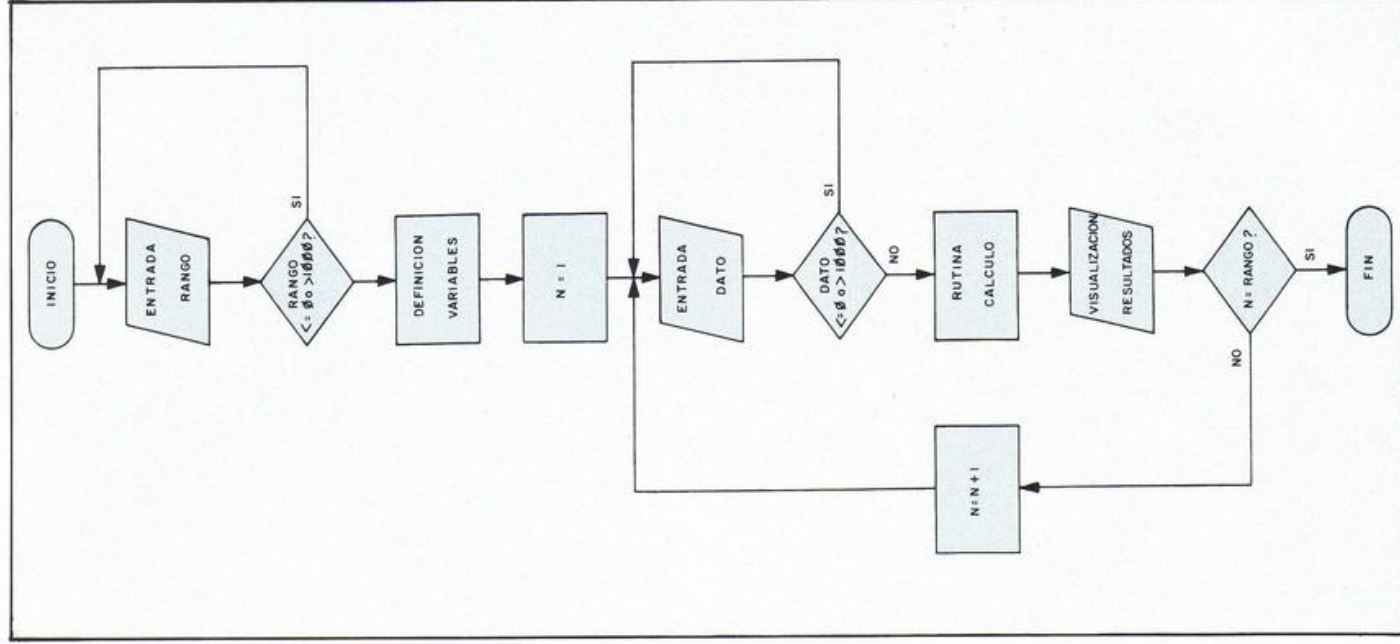


PROGRAMA 2

```

10 REM *****
11 HISTOGRAMAS *
12 *****
13 BORDER 4: PAPER 4: INK 0: C
14 LET memoria=270
15 REM *****
16 *****
17 CARATULA *
18 *****
19 PRINT AT 0,7;"Entrada de da
20 tos"
21 PRINT AT 1,7;"
22 PRINT AT 6,0;"Enero .....
23
24 PRINT "Febrero .....
25 PRINT "Marzo .....
26 PRINT "Abril .....
27 PRINT "Mayo .....
28 PRINT "Junio .....
29 PRINT "Julio .....
30 PRINT "Agosto .....
31 PRINT "Septiembre .....
32 PRINT "Octubre .....
33 PRINT "Noviembre .....
34 PRINT "Diciembre ....
35 REM *****
36 *****
37 ENTRADA DATOS *
38 *****
39 FOR n=1 TO 12
40 INPUT "Datos mes ";(n);
41 IF dato<0 OR dato>20 THEN G
42 O TO 200
43 PRINT AT 5+n,15;dato;AT 5+n
44 " millones"
45 GO TO memoria
46 LET memoria=memoria+20
47 PRINT n0;"Pulse una tecla p
48 a CONTINUA"
49 PAUSE 0
50 CLS
51 GO TO 510
52 REM *****
53 *****
54 MEMORIA *
55 *****
56 LET enero=dato
57 LET febrero=dato
58 LET marzo=dato
59 LET abril=dato
60 LET mayo=dato
61 LET junio=dato
62 LET julio=dato
63 LET agosto=dato
64 LET septiembre=dato
65 LET octubre=dato
66 LET noviembre=dato
67 LET diciembre=dato
68 REM *****
69 *****
70 IF a$="s" OR a$="S" THEN GO
71 TO 10
72 IF a$="n" OR a$="N" THEN ST
73 GO TO 1070
74 GO TO 1070

```



Programa «ESTADISTICA».

retorno al bucle principal.

1060-1100 : Decisión. de retorno al principio o terminar el programa.

La rutina de almacenamiento y extracción de datos, se ha realizado de una manera un tanto especial, ya que todavía no se han visto las matrices, que serán estudiadas en un capítulo posterior. ■

el valor "99", para poder ser actualizado en la primera jugada.

También se definen, para una mejor interpretación, las variables utilizadas como dirección de comienzo de las subrutinas.

102-104 : Inicialización de las variables utilizadas en cada partida.

106-1505 : Presentación del programa y visualización de instrucciones, si se desea.

160-190 : Mensaje de invitación a comenzar el juego.

200-240 : Visualización del número calculado por el ordenador.

250 : Si el ordenador está seguro del número, se ejecuta la subrutina "aciertito".

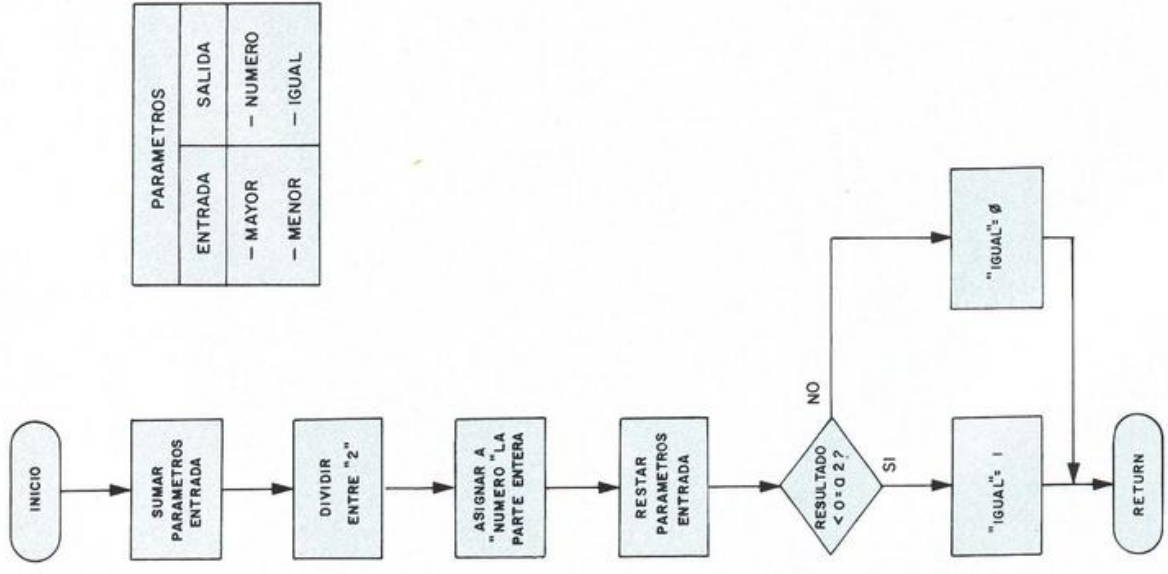
255-290 : Introducción de la clave correspondiente a la pista (M, N o I), si no es "igual", se salta a la línea 200 para calcular un nuevo número.

300 : El número ha sido acertado.

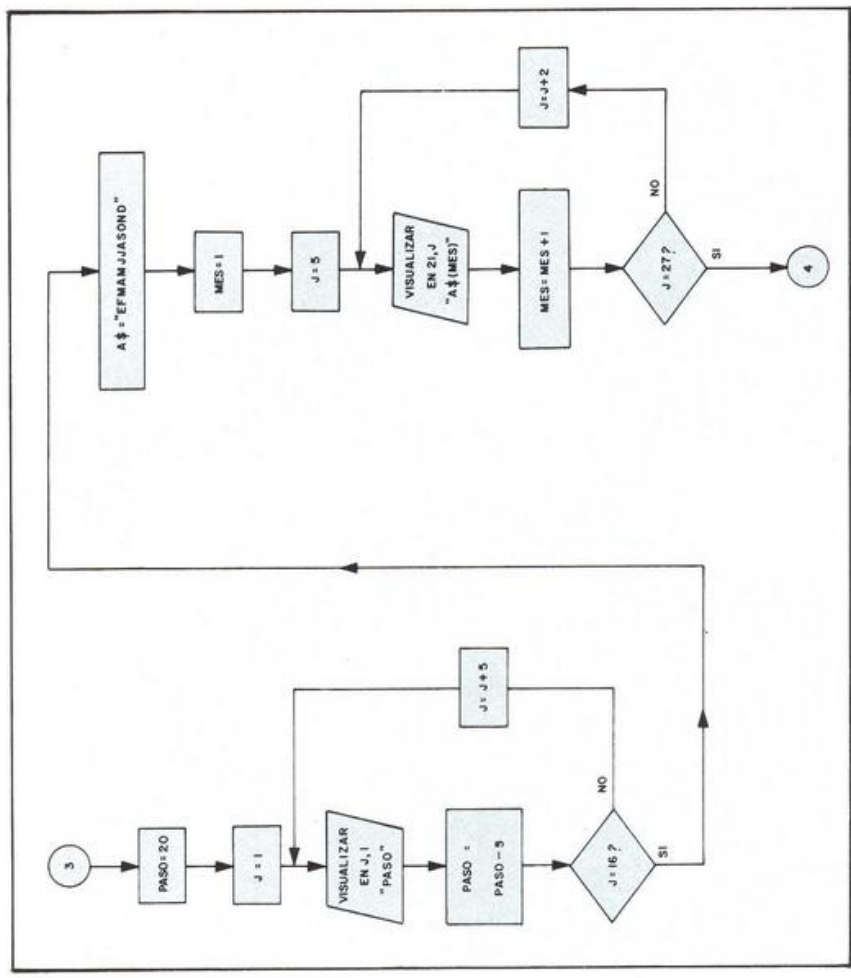
310 : ¿Se desea continuar?

320-330 : En caso afirmativo, se borra la pantalla y se comienza de nuevo en la línea "100".

1000-1070 : Subrutina "INS-TRUCCION". Se visualizan las instrucciones durante un tiempo y luego se borra la pantalla. Se llama a la



Programa «Adivino» subrutina «Cálculo».



Programa «Histograma» datos eje vertical y horizontal.

SUBROUTINAS

Normalmente, en un programa hay ciertos cálculos o funciones que se necesitan en distintas partes del mismo; en lugar de editarlas varias veces, conviene hacerlo una sola vez en formato *subrutina* o *subprograma*, de esta manera, este grupo de instrucciones sólo se ejecutan cuando el programa principal lo indica, mediante una instrucción de llamada a subrutina; cuando ésta termina de ejecutarse devuelve el control al programa principal.

Un bucle también repite varias veces una serie de instrucciones, pero siempre en una misma zona de programa; a diferencia, la subrutina puede ser llamada desde cualquier parte.

Dentro de la programación *estructurada*, la utilización de subrutinas es un hábito muy recomendable. Conviene para una mayor estructuración que todas las subrutinas estén localizadas en la zona final del programa; una a continuación de la otra.

GO SUB

Acceso al teclado

SQR



MODO K

Definición

Esta palabra clave se utiliza para retornar, de una subrutina, al programa principal.

Tipo de sentencia

Comando de programación.

Definición

Es la instrucción utilizada, dentro de un programa, para «llamar» a una subrutina, aunque también admite el formato de comando directo.

Su estructura general es:

SENTENCIA	ARGUMENTO
GO SUB	N.º de línea

Ejemplos:

— GO SUB 500.
— GO SUB 30.

RETURN

Acceso al teclado

STR \$



MODO K

Tipo de sentencia

Comando de programación.

Definición

Esta palabra clave se utiliza para retornar, de una subrutina, al programa principal.

Su estructura es la siguiente:

SENTENCIA	ARGUMENTO
RETURN	—

Utilización de «GO SUB» y «RETURN»

La llamada a subrutina «GO SUB» puede hacerse en cualquier parte del programa que se necesita. La palabra clave «RETURN», sin embargo, debe utilizarse *siempre* al final de cada subrutina.

Ejemplo:

```
10 REM *****
20 GO SUB *****
30 INPUT "¿A?"; A
40 GO SUB 100
50 PRINT "B"
60 STOP
100 REM *****
110 IF A=0 THEN LET "B"="IGUAL"
120 GO TO 30
130 RETURN *****
140 RETURN *****
```

Cuando se ejecuta la línea «40» el programa principal cede el control a la subrutina localizada en la línea «100», ésta se va ejecutando hasta encontrar la sentencia «RETURN», que causa el retorno al programa principal (línea 50).

A primera vista, parecen similares las sentencias «GO TO» y «GO SUB», ya que ambas provocan un salto al número de línea especificado en

```
10 LET SUBROUTINA=100
20 INPUT "Sumando 2"; A
30 INPUT "Sumando 3"; B
40 GO SUB SUBROUTINA
50 *****
60 PRINT "A+B+C="; A+B+C
70 *****
80 SUBROUTINA:
90 *****
100 PRINT "A+B="; A+B
110 PRINT "A+B+C="; A+B+C
120 RETURN
```

Programas

Como aplicación a las «subrutinas», se presentan dos programas:

— Adivino.

— Longitud.

El primero se trata de un juego de *adivinanza*, en el que el ordenador debe descubrir el número pensado por usted, que deberá estar comprendido entre «0» y «100».

Al principio del programa, se da la opción de poder visualizar las instrucciones del juego. El ordenador, posteriormente, presenta un número de pantalla; según la relación que exista entre éste y el número pensado por nosotros, deberemos introducir una de las siguientes claves:

«M» - Mayor.

«N» - Menor.

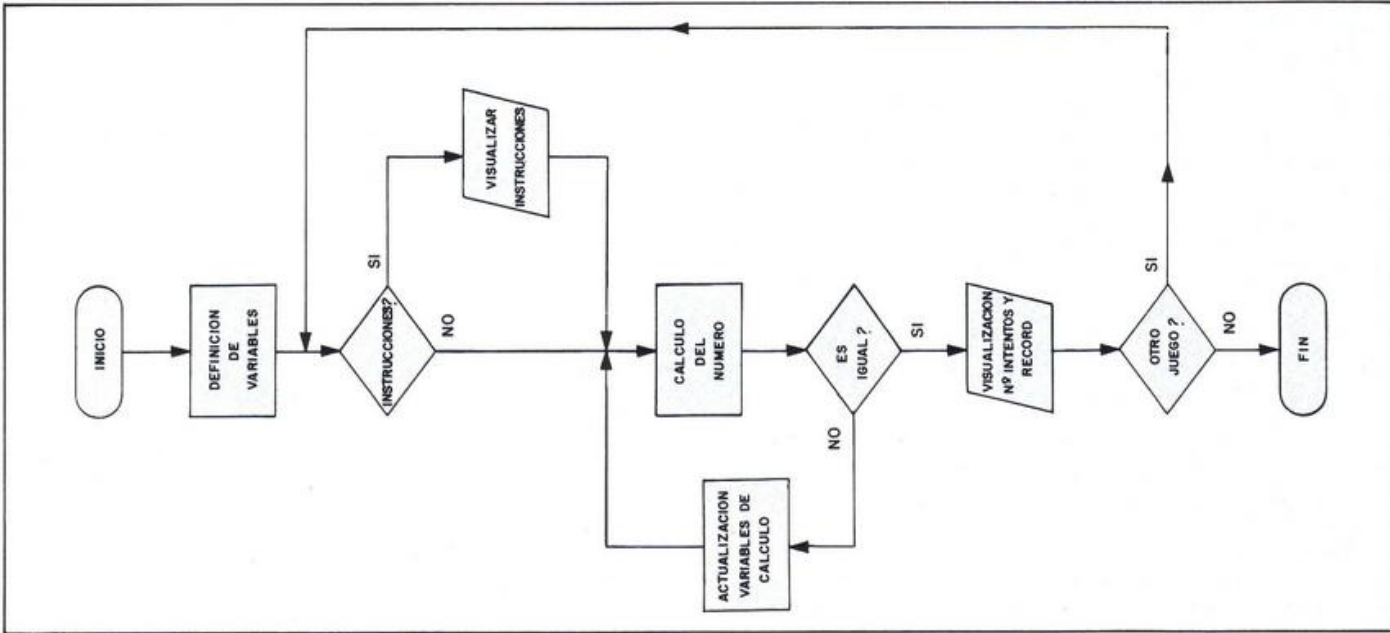
«I» - Igual.

La estructura general del programa es:

10 : Comentario con el nombre del programa.

12 : Asignación del color verde para el borde y el fondo, y azul para los caracteres.

14-80 : Definición de la variable «record», utilizada para almacenar el mínimo número de intentos; inicialmente tiene asignado



Estructura programa «Adivino».

PROGRAMA 1

```

10 REM
** CURSOR/BASIC **
** Adivino **
** **
12 BORDER 4: PAPER 4: INK 1: C
14 REM
** VARIABLES **
** **
15 LET record=99
20 LET instruccion=1000
30 LET error=1100
40 LET temporizacion=1200
50 LET calculo=1300
60 LET verificacion=1400
70 LET acierto=1500
80 LET seguir=1600
100 REM
** BUCLE CENTRAL **
** **
102 LET intento=1
104 LET mayor=101: LET menor=0
106 PRINT AT 4,7;"PROGRAMA ""AD
110 INPUT "Desee conocer las in
111 strucciones (S/N) >>>"; LINE a$
112 IF a$="S" OR a$="s" THEN GO
113 SUB instruccion: GO TO 160
114 IF a$="N" OR a$="n" THEN GO
115 TO 160
116 GO SUB error
117 GO TO 110
118 PRINT AT 9,5;"piense un num
119 ero entre 1 y 100"; AT 12,10;"0" y "1
120 GO SUB temporizacion
121 GO TO 116
122 LET calculo=0
123 PRINT AT 9,3;"Su numero es
124 el...";
125 LET calculo=calculo+1
126 GO SUB calculo
127 IF calculo=1 THEN GO SUB acie
128 TO 310
129 PRINT "Mayor/Menor/I=Ig
130 uo";
131 GO SUB verificacion
132 IF reconocimiento=0 THEN GO
133 TO 260
134 IF reconocimiento=1 THEN PR
135 INT AT 9,25;"LET intento=
136 intento+1: GO TO 200
137 IF reconocimiento=2 THEN GO
138 SUB acierto
139 GO SUB seguir
140 CLS
141 GO TO 100
142 REM
999

```

Error

Es necesario separar las subrutinas del resto del programa mediante sentencias del tipo:

- STOP.
- GO TO n.

ya que de lo contrario, podrían ejecutarse sin haber sido llamadas, provocando un error del tipo:

7 RETURN without GO SUB

Ejemplo:

```

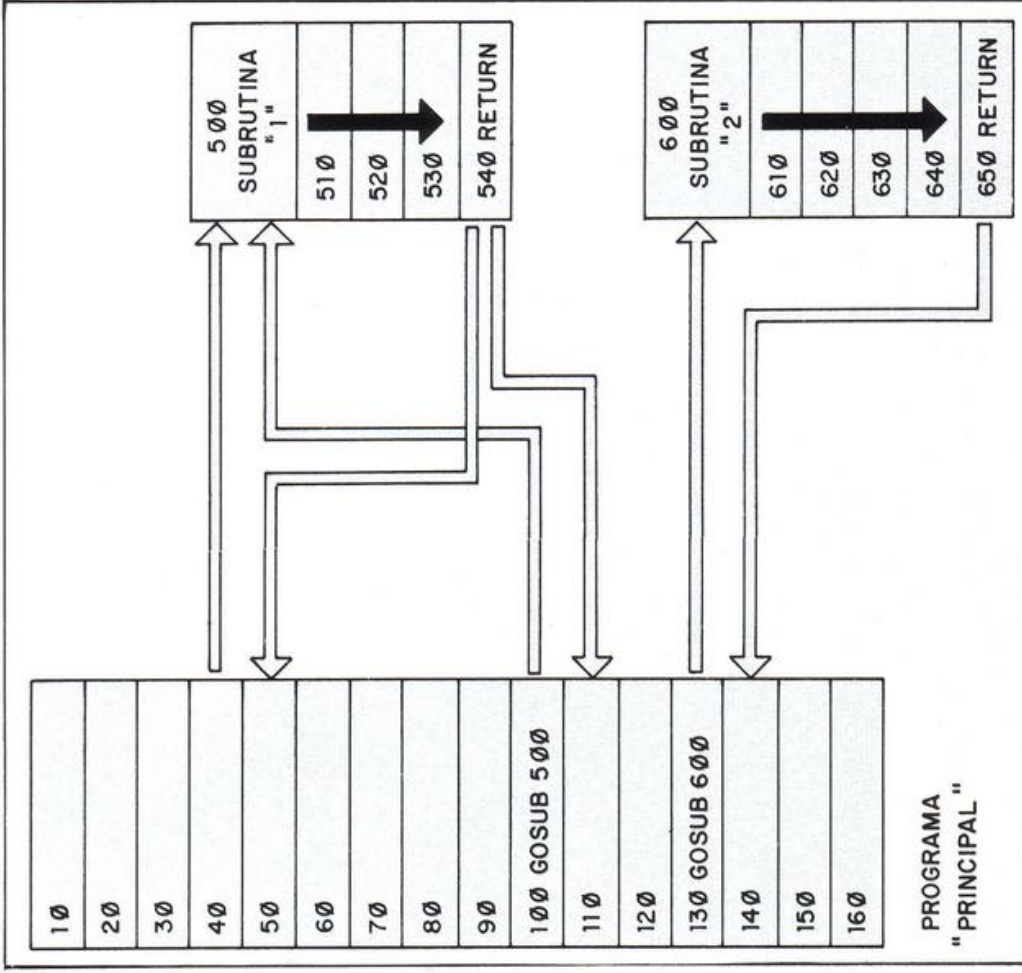
10 REM *****
11 ERROR
12 *****

```

```

1000 REM INSTRUCCION
1010 PRINT AT 8,3;"Debe pensar u
1020 n numero entre 0 y 100. El orde
1030 nador tratara de adivinarlo en el
1040 menor numero de intentos. Para f
1050 acilitarle la tarea, debe darle
1060 alguna pista."
1070 PRINT AT 16,8;"M si es mayo
1080 1090 PRINT AT 16,8;"N si es meno
1090 1100 PRINT AT 20,8;"I si es igua
1110 1120 LET retardo=1200
1130 GO SUB temporizacion
1140 CLS
1150 REM ERROR
1160 RETURN
1170 REM RESPUESTA
1180 PRINT AT 1,4;"Respuesta
1190 no reconocida."
1200 LET retardo=200
1210 GO SUB temporizacion
1220 REM RECONOCIMIENTO
1230 FOR X=1 TO retardo
1240 NEXT X
1250 RETURN
1260 REM CALCULO
1270 LET numero=INT ((menor+mayo
1280 (1/2))
1290 IF mayor-menor<=2 THEN LET
1300 igual=1: RETURN
1310 LET igual=0
1320 REM VERIFICACION
1330 RETURN
1340 IF a$="N" OR a$="n" THEN LE
1350 t menor=numero: GO TO 1450
1360 IF a$="S" OR a$="s" THEN LE
1370 t mayor=numero: GO TO 1450
1380 IF a$="I" OR a$="i" THEN LE
1390 t calculo=calculo+1: RETURN
1400 REM RECONOCIMIENTO=0
1410 LET reconocimiento=0
1420 RETURN
1430 REM RECONOCIMIENTO=1
1440 LET reconocimiento=1
1450 RETURN
1460 REM ACIERTO
1470 PRINT AT 12,8;"!!! ACERTE !
1480
1490 PRINT AT 15,8;"En ""intento
1500 intentos""
1510 LET retardo=400
1520 GO SUB temporizacion
1530 CLS
1540 IF intento<record THEN LET
1550 record=intento
1560 PRINT AT 9,7;"Mi record est
1570 a en:"
1580 PRINT AT 13,14;".....record:"
1590 RETURN
1600 REM SEGUIR
1610 INPUT "Quiere jugar otra ve
1620 z?"; LINE a$
1630 IF a$="S" OR a$="s" THEN RE
1640 TURN
1650 IF a$="N" OR a$="n" THEN ST
1660 OP
1670 GO SUB error
1680 GO TO 1610

```



Llamadas a «subrutinas».

su argumento, la diferencia ra- dica en que, el ordenador, cuando se ejecuta la senten- cia «GO SUB», apunta en una zona de la memoria denomina- da STACK o «Pila de GO SUB», memoria tipo «LIFO» (Last In- put First Output) en la que el último dato almacenado es el primero en ser recuperado, la dirección de retorno, formada por el número de línea y sen- tencia dentro de la línea de la

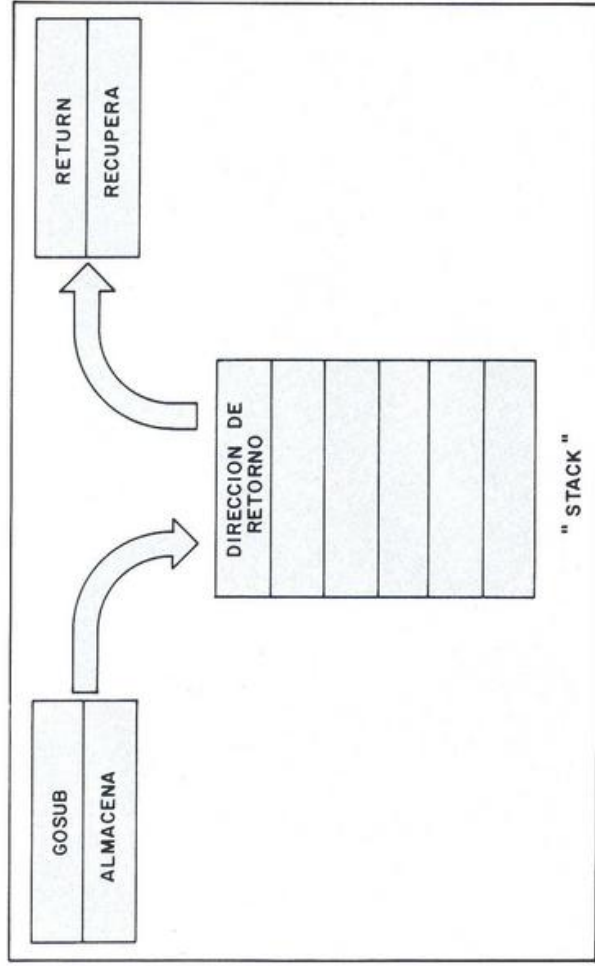
Las llamadas a subrutina pueden hacerse con variables numéricas, cuyo valor sea el número de línea donde están localizadas.

Ejemplo:

```

10 REM *****
11 VARIABLES
12 *****
30 LET a=100
30 LET b=100
30 LET c=100

```

Pila de «Gosub».

```

40 PRINT "SUBROUTINA 1"
50 GOSUB ENTRADA
60 GOSUB SUBROUTINA 2
70 STOP
80 REM
90 INPUT "Nombre >>> ", LINE A
100 PRINT "Apellidos >>> ", LIN
110 RETURN
120 PRINT "Tu nombre es: ", A,
130 RETURN

```

Ejemplo:

```

10 REM *****
20 SIN PARAMETROS *****
30 GOSUB 100 *****
40 PRINT AT 10,10,"MICROHOBBY" *****
50 STOP *****
60 REM *****
70 LET CLS *****
80 PRINT AT 10,10,"MICROHOBBY" *****
90 PRINT AT 10,11,"SEÑAL" *****
100 REM *****
110 LET CLS *****
120 PRINT AT 10,21,"color,AT 4,0," *****
130 NEXT A *****
140 NEXT A *****
150 FOR I=1 TO 150 *****
160 RETURN *****

```

Las subrutinas pueden clasificarse según la filosofía de su ejecución en:

- Subrutinas sin parámetros.
- Subrutinas con parámetros.

Una subrutina sin parámetros es aquella que realiza un cálculo o función siempre de la misma manera, es decir, con los mismos datos o valores.

El siguiente programa es una variable del anterior, permite borrar y colorear la pantalla del color que se desee.

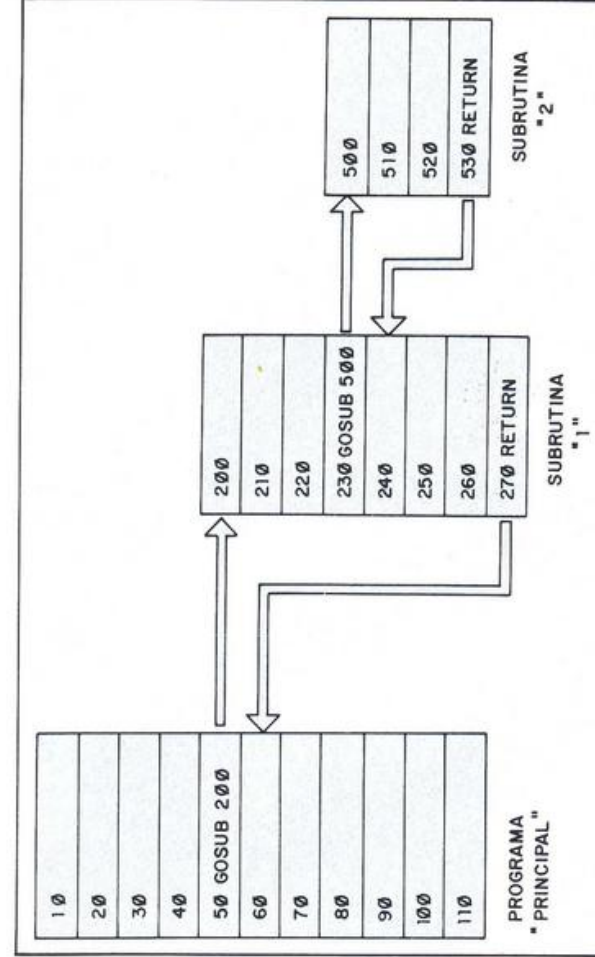
Antes de llamar a la subrutina debe asignarse, a la variable utilizada, el valor correspondiente al color.

```

10 REM *****
20 CON PARAMETROS *****
30 LET CLS *****
40 PRINT AT 10,10,"MICROHOBBY" *****
50 STOP *****
60 REM *****
70 LET CLS *****
80 PRINT AT 10,10,"MICROHOBBY" *****
90 PRINT AT 10,11,"SEÑAL" *****
100 REM *****
110 LET CLS *****
120 PRINT AT 10,21,"color,AT 4,0," *****
130 NEXT A *****
140 NEXT A *****
150 FOR I=1 TO 150 *****
160 RETURN *****

```

Se combina con otra subrutina localizada en la línea «200», que permite ver los mensajes sucesivos, ya que su



«Subrutinas» anidadas.

La única misión es la de temporizar.

Los parámetros pueden ser tanto de entrada como de salida, veamos un ejemplo en el que se combinan ambos.

```

10 REM *****
20 ENTRADA/SALIDA *****
30 INPUT "precio >>> ",ENTRADA *****
40 INPUT "impuesto de LUJO >>> *****
50 PRINT "precio *****
60 PRINT "impuesto de LUJO *****
70 PRINT "precio *****
80 PRINT "impuesto de LUJO *****
90 PRINT "precio *****
100 REM *****
110 LET CLS *****
120 PRINT AT 10,21,"color,AT 4,0," *****
130 NEXT A *****
140 NEXT A *****
150 FOR I=1 TO 150 *****
160 RETURN *****

```

Atendiendo al tipo de llamada, las subrutinas pueden clasificarse en:

- INCONDICIONALES.
- CONDICIONALES.

Incondicionales son aquellas que no necesitan de ninguna condición para que se ejecuten.

Ejemplo:

```

GO SUB 750

```

Una llamada a subrutina condicional, es aquella que necesita que se cumplan una o varias condiciones previas.

Ejemplo:

```

IF A$ = "PEREZ" THEN GO
SUB 100

```

Subrutinas anidadas

Al igual que los bucles, las subrutinas pueden anidarse, es decir, que desde una subrutina se puede llamar a otra y así sucesivamente. Las direcciones de retorno se van almacenando, como ya se comentó anteriormente, en el STACK;

la primera dirección en recuperarse corresponde con la de la última subrutina llamada.

Aunque no es muy frecuente, una subrutina puede llamarse a sí misma, esto es lo que en programación se denomina *subrutina Recursiva*.

Un ejemplo de subrutinas anidadas es el siguiente:

```

10 REM *****
20 ANIDADOS *****
30 LET FONDO=4 *****
40 GOSUB 100 *****
50 PRINT "Fondo: ",FONDO *****
60 RETURN *****
70 REM *****
80 LET FONDO=4 *****
90 GOSUB 100 *****
100 PRINT "Fondo: ",FONDO *****
110 RETURN *****
120 REM *****
130 FOR A=0 TO 15 *****
140 IF A=0 OR A=11 THEN GO SUB *****
150 NEXT A *****
160 PRINT "Fondo: ",FONDO *****
170 RETURN *****

```


Definición

Una de las ventajas de las tablas de datos es poder leer éstos cuantas veces desee-mos. Para volver al principio de la tabla o a cualquier parte de ella, se necesita restaurar el puntero o índice, esto se con-sigue con la sentencia «RES-TORE».

Su estructura es la siguiente:

SENTENCIA	ARGUMENTO
RESTORE	N.º de línea

Ejemplos:

- RESTORE 40.
- RESTORE 1230.
- RESTORE.

Cuando el argumento se omite, el intérprete BASIC to-ma por defecto el valor 0. Al ejecutar un programa con el comando directo «RUN» inicia-liza el puntero en la primera sentencia «DATA».

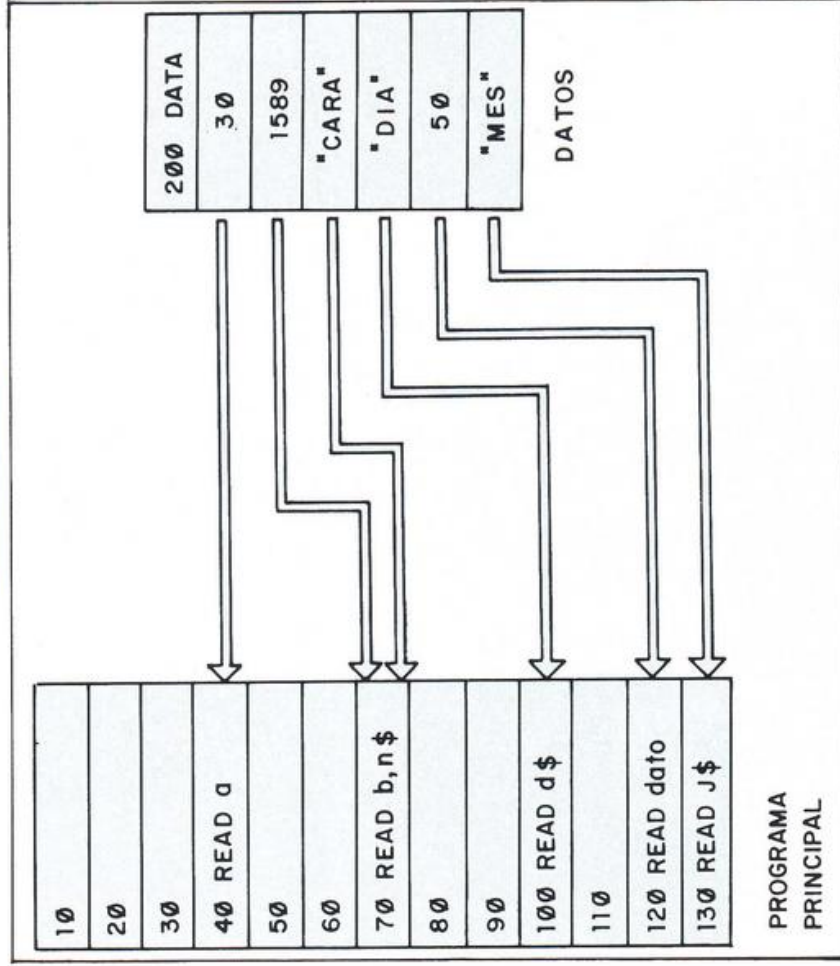
Si no se restaura el puntero al finalizar la lectura de la ta-bla, el siguiente dato que se quiera leer provocará un men-saje de error.

Ejemplo:

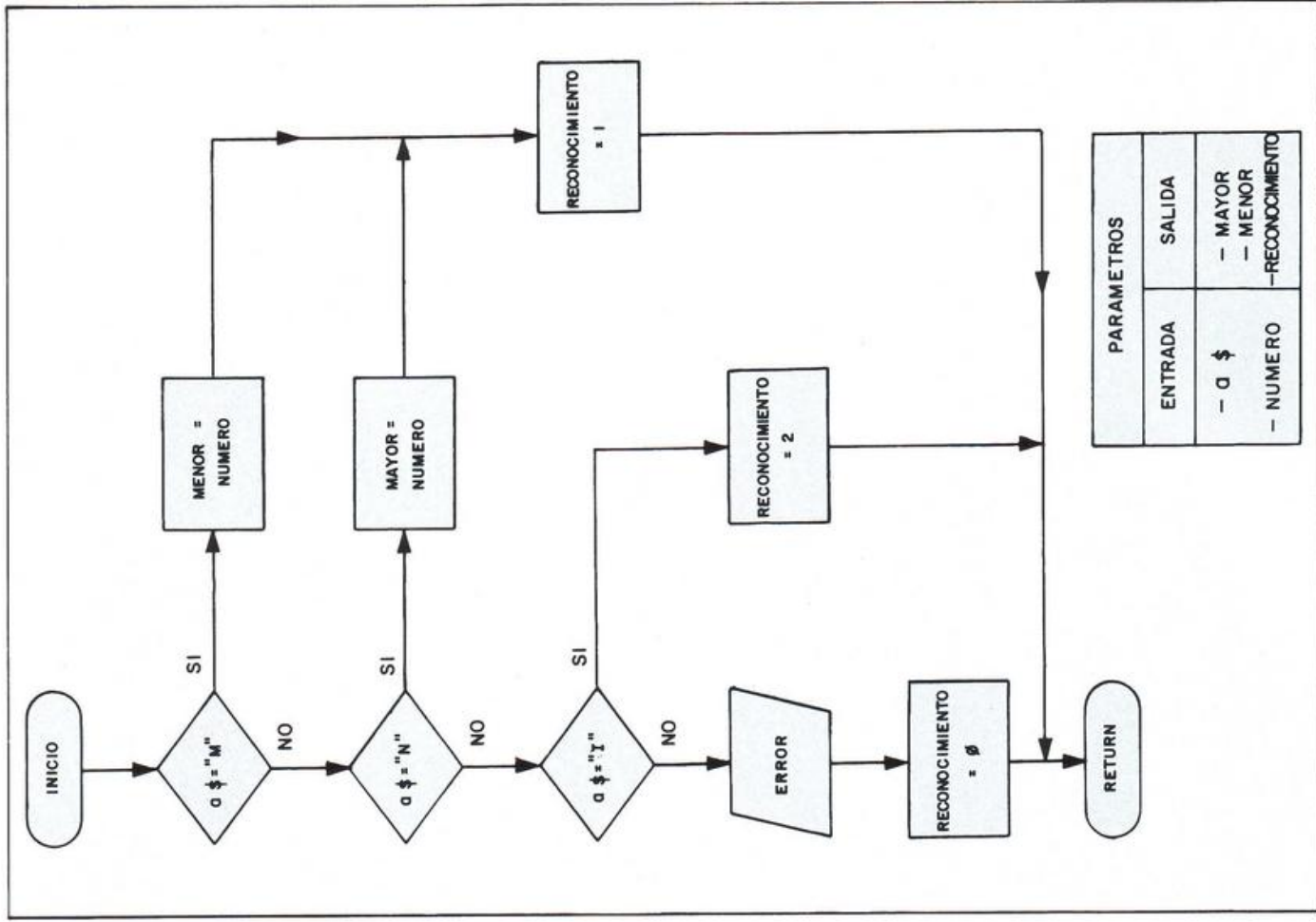
```
10 READ código
20 READ precio
30 PRINT código
40 PRINT precio
50 RESTORE
60 READ número
```

La sentencia «RESTORE» de la línea 50, permite leer la ta-bla dos veces. En la primera lectura, el valor «27» es asig-nado a la variable «código»; sin embargo, en la segunda se asigna a la variable «número»; otro tanto ocurre con el valor «3500» que se asigna primera-mente a la variable «precio», y posteriormente a «valor».

En el siguiente programa tiene como argumento la sen-tencia «RESTORE», una varia-ble de tipo numérico, cuyo va-



Lectura de datos.



Programa «Adivino» subrutina «Verificación».


```

10 REM *****
11 * CURSO/BASIC *
12 *****
13 * LONGITUD *
14 *****
15
20 BORDER 4: PAPER 4: INK 1: C
30 REM

40 LET MENU=1000
50 LET VISUALIZACION=2000
100 REM

*****
* VARIABLES *
*****
* BUCLE CENTRAL *
*****
*****

110 GO SUB MENU
120 INPUT "Unidad a transformar";UNID
125 IF UNID<1 OR UNID>7 THEN CLS
130 CLS
140 INPUT "Cantidad >>";VALOR
150 IF VALOR<0 OR VALOR>9999999 THEN GO TO 140
160 LET OPERACION=1000+(UNID*100)
170 GO SUB OPERACION
180 GO SUB VISUALIZACION
190 INPUT "Quiere continuar (S/N)";LINE$
200 IF LINE$="S" OR LINE$="Y" THEN CLS
210 IF LINE$="N" OR LINE$="N" THEN STOP
220 GO TO 100
999 REM

*****
* SUBROUTINAS *
*****
*****

1000 REM *****
1010 PRINT AT 1,5;"UNIDADES DE L";
1020 PRINT AT 2,5;" "
1030 PRINT AT 6,8;"1 - Milimetro";
1040 PRINT AT 8,8;"2 - Centimetro";
1050 PRINT AT 10,8;"3 - Decimetro";
1060 PRINT AT 12,8;"4 - Metro";
1070 PRINT AT 14,8;"5 - Decametro";
1080 PRINT AT 16,8;"6 - Hectometro";
1090 PRINT AT 18,8;"7 - Kilometro";
1095 RETURN
1100 REM *****
1110 LET VALOR=VALOR
1120 LET VALOR=VALOR/1000
1130 LET DECIMETRO=CENTIMETRO/10
1140 LET METRO=DECIMETRO/10

```

```

1150 LET DECAMETRO=METRO/10
1160 LET HECTOMETRO=DECAMETRO/10
1170 LET KILOMETRO=HECTOMETRO/10
1180 RETURN
1190 REM *****
1200 LET CENTIMETRO=VALOR
1210 LET METRO=CENTIMETRO/10
1220 LET DECIMETRO=METRO/10
1230 LET HECTOMETRO=DECIMETRO/10
1240 LET KILOMETRO=HECTOMETRO/10
1250 RETURN
1260 REM *****
1270 LET DECIMETRO=VALOR
1280 LET METRO=DECIMETRO/10
1290 LET HECTOMETRO=METRO/10
1300 LET KILOMETRO=HECTOMETRO/10
1310 RETURN
1320 LET METRO=VALOR
1330 LET CENTIMETRO=METRO*100
1340 LET DECIMETRO=CENTIMETRO/10
1350 LET HECTOMETRO=DECIMETRO/10
1360 LET KILOMETRO=HECTOMETRO/10
1370 RETURN
1380 REM *****
1390 LET METRO=VALOR
1400 LET CENTIMETRO=METRO*1000
1410 LET DECIMETRO=CENTIMETRO/100
1420 LET HECTOMETRO=DECIMETRO/10
1430 LET KILOMETRO=HECTOMETRO/10
1440 LET METRO=HECTOMETRO/10
1450 LET DECAMETRO=METRO/10
1460 LET HECTOMETRO=DECAMETRO/10
1470 LET KILOMETRO=HECTOMETRO/10
1480 RETURN
1490 REM *****
1500 LET DECAMETRO=VALOR
1510 LET METRO=DECAMETRO*100
1520 LET CENTIMETRO=DECAMETRO*10
1530 LET KILOMETRO=CENTIMETRO*10
1540 RETURN
1550 LET DECIMETRO=VALOR
1560 LET METRO=DECIMETRO*100
1570 LET HECTOMETRO=METRO/10
1580 LET KILOMETRO=HECTOMETRO/10
1590 RETURN
1600 REM *****
1610 LET HECTOMETRO=VALOR
1620 LET METRO=HECTOMETRO*100
1630 LET CENTIMETRO=HECTOMETRO*10
1640 LET KILOMETRO=CENTIMETRO*10
1650 RETURN
1660 LET METRO=HECTOMETRO*1000
1670 LET DECAMETRO=METRO/10
1680 LET KILOMETRO=DECAMETRO/10
1690 RETURN
1700 REM *****
1710 LET KILOMETRO=VALOR
1720 LET METRO=KILOMETRO*1000
1730 LET CENTIMETRO=KILOMETRO*100
1740 LET DECIMETRO=KILOMETRO*10
1750 LET METRO=KILOMETRO*1000
1760 LET DECAMETRO=KILOMETRO*100
1770 LET KILOMETRO=KILOMETRO*10
1780 RETURN
1790 REM *****
1800 PRINT AT 3,10;"RESULTADOS";
1810 PRINT AT 4,10;"Milimetros";
1820 PRINT AT 5,10;"Centimetros";
1830 PRINT AT 6,10;"Decimetros";
1840 PRINT AT 7,10;"Metros";
1850 PRINT AT 8,10;"Decametros";
1860 PRINT AT 9,10;"Hectometros";
1870 PRINT AT 10,10;"Kilometros";
1880 RETURN

```

líneas del programa, hasta que encuentre la primera que contiene «DATA»; posteriormente, asigna a la variable el primer dato de ésta. La siguiente vez que encuentra «READ» se le asigna el segundo dato, y así sucesivamente.

Ejecute el siguiente programa:

```

10 REM *****
11 * BUCLE READ *
12 *****
13 LET TOTAL=0
14 READ VALOR
15 PRINT VALOR
16 LET TOTAL=TOTAL+VALOR
17 NEXT VALOR
18 PRINT TOTAL
19
20 DATA "MICROHOBBY", "SEMANAL"
21 DATA "SPECTRUM", "12 INCLIN"

```

según se van ejecutando las sentencias «READ», se van asignando los sucesivos datos, así:

a\$ tendrá el valor «MICROHOBBY»

b\$ = «SEMANAL»

número = 1985

f\$ = «BASIC/SINCLAIR»

q\$ = «SPECTRUM»

valor = 16

Para conocer el ordenador cuál es el siguiente dato que tiene que leer, la tabla de datos tiene un *puntero* o *índice* que lo señala y que se incrementa en uno cada vez que se hace una lectura.

Una cosa importante es que no pueden realizarse más lecturas que datos haya en el programa, ya que tendríamos un mensaje de error; sin embargo, puede haber más datos que lecturas.

La instrucción «READ» puede formar parte de un bucle,

de esta manera puede agilizar se la lectura de datos.

Ejemplo:

```

10 REM *****
11 * BUCLE READ *
12 *****
13 LET TOTAL=0
14 READ VALOR
15 PRINT VALOR
16 LET TOTAL=TOTAL+VALOR
17 NEXT VALOR
18 PRINT TOTAL
19
20 DATA 23,45,67,45,21,67,89,3
21 DATA 12,45,63,27,54,12,67,7
22 DATA 34,56,71,89,21,83

```

Dentro de una tabla de datos puede leerse uno determinado; en la siguiente aplicación, introduciendo un número entre 1 y 12, el ordenador nos visualiza el mes correspondiente:

```

10 REM *****
11 * MESES *
12 *****
20 INPUT "Mes (1-12) >>";MES
30 IF MES<1 OR MES>12 THEN GO TO 20
40 LET PUNTERO=1
50 IF PUNTERO>12 THEN GO TO 50
60 READ VALOR
70 PRINT VALOR
80 LET PUNTERO=PUNTERO+1
90 REM *****
100 DATA "ENERO", "FEBRERO", "MARZO", "ABRIL", "MAYO", "JUNIO", "JULIO", "AGOSTO", "SEPTIEMBRE", "OCTUBRE", "NOVIEMBRE", "DICIEMBRE"

```

Veamos otro ejemplo de aplicación de las sentencias «READ/DATA».

```

10 REM *****
11 * PRIMOS *
12 *****
15 PRINT "Numero Primo", "Cuadrado"
20 DATA 2,3,5,7,11,13,17,19

```

```

17 PRINT " "
18 IF NOT $150 THEN STOP
19 PRINT " "
20 PRINT " "
21 GO TO 20
22 REM *****
23 * DATOS *
24 *****
25 DATA 2,3,5,7,11,13,17,19
26 DATA 4,9,16,25,36,49,64,81

```

En este caso, el ordenador calcula el cuadrado de los números primos menores de cincuenta, que se han introducido como constantes en una serie de «DATA». Al final de los datos se introduce el valor «0», como código de ruptura; cuando se realiza la lectura de este código, el programa se para, de esta manera no es necesario llevar la cuenta de la cantidad de datos. Para preguntar por esta condición se ha utilizado:

IF NOT primo THEN STOP

la condición se hace verdadera y por tanto se ejecuta «STOP», cuando la variable «primo» es igual a «0», ver función «NOT» (pág. 35).

RESTORE

Acceso al teclado



Tipo de sentencia

Comando de programación.

DATOS DE UN PROGRAMA

Cuando estudiamos las sentencias «LET» e «INPUT», vimos que se podían asignar o introducir datos en nuestros programas; pero cuando estos datos son de valor constante y numerosos, podemos utilizar las sentencias «DATA» para almacenarlos y «READ» para leerlos.

READ

Acceso al teclado



MODO E

- READ j\$.
 - READ numero, a\$, nota.
 - READ valor.
- El dato es asignado a la variable del argumento.

DATA

Acceso al teclado

50 DATA 10, 20, 67
100 DATA 5, 15, 27, 33
170 DATA «MICRO», «CHIP»
300 DATA «ENTRADA», 30

aunque por razones de claridad y estructuración del programa, conviene que estén todos juntos al principio o a final de éste.

Ejemplo:

1000 DATA 10, 20, 67, 5
1010 DATA 15, 27, 33
1020 DATA «MICRO», «CHIP», «ENTRADA»
1030 DATA 30

La cantidad de valores a incluir dentro de una instrucción «DATA», viene determinada únicamente por la capacidad de una línea (22 filas).

Utilización de «READ» y «DATA»

Cuando el intérprete BASIC encuentra una sentencia del tipo:

READ variable

analiza una por una todas las

SENTENCIA	ARGUMENTO
DATA	constante, constante...

Las constantes pueden ser numéricas o alfanuméricas.

Ejemplos:

- DATA 30, 47, 3225, 17.14
- DATA «MICROHOBBY», «CURSO», «BASIC»
- DATA «SPECTRUM», 23, 10, «MICRO»
- DATA 8

Tipo de sentencia

Comando de entrada.

Definición

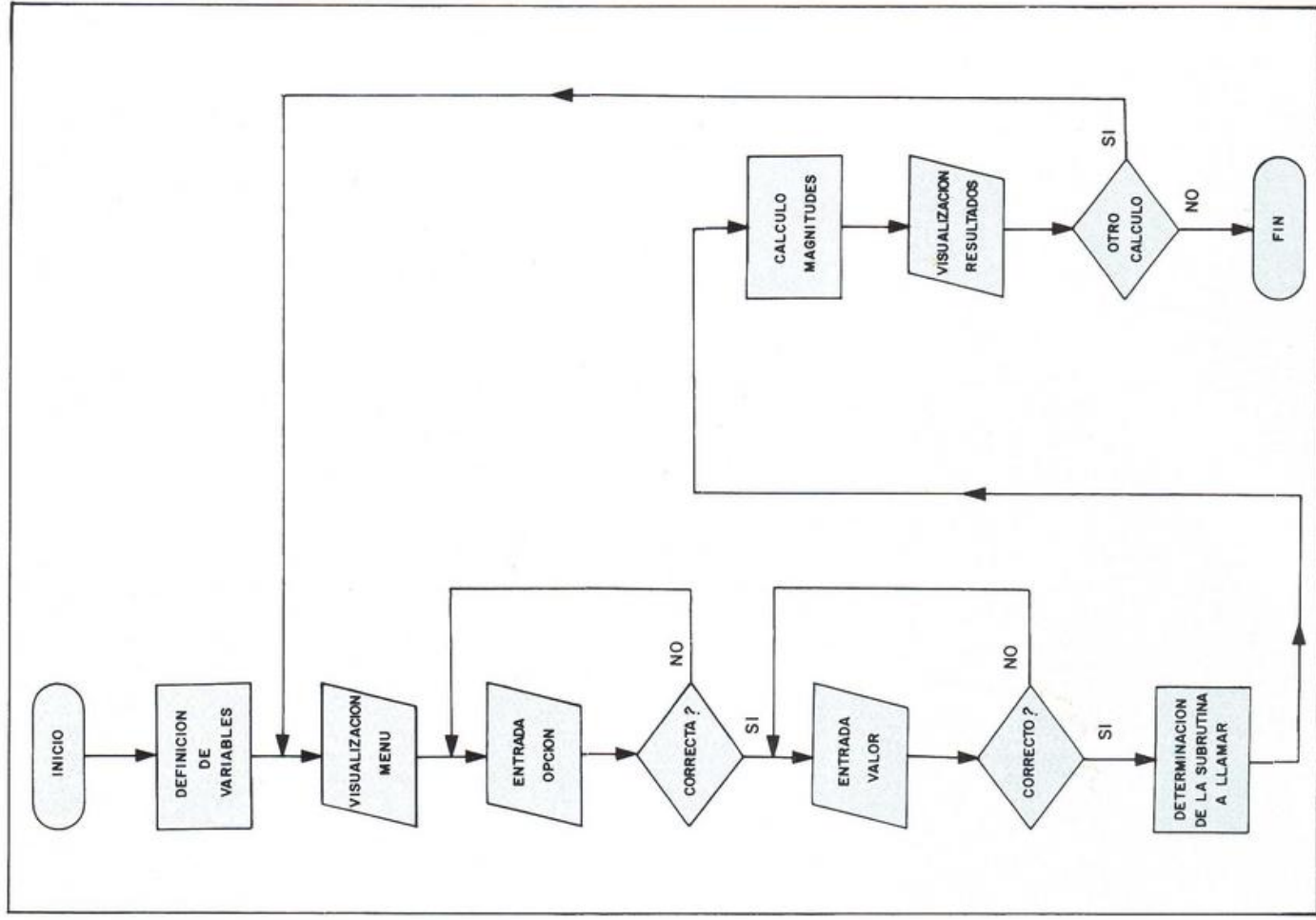
Con esta sentencia, se pueden leer los datos que haya dentro de un programa; su estructura es:

SENTENCIA	ARGUMENTO
READ	variable, variable...

Las variables pueden ser tanto numéricas como de cadena, pero teniendo en cuenta que deben ir en concordancia con el dato leído, y separadas por comas.

Ejemplos:

- READ día, mes.



Estructura programa «longitud».


```

15 BORDER 4
20 INPUT "TEMPORIZACION (1-10) ";
30 IF 1<tempo<10 THEN
40 GO TO 20
50 BORDER 1
60 BORDER 1
70 BORDER 1
80 PRINT "PAUSA"
90 PRINT "00 AT 1.00"
100 PRINT "00"
110 IF 1<tempo<10 THEN
120 IF 1<tempo<10 THEN
130 GO TO 20
140 GO TO 20
150 GO TO 20

```

- 30 : Inicialización del cursor en la posición 0, 0.
- 40 : Visualización del cursor. Se utiliza uno de los gráficos predefinidos.
- 50 : Pausa hasta que se pulsa una tecla.
- 60-100 : Verificación de la tecla pulsada. Su código ASCII debe estar comprendido entre el número 32 (espacio) y el 122 (z) o ser uno de los códigos de control del programa:

"<" = (SYMBOL SHIFT + Q)
 ">" = (SYMBOL SHIFT + W)
 "=" = (SYMBOL SHIFT + E)

PROGRAMAS

El programa número «1» simula una máquina de escribir. los controles son:

- SYMBOL SHIFT + Q, borra el carácter anterior.
- SYMBOL SHIFT + W, provoca un retorno de carro.
- SYMBOL SHIFT + E, termina la edición de la página.

Una vez editada la página, puede sacarse una copia por impresora seleccionando la opción «L» o grabarse en cinta con «G». La opción «C» permite editar una nueva página o terminar.

La estructura es la siguiente:

- 10 : Comentario con el nombre del programa.
- 20 : Asignación del color rojo para borde, azul para el fondo y blanco para los caracteres.

- guiente línea, simulando el retorno de carro de una máquina de escribir (control = SYMBOL SHIFT + W).
- 1105 : Borrado del cursor.
- 1110 : Inicialización de la nueva posición del cursor.
- 1120 : Salto a la línea que comprueba el final de página y línea.
- 1205 : Borrado del cursor.
- 1210 : Visualización de las opciones:

- L - Listar
- G - Grabar
- C - Continuar
- 1220-1250 : Comprobación de la opción elegida. Imprimir el contenido de la pantalla. Se utiliza la sentencia «COPY».
- 1310 : Salto a la visualización de las opciones.
- 1350-1380 : Grabar en cinta el contenido de la pantalla. Debe introducirse previamente el nombre que deseamos asignarle. Se utiliza como argumento de «SAVE» la palabra clave «SCREEN».
- 1390 : Salto a la visualización de opciones.
- 1500-1550 : Decisión para editar una nueva página o no.

El programa número «2» permite desplazar, con ayuda de unas teclas utilizadas como cursor, un asterisco (*) a través de la pantalla, que va dejando un rastro de puntos (.) por donde va pasando.

lor está en función de un «INPUT»; dependiendo de éste la inicialización del puntero y, por tanto, del acceso a los datos.

```

10 REM *****
11 RESTORE *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 INPUT "CLAVE (1 a 5) ";
30 IF CLAVE=1 OR CLAVE=5 THEN
40 GO TO 20
50 RESTORE *****
60 *****
70 *****
80 *****
90 *****
100 *****
110 *****
120 *****
130 *****
140 *****
150 *****
160 *****
170 *****
180 *****
190 *****
200 *****
210 *****
220 *****
230 *****
240 *****
250 *****
260 *****
270 *****
280 *****
290 *****
300 *****
310 *****
320 *****
330 *****
340 *****
350 *****
360 *****
370 *****
380 *****
390 *****
400 *****
410 *****
420 *****
430 *****
440 *****
450 *****
460 *****
470 *****
480 *****
490 *****
500 *****
510 *****
520 *****
530 *****
540 *****
550 *****
560 *****
570 *****
580 *****
590 *****
600 *****
610 *****
620 *****
630 *****
640 *****
650 *****
660 *****
670 *****
680 *****
690 *****
700 *****
710 *****
720 *****
730 *****
740 *****
750 *****
760 *****
770 *****
780 *****
790 *****
800 *****
810 *****
820 *****
830 *****
840 *****
850 *****
860 *****
870 *****
880 *****
890 *****
900 *****
910 *****
920 *****
930 *****
940 *****
950 *****
960 *****
970 *****
980 *****
990 *****
1000 *****

```

Errores

Cuando se manejan sentencias del tipo READ/DATA, hay tres tipos de error que suelen producirse frecuentemente.

- a) Cuando se ejecuta una sentencia «READ» y el puntero se encuentra al final de la tabla, indicando que no hay más datos. El mensaje que se visualiza es:

E Out of DATA

Ejemplo:

```

10 REM *****
11 ERROR 1 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****

```

se pretende leer cinco datos, cuando en realidad sólo hay cuatro.

- b) Cuando al confeccionar la tabla de datos se incluye un valor numérico en lugar de uno alfanumérico, o viceversa. Al ejecutarse di-

500 DATA
30
127
200
510 DATA
60
32
520 DATA
114
123
15
62
530 DATA
94
DATOS

INDICE

Mecanismo «Restore 520».

cho programa y tratar de leer un dato que no está en concordancia con el de la variable que acompaña a «READ», se produce el error:

C Nonsense in BASIC

Ejemplo:

```

10 REM *****
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****

```

En la línea 140, el segundo dato debería ser alfanumérico.

PROGRAMA 1

```

10 REM
*****
** CURSO/BASIC **
*****
** GEOGRAFIA 1 **
*****
**
**
**
20 BORDER 4: PAPER 4: INK 1: C
30 GO SUB 1000
40 REM
*****
** BUCLE PRINCIPAL **
*****
**
**
**
50 PRINT AT 1,3,"AUTONOMIAS DE
LA PENINSULA,"
60 PRINT AT 2,3,"
70 FOR c=5 TO 19
80 READ a$
90 LET opcion=c-4
100 PRINT AT c,3;opcion;
110 IF opcion<10 THEN PRINT " "
120 PRINT " " ;a$
130 INPUT CAT 0,1,"Que Autonoma
desea visualizar?"
140 IF opcion<1 OR opcion>15 TH
EN GO TO 140
150 CLS
160 LET direccion=2000+(opcion*
100)
170 RESTORE direccion
180 READ a$
190 PRINT AT 1,3;a$
210 PRINT " "
220 READ numero
230 FOR n=1 TO numero
240 PRINT " " ;n
250 READ a$,b$
260 PRINT a$,"(" ;b$;") "
270 NEXT n
280 INPUT "ENTER" para
retornar a LINE a$
290 RESTORE 1100
300 CLS
310 GO TO 40
1000 REM
*****
** DEFINICION "N" **
*****
**
**
**
1010 RESTORE 1060
1020 FOR d=USR "n" TO USR "n"+7
1030 READ dato
1040 POKE d,dato
1050 NEXT d
1060 DATA 24,66,98,82,74,70,66,0
1070 RETURN
1100 REM
*****
** MENU **
*****
**
**
**
1110 DATA "ANDALUCIA","ARAGON","
ASTURIAS","CANTABRIA","CASTILLA-
LA MANCHA","
1120 DATA "CASTILLA-LEON","CATAL
UNA","COMUNIDAD VALENCIANA","EXT
REMOURA","GALICIA","
1130 DATA MADRID","MURCIA","NAV

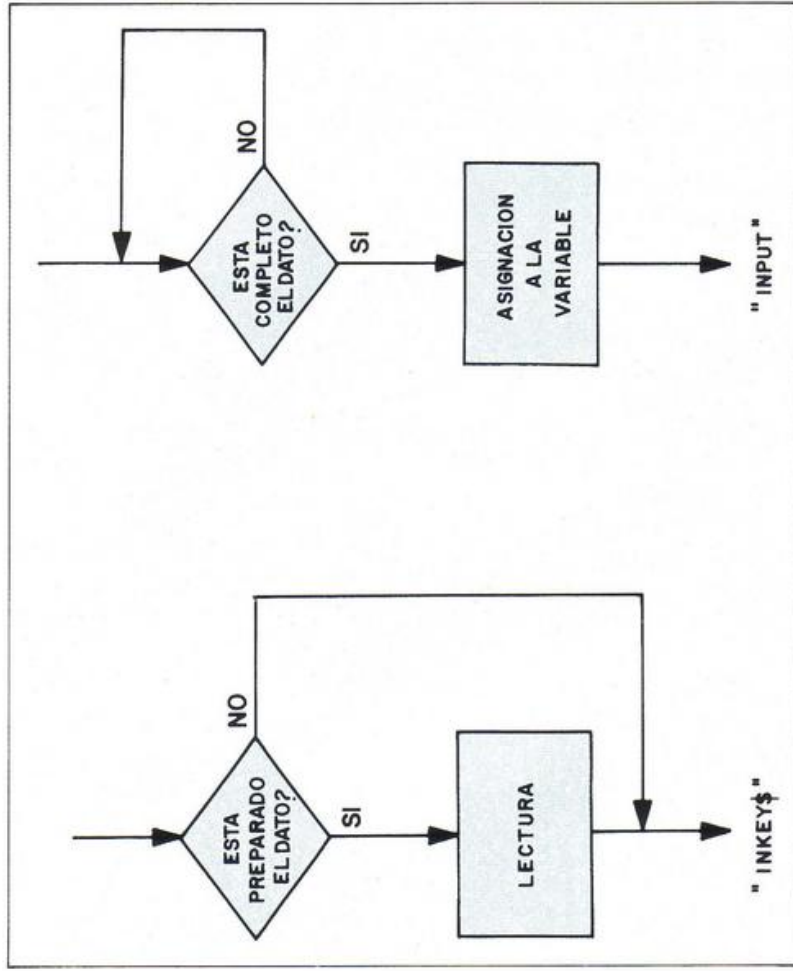
```

```

ARRA", "PAIS VASCO", "LA RIOJA",
2000 REM
*****
* PROVINCIAS *
*****
2100 DATA "ANDALUCIA", 8, "ALMERIA",
DORA", "CORDOBA", "GRANADA", "GRANADA",
EN", "HUELVA", "HUELVA", "JAEN", "JAEN",
SEVILLA",
2200 DATA "ARAGON", 3, "HUESCA",
UESA", "ZARAGOZA", "TERUEL", "ZARAGOZA",
2300 DATA "ASTURIAS", 1, "ASTURIA",
2400 DATA "CANTABRIA", 1, "CANTABRIA",
IA", "SANTANDER",
2500 DATA "CASTILLA-LA MANCHA",
ALBACETE", "ALBACETE", "CIUDAD REAL",
EAL", "CIUDAD REAL", "CIUDAD REAL",
NCA", "GUADALAJARA", "GUADALAJARA",
TOLEDO",
2600 DATA "CASTILLA-LEON", 9, "AVILA",
LA", "AVILA", "BURGOS", "BURGOS",
EON", "LEON", "PALENCIA", "PALENCIA",
IA", "SALAMANCA", "SALAMANCA", "SEGOVIA",
IA", "SEGOVIA", "SORIA", "SORIA",
AL", "ZAMORA", "VALLADOLID", "ZAMORA",
2700 DATA "CATALUNA", 4, "BARCELONA",
TARRAGONA", "LEIDA", "TARRAGONA",
TARRAGONA",
2800 DATA "COMUNIDAD VALENCIANA",
3, "ALICANTE", "ALICANTE", "CASPE",
LON", "C DE LA PLANA", "VALENCIA",
VALENCIA", "EXTREMADURA", 2, "BADAJOZ",
2900 DATA "EXTREMADURA", 2, "BADAJOZ",
3000 DATA "GALICIA", 4, "LA CORUNA",
NSE", "ORENSE", "PONTEVEDRA", "PONTEVEDRA",
3100 DATA "MADRID", 1, "MADRID",
ADRID",
3200 DATA "MURCIA", 1, "MURCIA",
URCIA",
3300 DATA "NAVARRA", 1, "NAVARRA",
"PAAMPLONA",
3400 DATA "PAIS VASCO", 3, "ALAVA",
3500 DATA "GUIPUZCOA", "GUIPUZCOA",
3600 DATA "LA RIOJA", 1, "LA RIOJA",
3700 DATA "MURCIA", 1, "MURCIA",
3800 DATA "MURCIA", 1, "MURCIA",
3900 DATA "MURCIA", 1, "MURCIA",
4000 DATA "MURCIA", 1, "MURCIA",
4100 DATA "MURCIA", 1, "MURCIA",
4200 DATA "MURCIA", 1, "MURCIA",
4300 DATA "MURCIA", 1, "MURCIA",
4400 DATA "MURCIA", 1, "MURCIA",
4500 DATA "MURCIA", 1, "MURCIA",
4600 DATA "MURCIA", 1, "MURCIA",
4700 DATA "MURCIA", 1, "MURCIA",
4800 DATA "MURCIA", 1, "MURCIA",
4900 DATA "MURCIA", 1, "MURCIA",
5000 DATA "MURCIA", 1, "MURCIA",
5100 DATA "MURCIA", 1, "MURCIA",
5200 DATA "MURCIA", 1, "MURCIA",
5300 DATA "MURCIA", 1, "MURCIA",
5400 DATA "MURCIA", 1, "MURCIA",
5500 DATA "MURCIA", 1, "MURCIA",
5600 DATA "MURCIA", 1, "MURCIA",
5700 DATA "MURCIA", 1, "MURCIA",
5800 DATA "MURCIA", 1, "MURCIA",
5900 DATA "MURCIA", 1, "MURCIA",
6000 DATA "MURCIA", 1, "MURCIA",
6100 DATA "MURCIA", 1, "MURCIA",
6200 DATA "MURCIA", 1, "MURCIA",
6300 DATA "MURCIA", 1, "MURCIA",
6400 DATA "MURCIA", 1, "MURCIA",
6500 DATA "MURCIA", 1, "MURCIA",
6600 DATA "MURCIA", 1, "MURCIA",
6700 DATA "MURCIA", 1, "MURCIA",
6800 DATA "MURCIA", 1, "MURCIA",
6900 DATA "MURCIA", 1, "MURCIA",
7000 DATA "MURCIA", 1, "MURCIA",
7100 DATA "MURCIA", 1, "MURCIA",
7200 DATA "MURCIA", 1, "MURCIA",
7300 DATA "MURCIA", 1, "MURCIA",
7400 DATA "MURCIA", 1, "MURCIA",
7500 DATA "MURCIA", 1, "MURCIA",
7600 DATA "MURCIA", 1, "MURCIA",
7700 DATA "MURCIA", 1, "MURCIA",
7800 DATA "MURCIA", 1, "MURCIA",
7900 DATA "MURCIA", 1, "MURCIA",
8000 DATA "MURCIA", 1, "MURCIA",
8100 DATA "MURCIA", 1, "MURCIA",
8200 DATA "MURCIA", 1, "MURCIA",
8300 DATA "MURCIA", 1, "MURCIA",
8400 DATA "MURCIA", 1, "MURCIA",
8500 DATA "MURCIA", 1, "MURCIA",
8600 DATA "MURCIA", 1, "MURCIA",
8700 DATA "MURCIA", 1, "MURCIA",
8800 DATA "MURCIA", 1, "MURCIA",
8900 DATA "MURCIA", 1, "MURCIA",
9000 DATA "MURCIA", 1, "MURCIA",
9100 DATA "MURCIA", 1, "MURCIA",
9200 DATA "MURCIA", 1, "MURCIA",
9300 DATA "MURCIA", 1, "MURCIA",
9400 DATA "MURCIA", 1, "MURCIA",
9500 DATA "MURCIA", 1, "MURCIA",
9600 DATA "MURCIA", 1, "MURCIA",
9700 DATA "MURCIA", 1, "MURCIA",
9800 DATA "MURCIA", 1, "MURCIA",
9900 DATA "MURCIA", 1, "MURCIA",
1000 DATA "MURCIA", 1, "MURCIA",
1010 DATA "MURCIA", 1, "MURCIA",
1020 DATA "MURCIA", 1, "MURCIA",
1030 DATA "MURCIA", 1, "MURCIA",
1040 DATA "MURCIA", 1, "MURCIA",
1050 DATA "MURCIA", 1, "MURCIA",
1060 DATA "MURCIA", 1, "MURCIA",
1070 DATA "MURCIA", 1, "MURCIA",
1080 DATA "MURCIA", 1, "MURCIA",
1090 DATA "MURCIA", 1, "MURCIA",
1100 DATA "MURCIA", 1, "MURCIA",
1110 DATA "MURCIA", 1, "MURCIA",
1120 DATA "MURCIA", 1, "MURCIA",
1130 DATA "MURCIA", 1, "MURCIA",
1140 DATA "MURCIA", 1, "MURCIA",
1150 DATA "MURCIA", 1, "MURCIA",
1160 DATA "MURCIA", 1, "MURCIA",
1170 DATA "MURCIA", 1, "MURCIA",
1180 DATA "MURCIA", 1, "MURCIA",
1190 DATA "MURCIA", 1, "MURCIA",
1200 DATA "MURCIA", 1, "MURCIA",
1210 DATA "MURCIA", 1, "MURCIA",
1220 DATA "MURCIA", 1, "MURCIA",
1230 DATA "MURCIA", 1, "MURCIA",
1240 DATA "MURCIA", 1, "MURCIA",
1250 DATA "MURCIA", 1, "MURCIA",
1260 DATA "MURCIA", 1, "MURCIA",
1270 DATA "MURCIA", 1, "MURCIA",
1280 DATA "MURCIA", 1, "MURCIA",
1290 DATA "MURCIA", 1, "MURCIA",
1300 DATA "MURCIA", 1, "MURCIA",
1310 DATA "MURCIA", 1, "MURCIA",
1320 DATA "MURCIA", 1, "MURCIA",
1330 DATA "MURCIA", 1, "MURCIA",
1340 DATA "MURCIA", 1, "MURCIA",
1350 DATA "MURCIA", 1, "MURCIA",
1360 DATA "MURCIA", 1, "MURCIA",
1370 DATA "MURCIA", 1, "MURCIA",
1380 DATA "MURCIA", 1, "MURCIA",
1390 DATA "MURCIA", 1, "MURCIA",
1400 DATA "MURCIA", 1, "MURCIA",
1410 DATA "MURCIA", 1, "MURCIA",
1420 DATA "MURCIA", 1, "MURCIA",
1430 DATA "MURCIA", 1, "MURCIA",
1440 DATA "MURCIA", 1, "MURCIA",
1450 DATA "MURCIA", 1, "MURCIA",
1460 DATA "MURCIA", 1, "MURCIA",
1470 DATA "MURCIA", 1, "MURCIA",
1480 DATA "MURCIA", 1, "MURCIA",
1490 DATA "MURCIA", 1, "MURCIA",
1500 DATA "MURCIA", 1, "MURCIA",
1510 DATA "MURCIA", 1, "MURCIA",
1520 DATA "MURCIA", 1, "MURCIA",
1530 DATA "MURCIA", 1, "MURCIA",
1540 DATA "MURCIA", 1, "MURCIA",
1550 DATA "MURCIA", 1, "MURCIA",
1560 DATA "MURCIA", 1, "MURCIA",
1570 DATA "MURCIA", 1, "MURCIA",
1580 DATA "MURCIA", 1, "MURCIA",
1590 DATA "MURCIA", 1, "MURCIA",
1600 DATA "MURCIA", 1, "MURCIA",
1610 DATA "MURCIA", 1, "MURCIA",
1620 DATA "MURCIA", 1, "MURCIA",
1630 DATA "MURCIA", 1, "MURCIA",
1640 DATA "MURCIA", 1, "MURCIA",
1650 DATA "MURCIA", 1, "MURCIA",
1660 DATA "MURCIA", 1, "MURCIA",
1670 DATA "MURCIA", 1, "MURCIA",
1680 DATA "MURCIA", 1, "MURCIA",
1690 DATA "MURCIA", 1, "MURCIA",
1700 DATA "MURCIA", 1, "MURCIA",
1710 DATA "MURCIA", 1, "MURCIA",
1720 DATA "MURCIA", 1, "MURCIA",
1730 DATA "MURCIA", 1, "MURCIA",
1740 DATA "MURCIA", 1, "MURCIA",
1750 DATA "MURCIA", 1, "MURCIA",
1760 DATA "MURCIA", 1, "MURCIA",
1770 DATA "MURCIA", 1, "MURCIA",
1780 DATA "MURCIA", 1, "MURCIA",
1790 DATA "MURCIA", 1, "MURCIA",
1800 DATA "MURCIA", 1, "MURCIA",
1810 DATA "MURCIA", 1, "MURCIA",
1820 DATA "MURCIA", 1, "MURCIA",
1830 DATA "MURCIA", 1, "MURCIA",
1840 DATA "MURCIA", 1, "MURCIA",
1850 DATA "MURCIA", 1, "MURCIA",
1860 DATA "MURCIA", 1, "MURCIA",
1870 DATA "MURCIA", 1, "MURCIA",
1880 DATA "MURCIA", 1, "MURCIA",
1890 DATA "MURCIA", 1, "MURCIA",
1900 DATA "MURCIA", 1, "MURCIA",
1910 DATA "MURCIA", 1, "MURCIA",
1920 DATA "MURCIA", 1, "MURCIA",
1930 DATA "MURCIA", 1, "MURCIA",
1940 DATA "MURCIA", 1, "MURCIA",
1950 DATA "MURCIA", 1, "MURCIA",
1960 DATA "MURCIA", 1, "MURCIA",
1970 DATA "MURCIA", 1, "MURCIA",
1980 DATA "MURCIA", 1, "MURCIA",
1990 DATA "MURCIA", 1, "MURCIA",
2000 DATA "MURCIA", 1, "MURCIA",
2010 DATA "MURCIA", 1, "MURCIA",
2020 DATA "MURCIA", 1, "MURCIA",
2030 DATA "MURCIA", 1, "MURCIA",
2040 DATA "MURCIA", 1, "MURCIA",
2050 DATA "MURCIA", 1, "MURCIA",
2060 DATA "MURCIA", 1, "MURCIA",
2070 DATA "MURCIA", 1, "MURCIA",
2080 DATA "MURCIA", 1, "MURCIA",
2090 DATA "MURCIA", 1, "MURCIA",
2100 DATA "MURCIA", 1, "MURCIA",
2110 DATA "MURCIA", 1, "MURCIA",
2120 DATA "MURCIA", 1, "MURCIA",
2130 DATA "MURCIA", 1, "MURCIA",
2140 DATA "MURCIA", 1, "MURCIA",
2150 DATA "MURCIA", 1, "MURCIA",
2160 DATA "MURCIA", 1, "MURCIA",
2170
```

c) En las tablas de datos con valores de cadena, puede suceder que se nos olvide

grama, tomará dicho dato como variable numérica; si no existe ninguna variable



Diferencias entre "INKEYS" e "INPUT".

10 FOR x = 1 TO tiempo

Integer out of range

La estructura de este comando es:

SENTENCIA	ARGUMENTO
PAUSE	expresión numérica

Ejemplos:

- PAUSE 10.
- PAUSE 950.
- PAUSE 30 * 5.

Si se introduce un número fraccionario, éste se redondea al valor entero más cercano. El rango de valores está comprendido entre 0 y 65535, cuando se introduce uno fuera de margen aparece el mensaje.

La realización que existe entre el número del argumento y el tiempo de temporización, en segundos es:

$$\text{tiempo} = \frac{n}{50}$$

por lo tanto «PAUSE 1000», detiene la ejecución del programa durante «2» segundos. Dividiendo entre 50 el valor máximo (65535) obtenemos el mayor tiempo de temporización:

$$65535/50 = 1310.7 \text{ sg.}$$

21 min. 50.7 sg.

Si durante la ejecución de una sentencia «PAUSE» se presiona una tecla, la temporización se interrumpe y continúa el programa en la siguiente instrucción.

El valor « ϕ », asignado a «PAUSE», provoca una temporización indefinida hasta que se pulsa una tecla.

Puede utilizarse una variable del tipo numérico como argumento de esta sentencia.
Ejemplo:

Ejemplo:

```

10 REM *****
      PAUSA VARIABLE
*****

```


Para un ángulo comprendido entre «0» y «180» grados el valor del seno es positivo, y negativo entre «180» y «360».

COS

Acceso al teclado



ACS

Definición

La función «COS» calcula el coseno de un ángulo, éste debe estar expresado en radianes.

Ejemplo:

- LET n = COS 2
- PRINT COS (17 * PI/180)
- LET valor = COS total
- PRINT COS (320 * PI/180)

El coseno de uno de los ángulos de un triángulo rectángulo es la razón que hay entre el cateto adyacente y la hipotenusa.

El valor del seno es positivo para un ángulo comprendido entre «0» y «90» o entre «270» y «360». Es negativo entre «90» y «270».

TAN

Acceso al teclado

TAN



ATN

MODO E

El valor del argumento debe estar comprendido entre +1 y -1, de lo contrario, se visualizará el mensaje de error.

A Invalid argument

Definición

«TAN» retorna la tangente de un ángulo expresado en radianes.

Ejemplos:

- PRINT TAN (45 * PI/180)
- LET C = TAN alfa
- PRINT TAN 1
- LET d = TAN (beta + 2)

La tangente de un ángulo es la razón que hay entre el cateto opuesto y el cateto adyacente, de un triángulo rectángulo.

ASN

Acceso al teclado

SIN



ASN

Definición

La función «ASN» calcula el arcoseno, es decir, el valor de un ángulo a partir de su seno. El valor retornado está expresado en radianes.

Ejemplos:

- LET a = ASN 0,5
- PRINT 180/PI * ASN 1
- LET C = ASN (alfa) * 180/PI
- PRINT ASN 0,7

El valor del argumento debe estar comprendido entre +1 y -1, de lo contrario, se visualizará el mensaje de error.

MODO E



ATN

PROGRAMA 1

```

10 REM
*****
* CURSOR/BASIC
*****
*
* MAQUINA
*****
*
LS 20 BORDER 2: PAPER 1: INK 7: C
30 LET X=0: LET Y=X
40 PRINT AT Y,X:
50 PAUSE 0
60 LET A=INKEY$
70 IF A="<" THEN GO TO 1000
80 IF A=">" THEN GO TO 1100
90 IF A="=" THEN GO TO 1200
95 IF (A=" ") AND (X="Z") THEN
EN GO TO 110
100 GO TO 110
110 PRINT AT Y,X:A$
120 LET X=X+1
130 IF X=32 THEN LET X=0: LET Y
=Y+1
140 IF Y=22 THEN GO TO 1210
150 GO TO 40
1000 REM
*****
* BORRADO
*
*****
1005 PRINT AT Y,X: " "
1010 LET X=X-1
1020 IF X=-1 THEN LET X=31: LET
Y=Y-1
1030 IF Y=-1 AND X=31 THEN LET Y
=0: LET X=0
1040 GO TO 40
1100 REM
*****
* RETORNO
*
*****
1105 PRINT AT Y,X: " "
1110 LET X=0: LET Y=Y+1
1120 GO TO 140
1200 REM
*****
* ALMACENAR
*
*****

```

```

1205 PRINT AT Y,X: " "
1210 PRINT #0,AT 1,1,"L-Lista /
G-Graba / C-Continúa"
1220 IF INKEY$="" THEN GO TO 122
1230 IF INKEY$="L" OR INKEY$="L"
THEN GO TO 1260
1240 IF INKEY$="G" OR INKEY$="G"
THEN GO TO 1350
1245 IF INKEY$="C" OR INKEY$="C"
THEN GO TO 1500
1250 GO TO 1220
1260 REM
*****
* IMPRESORA
*
*****

```

```

1262 INPUT #0,AT 0,2:"Conecte la
impresora y pulse"
1270 PRINT #0,AT 1,10:"una tecla"
1280 PAUSE 0
1285 INPUT #0
1290 COPY
1300 PRINT "Impresión
terminada": PAUSE 100
1310 GO TO 1210
1350 REM
*****
* GRABACION
*
*****

```

```

1360 INPUT "Nombre (max 10 carac
teres) para el archivo n°
1370 PRINT #0,AT 1,5:"Grabacion
terminada": PAUSE 100
1390 GO TO 1210
1500 REM
*****
* CONTINUACION?
*
*****

```

```

1510 INPUT #0
1520 PRINT #0,AT 1,0:"Quiere edi
tar otra página (S/N)"
1530 IF INKEY$="S" OR INKEY$="S"
THEN GO TO 10
1540 IF INKEY$="N" OR INKEY$="N"
THEN CLS: STOP
1550 GO TO 1530

```

Tenemos la posibilidad de elegir las teclas que vamos a utilizar como cursores. El programa tiene asignadas por defecto, es decir, si no se eligen otras, las siguientes:

- 7 - ARRIBA
- 6 - ABAJO
- 8 - DERECHA
- 9 - IZQUIERDA

La velocidad es otro de los parámetros que se puede elegir, su valor varía entre «1» (rápida) y «9» (lenta).

Aparte de las teclas utilizadas como cursor, existen otras dos que en combinación con SYMBOL SHIFT realizan una determinada función.

SYMBOL SHIFT + B, borra la pantalla.
SYMBOL SHIFT + C, fin del programa.
La estructura del programa es la siguiente:

- 10 : Comentario con el nombre del programa.
- 20 : Asignación del cursor rojo para el borde, verde para el fondo y negro para los caracteres.
- 30 : Llamada a la subrutina que presenta el menú con los cursores.
- 40-90 : Verificación de la opción elegida.
- 100 : Asignación de los valores por defecto.

Asignación del cursor rojo para el borde, verde para el fondo y negro para los caracteres.
Llamada a la subrutina que presenta el menú con los cursores.
Verificación de la opción elegida.
Asignación de los valores por defecto.

11

PRAMA 2

PROG

($\sqrt{-144}$) se visualizará el mensaje de error:

A Invalid argument

Ejemplo:

— PRINT SQR —144

Cuando se evalúa una variable, ésta puede tomar un valor negativo, para asegurarnos que al calcular su raíz no nos dé error, podemos utilizar la función «ABS».

```
10 LET raíz = -144
20 PRINT SQR (ABS raíz)
```

Aunque el Spectrum sólo tiene una función de *radicación* (SQR), cuyo índice es 2, se pueden obtener raíces de cualquier orden, para ello hay que basarse en la siguiente igualdad:

$$\sqrt[n]{r} = r^{\frac{1}{n}}$$

donde «r» es el radicando e «i» el índice.

Ejemplo:

— Raíz cúbica de 27
PRINT 27 \uparrow (1/3)
— Raíz quinta de 32
PRINT 32 \uparrow (1/5)

PI

Acceso al teclado

PI



Definición

«PI» es el nombre de una letra griega (π) usada como constante en multitud de cálculos matemáticos. Su valor aproximado es:

$$\pi \simeq 3,14159265...$$

Ejemplos:

a) Cálculo de la longitud de una circunferencia de radio 6.

```
10 LET radio = 6
20 PRINT 2 * PI * radio
```

b) Cálculo de la impedancia que presenta una bobina de 0,5 Henrys a la frecuencia de 12 KHz.

```
10 LET L = 0,5
20 LET F = 12000
```

BIN

Acceso al teclado

BIN



Definición

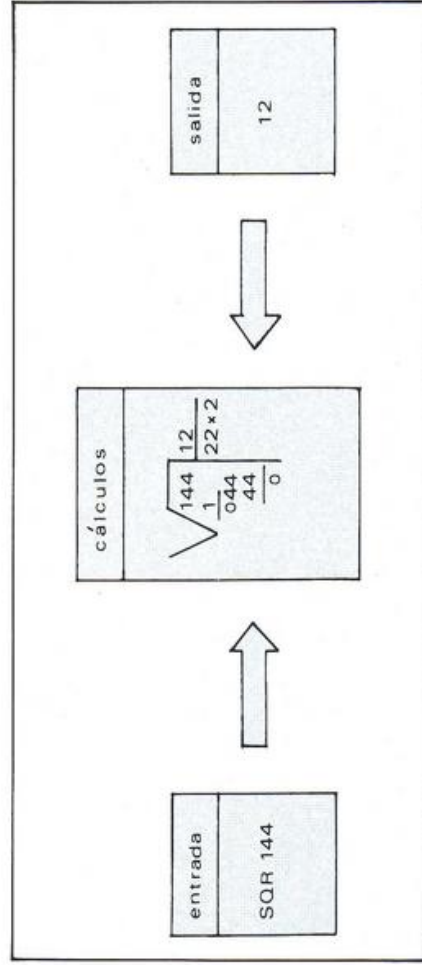
Permite la representación de los números en notación binaria, para una mayor aclaración conviene consultar el capítulo «CONSTANTES Y VARIABLES» (Pág. 30).

Ejemplos:

```
— LET a = BIN 1001
— PRINT —BIN 11011*2
— LET c = BIN 110/na
— PRINT SQR BIN 1111
```

El mayor número que se puede representar en este tipo de notación, es «65535».

En la página 25 se presenta el programa «CODEBIN» que realiza la transformación inversa, es decir, pasar de notación decimal a binaria.



Ejemplo «Función».

150 MICROBASIC

```
*****
* BORRADO *
*****
1610 FOR n=1 TO 20
1620 PRINT AT n,1;"
1630 NEXT n
1640 RETURN
```

```
1510 IF POSX<1 THEN LET POSX=1
1520 IF POSY>30 THEN LET POSY=30
1530 IF POSY<1 THEN LET POSY=1
1540 IF POSY>20 THEN LET POSY=20
1550 PRINT AT POSY,POSX;"*
1560 FOR n=1 TO velocidad
1570 NEXT n
1580 REM
1600 REM
```

110 : Retardo de un segundo aproximadamente. Llamada a la subrutina que pregunta la velocidad.

120 : Borrado de pantalla.

130-210 : Rutina que dibuja, con ayuda de los gráficos predefinidos, un recuadro.

230-240 : Inicialización de las coordenadas del asterisco.

250 : Llamada a la subrutina que dibuja el asterisco.

260-300 : Determinación de la tecla pulsada, cálculo de la nueva posición y llamada a la subrutina de visualización.

310 : Si la tecla pulsada es el asterisco (*), la opción de borrar es la elegida.

320 : Si la tecla pulsada es la interrogación (?), el programa termina su ejecución.

340 : Llamada a la subrutina de borrado.

1000-1070 : Subrutina que visualiza el menú con la asignación de cursores inicial.

1100-1220 : Subrutina utilizada para modificar la asignación inicial de cursores.

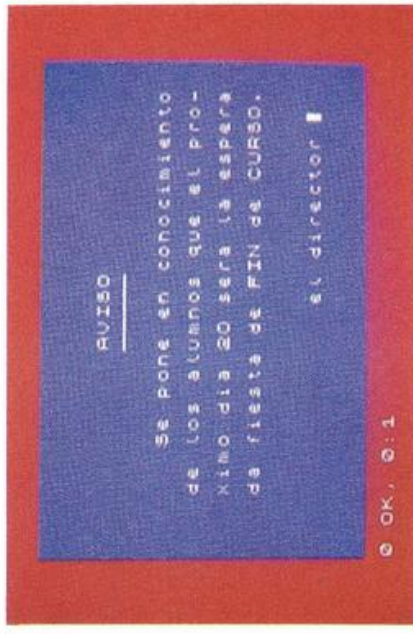
1300-1410 : Subrutina para la

introducción de la velocidad.

1500-1580 : Verificación de que las coordenadas del asterisco se encuentran dentro del recuadro, visualización

del mismo y temporización variable, dependiente de la velocidad elegida.
1600-1640 : Borrado de la parte interior del recuadro.

Programa «Movimiento».



Programa «Máquina».

MICROBASIC 147

FUNCIONES

Existen una serie de funciones definidas dentro del BASIC, que pueden clasificarse en:

- Numéricas.
- De cadena.
- Aparte de estas funciones, ya fijas, el usuario puede definir las suyas propias.

En general, una función proporciona un resultado después de haber efectuado unos cálculos con un dato denominado *parámetro*; éste forma parte del argumento de la instrucción. Para cada valor, la función retorna un resultado distinto.

Aunque no es necesario incluir los parámetros de la función entre paréntesis, es conveniente hacerlo, en algunos casos, por motivos de claridad. También hay que tener en cuenta que las funciones tienen mayor prioridad que las operaciones. Las funciones no se utilizan directamente como comandos, por el contrario necesitan ir acompañadas de las palabras clave «PRINT» (visualización), «LET» (asignación), «IF... THEN», etc.

Funciones numéricas

Estas pueden clasificarse en los siguientes grupos:

- Matemáticas.
- Trigonométricas.
- Exponenciales.
- Logarítmicas.
- Aleatorias.
- Definidas.

```
30 IF ABS NUMERO > 99999999 THEN
  GOTO 20
40 IF "NUMERO" THEN LET ABS "SI"
50 IF "NUMERO" THEN LET ABS "PO"
60 LET ABS "NUMERO"
70 LET ABS "NUMERO"
80 LET ABS "NUMERO"
90 LET ABS "NUMERO"
100 LET ABS "NUMERO"
110 LET ABS "NUMERO"
120 GOTO 20
```

ABS

Aceso al teclado

ABS

G THEN
GOTO

MODO E

Definición

Esta función retorna el valor absoluto del argumento.

Ejemplos:

- LET a = ABS 30.
- PRINT ABS (-100 * 2)
- IF ABS valor < > 30 THEN...
- FOR n = 1 TO ABS X

El valor absoluto de una expresión se calcula despreciando su signo, por tanto, el resultado de las siguientes instrucciones será el mismo.

```
PRINT ABS 3542
PRINT ABS -3542
PRINT ABS +3542
```

En el siguiente programa se visualiza el valor absoluto de cualquier número comprendido entre 99999999 y -99999999.

```
10 REM *****
20 INPUT "NUMERO >>>"; NUMERO
30 IF ABS NUMERO > 99999999 OR
  ABS NUMERO < -99999999 THEN
  LET ABS "NUMERO"
  GOTO 20
40 LET ABS "NUMERO"
50 LET ABS "NUMERO"
60 LET ABS "NUMERO"
70 LET ABS "NUMERO"
80 LET ABS "NUMERO"
90 LET ABS "NUMERO"
100 GOTO 20
```

Definición

La función «INT» retorna el valor entero de una expresión:

Ejemplos:

- LET X = INT - Y
- PRINT INT (X + 3)
- IF INT n = n THEN ...
- FOR Z = INT t TO INT j

«INT» redondea por defecto el argumento, despreciando sus decimales. Redondear por defecto significa asumir el valor entero *inmediato inferior*, por tanto, se obtendrá el mismo resultado con cualquiera de las dos sentencias siguientes:

```
PRINT INT 3.000001
PRINT INT 3.999999
```

en ambos casos el resultado es «3».

Con los números negativos ocurre una cosa curiosa, ya que al efectuar el redondeo por defecto, aumenta su valor absoluto:

Ejemplo:

```
PRINT INT -3.000001
PRINT INT -3.999999
```

el resultado de ambas funciones es «-4».

El siguiente programa calcula si el número entero introducido por el teclado es *par* o *impar*.

```
10 REM *****
20 INPUT "NUMERO >>>"; NUMERO
30 IF ABS NUMERO > 99999999 OR
  ABS NUMERO < -99999999 THEN
  LET ABS "NUMERO"
  GOTO 20
40 LET ABS "NUMERO"
50 LET ABS "NUMERO"
60 LET ABS "NUMERO"
70 LET ABS "NUMERO"
80 LET ABS "NUMERO"
90 LET ABS "NUMERO"
100 GOTO 20
```

Con ayuda de la función «INT» se averigua en la línea 30 si la variable «numero» tiene parte fraccionaria.

En el siguiente ejemplo se visualiza la parte entera y fraccionaria del número positivo que se introduce por teclado.

Las pequeñas diferencias que pueden existir, entre las partes fraccionarias son debidas a los cálculos.

```
10 REM *****
20 INPUT "NUMERO >>>"; NUMERO
30 IF ABS NUMERO > 99999999 OR
  ABS NUMERO < -99999999 THEN
  LET ABS "NUMERO"
  GOTO 20
40 LET ABS "NUMERO"
50 LET ABS "NUMERO"
60 LET ABS "NUMERO"
70 LET ABS "NUMERO"
80 LET ABS "NUMERO"
90 LET ABS "NUMERO"
100 GOTO 20
```

SGN

Aceso al teclado

SGN

F TO
FOR

MODO E

Definición

La función «SGN» nos indica qué signo tiene la expresión que estamos evaluando. Los posibles valores que puede retornar son:

```
1 : Si es positiva.
-1 : Si es negativa.
0 : Si es 0
```

Ejemplos:

- PRINT SGN -30
- LET a = SGN C
- PRINT SGN (7 * (-5))
- LET X = 7 * SGN Z

El siguiente programa nos indica si el número introducido por teclado es positivo o negativo.

```
10 REM *****
20 INPUT "NUMERO >>>"; NUMERO
30 IF ABS NUMERO > 99999999 OR
  ABS NUMERO < -99999999 THEN
  LET ABS "NUMERO"
  GOTO 20
40 LET ABS "NUMERO"
50 LET ABS "NUMERO"
60 LET ABS "NUMERO"
70 LET ABS "NUMERO"
80 LET ABS "NUMERO"
90 LET ABS "NUMERO"
100 GOTO 20
```

SQR

Aceso al teclado

SQR

H
GOSUB

MODO E

CIRCLE

Definición

La función «SQR» calcula la raíz cuadrada del argumento.

Ejemplos:

- PRINT SQR 144
- LET r = SQR 625 + 13
- PRINT SQR raíz
- LET n = SQR (25 + a)

«SQR» sólo calcula raíces de tipo *real*, si se pretende calcular una *imaginaria*

Definición de la función:

```
10 DEF FN c (r, g) = PI * r ^ 2 / 360
  * g
```

Un ejemplo de utilización podría ser el siguiente:

```
20 INPUT "Radio: ", radio
30 INPUT "grados: ", grados
40 PRINT FN c (radio, grados)
```

Errores

Hay una serie de mensajes típicos de error que se visualizan al manejar incorrectamente las funciones.

a) Al intentar manejar una función no definida previamente aparece el mensaje:

P FN without DEF

Ejemplo:

```
10 REM *****
  *****
  ERROR "P" *****
  *****
20 INPUT "temperatura? ", calor
30 LET x=FN a(calor)
```

b) Cuando se hace una llamada a una función definida, y no coincide el número de parámetros aparece el mensaje:

Q Parameter error

Este mensaje también aparece cuando alguno de los parámetros no es del mismo tipo que los definidos en la función; es decir, un parámetro de cadena en

una función numérica o viceversa.
Ejemplo:

```
10 REM *****
  *****
  ERROR "Q" *****
  *****
20 DEF FN a(x,y)=x+y/2
30 INPUT "Numero 1 ", n1
40 INPUT "Numero 2 ", n2
50 PRINT FN a(n1)
```

en este caso falta un parámetro en la llamada de la función. También daría error las siguientes instrucciones:

```
50 PRINT FN r (n1, n2, 10)
```

(sobra un parámetro)

```
50 PRINT FN r (n1, "pepe")
```

(parámetro de cadena)

Radianes

Aplicación de la trigonometría

Los programas que a continuación se presentan, son una pequeña muestra de las múltiples aplicaciones de la trigonometría.

a) Calcular la altura que alcanza una escalera apoyada en una pared, conociendo su longitud y el ángulo que forma con el suelo.

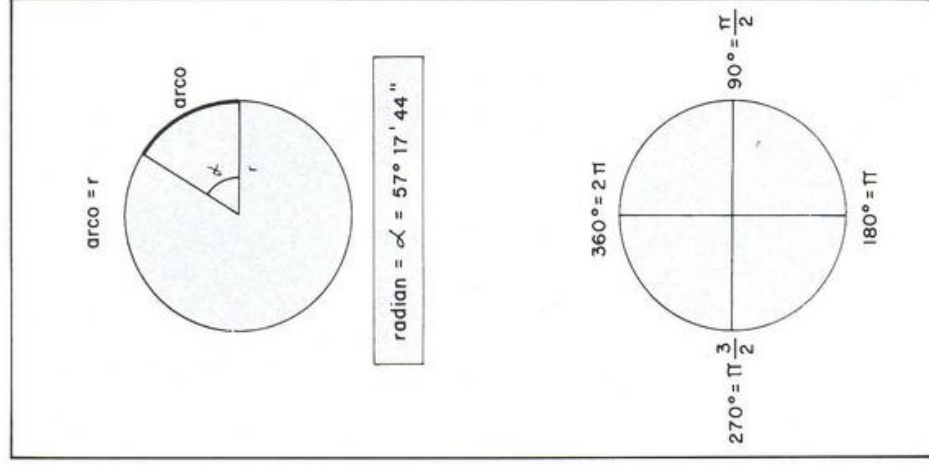
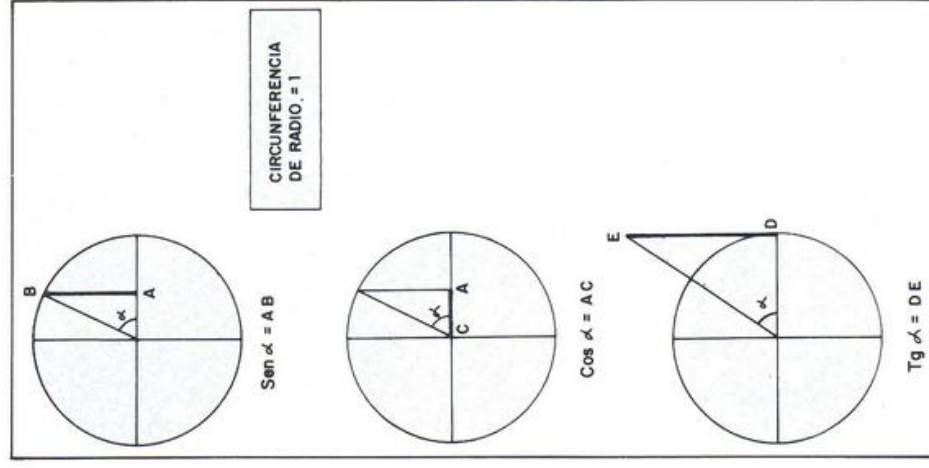
```
10 REM *****
  *****
  ESCALERA *****
  *****
```

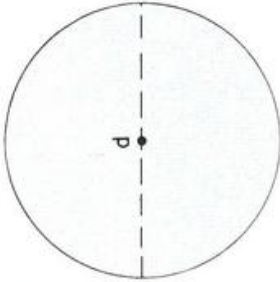
b) Calcular la altura de una torre, conociendo la distancia que nos separa de ella y la visual hasta su parte más alta.

```
10 BORDER 4: PAPER 4: INK 0: C
20 INPUT "Longitud escalera >>>"
30 CLS
40 IF longitud<1 OR longitud>3
50 THEN PRINT "Longitud ..... Lon"
60 INPUT "Inclinación (grados) >>>"
70 IF inclinacion<1 OR inclinacion>90
80 THEN PRINT "Inclinación (grados) >>>"
90 LET altura=longitud*SIN(rad
100 PRINT "Altura ..... Alt"
110 GOTO 30
120 REM *****
130 RETURN
```

Funciones trigonométricas

```
10 REM *****
  *****
  TORRE *****
  *****
20 BORDER 4: PAPER 4: INK 0: CLS
30 INPUT "Distancia base >>>"
40 IF distancia<1 OR distancia>500
50 THEN PRINT "Distancia base >>>"
60 INPUT "Visual (grados) >>>"
70 IF grados<1 OR grados>90
80 THEN PRINT "Visual (grados) >>>"
90 LET altura=distancia*TAN(rad
100 PRINT "Altura ..... Alt"
110 GOTO 30
120 REM *****
130 RETURN
```



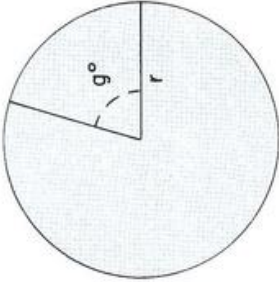


AREA DEL CIRCULO

$$S = \pi \left(\frac{d}{2} \right)^2$$

↓

$$\text{DEF FN } S(d) = \pi * (d/2.) \uparrow 2$$



AREA DEL SECTOR CIRCULAR

$$S = \frac{\pi r^2}{360} \cdot g^\circ$$

↓

$$\text{DEF FN } c(r, g) = \pi * r \uparrow 2 / 360 * g$$

exponente unidad da como resultado el propio número:

$$n^1 = n$$

Ejemplo:

PRINT 27 ↑ 1

Función logarítmica

Es la inversa de la función exponencial.

Ejemplo:

$$y = \log_a x$$

La expresión anterior se lee de la siguiente forma: «y» es igual al logaritmo de «x» en base «a»; donde «a» es la base de los logaritmos.

Se denomina logaritmo de un número al exponente a que es preciso elevar la base para obtener dicho número.

$$\log_a x = y$$

$$a^y = x$$

Las bases de los logaritmos más utilizadas son la *decimal* y la *neperiana*. Los logaritmos de base decimal son también conocidos como logaritmos *vulgares* o de *Briggs*.

Ejemplo:

$$y = \log_{10} x$$

Los logaritmos neperianos deben su nombre a John Neper (matemático inglés) y tienen como base el conocido número «e». También son llamados logaritmos *naturales*.

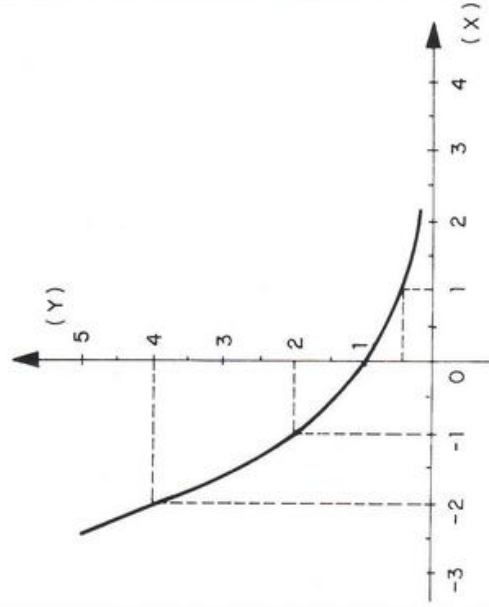
Ejemplo:

$$y = \log_e x$$

abreviadamente, puede escribirse el logaritmo neperiano como «ln».

Ejemplo:

$$y = \ln x$$



$$Y = \left(\frac{1}{2} \right)^X$$

Función exponencial decreciente

La función logarítmica incluida en el juego de senfencias del Spectrum, es la de base neperiana.

Ejemplos:

- LET a = LN 7
- PRINT LN suma
- LET b = 2 * LN k
- PRINT LN (17/n)

Cuando el argumento es igual o inferior a «0», se visualiza el mensaje:

A Invalid argument

La utilización de los logaritmos es de uso frecuente en calculadoras de las denominadas científicas, y cómo no, en el Spectrum, ya que basándose en dos de los teoremas de dicha función se realizan los cálculos internos de potenciación y radicación.

a) El logaritmo de una potencia es igual al producto del exponente por el logaritmo de la base.

Acceso al teclado

LN

LN



MODO E

BEEP

Definición

La función «LN» retorna el logaritmo neperiano del argumento.

Ejemplos:

a) Calcular el logaritmo en base 10 de 4 ($\log_{10} 4$).

PRINT LN (4)/LN (10)

b) Calcular el logaritmo de 15 en base 2 ($\log_2 15$).

PRINT LN (15)/LN 2

$$\log_a x = \frac{\ln x}{\ln a}$$

que dice, que el logaritmo de un número «x» en base «a» es igual al cociente entre el logaritmo neperiano del número y el logaritmo neperiano de la base.

Definición de funciones

Compare los resultados anteriores con los proporcionados por «144 ↑ (1/2)».

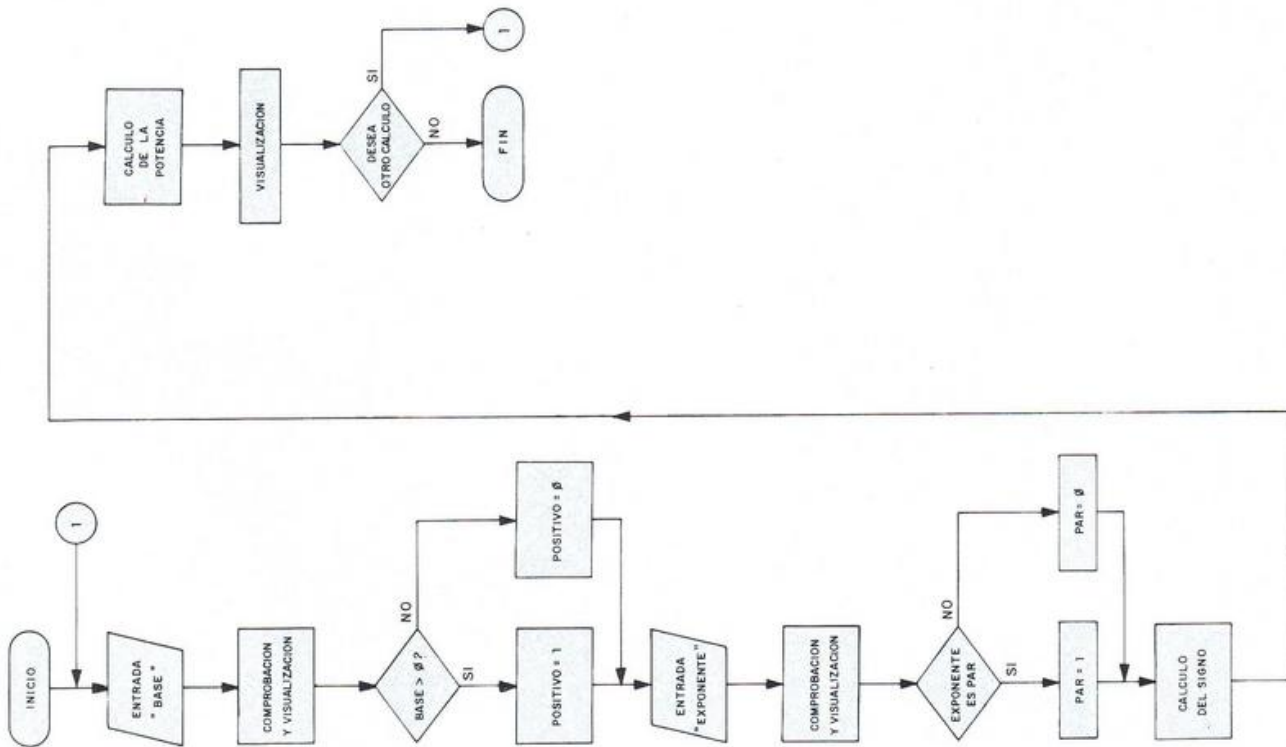
Aunque el Spectrum solamente calcula, de forma directa, los logaritmos de base neperiana, pueden calcularse en cualquier otra base, teniendo en cuenta la siguiente igualdad:

PROGRAMA 1

```

10 REM *****
* POTENCIACION *
*****
20 INPUT "Base >>> "; base
30 IF (ABS base > 999) OR (base =
0) THEN GO TO 20
40 PRINT "Base .....", base
50 IF base > 0 THEN LET positivo
=1; GO TO 70
60 LET positivo=0
70 INPUT "Exponente >>> "; expo
nente
80 IF ABS exponente > 10 OR INT
exponente <> exponente THEN GO TO
70
90 PRINT "Exponente ....", exp
onente
100 LET divisor=exponente/2
110 LET entero=INT divisor
120 IF divisor=entero THEN LET
par=1; GO TO 140
130 LET par=0
140 IF (positivo=0 AND par=1) 0
R (positivo=1) THEN LET s$="+";
GO TO 160
150 LET s$="-"
160 LET resultado=EXP (exponent
e*LN (ABS base))
170 PRINT base; " elevado a "; ex
ponente; " es: "; s$; resultado
180 PRINT #0; "Desea otro calcul
o (S/N)";
200 IF INKEY$="" THEN GO TO 200
210 LET a$=INKEY$
220 IF a$="S" OR a$="s" THEN CL
S: GO TO 10
230 IF a$="N" OR a$="n" THEN CL
S: STOP TO 210

```



Programa «potencia».

no multiplica «13» veces el número «2», sino que por el contrario, lo calcula de acuerdo con el teorema expuesto anteriormente.

Introduzca:

PRINT EXP (13 * LN 2)

realice la siguiente operación:

PRINT 2 ^ 13 - (EXP (13 * LN 2))

en este caso el resultado sí es cero.

Al realizar la operación $2 \wedge 13$ no se visualiza «8192.000022888184» ya que, debido al sistema de presentación del Spectrum, este valor queda redondeado a «8192».

b) El otro teorema dice que el logaritmo de una raíz es igual al cociente entre el logaritmo del radicando y el índice de la raíz.

$$\ln \sqrt[b]{a} = \ln(a)/b$$

luego

$$\sqrt[b]{a} = \text{exponencial} (\ln(a)/b)$$

Al igual que en el caso anterior, la radicación también se realiza con ayuda de los logaritmos.

Realice los siguientes pasos:

PRINT SQR 144
PRINT SQR 144 - 12
PRINT EXP (LN 144/2)
PRINT SQR 144 - (EXP (LN 144/2))

Las explicaciones dadas en el caso de la potenciación también son válidas para la potenciación.

PRINT 2 ^ 13

observará que el resultado es «8192»; si restamos ambos valores, el resultado debería ser «0».

PRINT 2 ^ 13 - 8192

¿Por qué razón no es «0»? es- to se debe a que el ordenador

$$\ln a^b = b \cdot \ln a$$

por tanto

$$a^b = \text{exponencial} (b \cdot \ln a)$$

Para comprobar que el ordenador realiza la potenciación con ayuda de los logaritmos, introduzca el siguiente comando directo:

se presenta un menú con tres opciones:

```

1 - REPASAR
2 - EXAMEN
3 - FIN

```

La opción «1» permite repasar la tabla de multiplicar del número que elijamos, ya que se nos visualiza ésta en la pantalla.

La opción «EXAMEN» tiene otras dos opciones:

```

1 - PARCIAL
2 - GENERAL

```

Eligiendo la primera, el ordenador nos pregunta la tabla de multiplicar de un número determinado, en cambio con la segunda, las preguntas son sobre cualquier número.

La opción «3» permite parar la ejecución del programa. La estructura general es:

```

10 : Comentario con el nombre del programa.
20 : Asignación del color verde para fondo y borde, y negro para los caracteres.
100 : Entrada del nombre.
110-132 : Presentación del menú.
140-170 : Detección de la opción elegida.
1000 : Comienzo de la opción (1) (REPASAR).
1002 : Borde de color azul.
1010-1020 : Entrada y comprobación del número de tabla.

```

```

10 REM *****
15 PRINT "Randomize" "Seed"
20 INPUT "Numero" ; N
30 RANDOMIZE N
40 PRINT "Seed" ; PEEK 23670+PEEK 23671+255
50 GO TO 20

10 REM *****
15 PRINT "Randomize" "Seed"
20 INPUT "Numero" ; N
30 RANDOMIZE N
40 PRINT "Seed" ; PEEK 23670+PEEK 23671+255
50 GO TO 20

```

Puede observar que, en ocasiones, existe una diferencia entre ambas, de una unidad debido al retardo que hay entre la ejecución de «RANDOMIZE» y la visualización del contenido de la variable «FRAMES»; recuerde que «FRAMES» se incrementa según transcurre el tiempo y «SEED» se actualiza a la ejecución de «RANDOMIZE».

El programa número «2» pertenece al grupo denominado «UTILIDADES» y tiene aplicación en el campo de la enseñanza, ya que permite a los principiantes estudiantes de la EGB repasar la tabla de multiplicar o contestar las preguntas que el ordenador realice sobre el tema.

Lo primero que hace el ordenador es preguntarnos nuestro nombre, ya que en diversas ocasiones hace referencia a él. Posteriormente

```

1022-1080 : Visualización de la tabla.
1090-1140 : Selección de una nueva tabla, o salto al menú principal.
2000 : Comienzo de la opción (2).
2010 : Borde de color rojo.
2022-2050 : Presentación del menú secundario.
2060-2100 : Detección de la opción elegida.
2200 : Comienzo de las preguntas.
2212 : Inicialización de la secuencia de números aleatorios, ésta depende del tiempo que lleve conectado el ordenador.
2270-2250 : Visualización de rólulos e inicialización de variables.
2260-2360 : Generación aleatoria de la pregunta, dependiendo de la opción elegida.
2400 : Cálculo del resultado.
2405-2420 : Visualización de la pregunta y entrada de la respuesta.
2440 : Visualización de la respuesta.
2450 : Comienzo del análisis.
2460 : Comparación de los resultados.
2480-2490 : Mensaje de acierto.
2510-2520 : Mensaje de fallo.
2530-2560 : Visualización del número de aciertos y porcentaje.
2570-2620 : Selección de otra pregunta o salto al menú principal.
2630 : Incremento de la variable (pregunta).
2640-2660 : Borrado del mensaje.
2670-2680 : Selección de la línea de salto para una nueva pregunta.

```

FUNCION ALEATORIA

Como su propio nombre indica, una función aleatoria es aquella que retorna un valor al azar, es decir, aquel que no sigue ninguna ley o algoritmo; en realidad esto no es del todo cierto ya que es bastante difícil, por no decir imposible, implementar una función de este tipo en un ordenador, por tanto, éstos incluyen una función llamada *pseudoaleatoria*.

La función pseudoaleatoria, siguiendo un algoritmo, elige adecuadamente ciertos números para que parezcan aleatorios. En el caso del Spectrum existe una secuencia cíclica formada por «65536» números distintos.

Las funciones aleatorias tienen un extenso campo de aplicación en los juegos y en los programas didácticos.

RND

Acceso al teclado

RND



Definición

«RND» retorna uno de los

POS.	NUM.
1	.0022735596
2	.0034179688
3	.0045623779
4	.0057067871
5	.0068511963
6	.0079956055
7	.0091400146
8	.010284424

INDICE

6	SEED
---	------

65533	0.99656677
65534	0.99771118
65535	0.99855559

Variable «SEED».

«65536» números que forman la secuencia de números aleatorios. Al ser una función, debe ir acompañada de sentencias tales como «PRINT», «LET», «IF...» etc.

Ejemplos:

```

- LET a = RND
- LET b = INT (RND + 5)
- PRINT RND
- PRINT 3 + RND

```

Ejecute el siguiente programa:

observará que todos los números son inferiores a «1», ya que el valor retornado por «RND» está comprendido entre «0» y «1», en alguna ocasión puede valer «0», pero nunca «1».

Para conseguir números aleatorios comprendidos entre otros rangos, por ejemplo,

Existe un pequeño algoritmo para obtener números aleatorios enteros comprendidos entre dos cualesquiera, ambos inclusive:

```
INT ((Y-X+1) * RND) + X
```

donde «X» es el menor e «Y» el mayor, veamos unos ejemplos:

```
10 LET Y = INT (22 * RND)
20 LET X = INT (32 * RND)
30 PRINT AT Y, X: ( )
40 GOTO 10
```

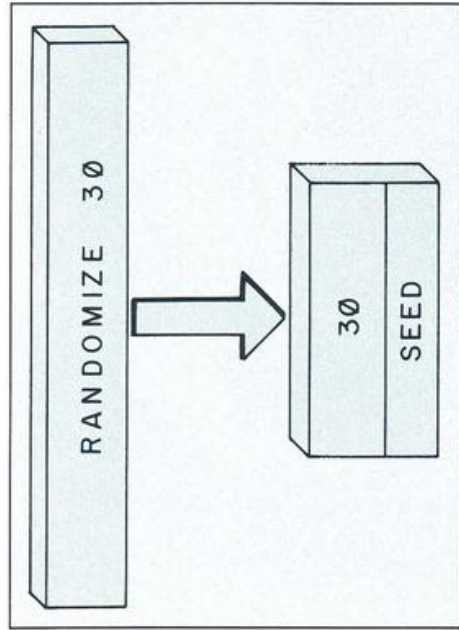
Este programa visualiza un asterisco (*) en una posición aleatoria de la pantalla, si tiene paciencia, al cabo de un rato, verá la pantalla llena de este símbolo, en total «704».

Una variante de este programa podría consistir en introducir estas tres líneas suplementarias:

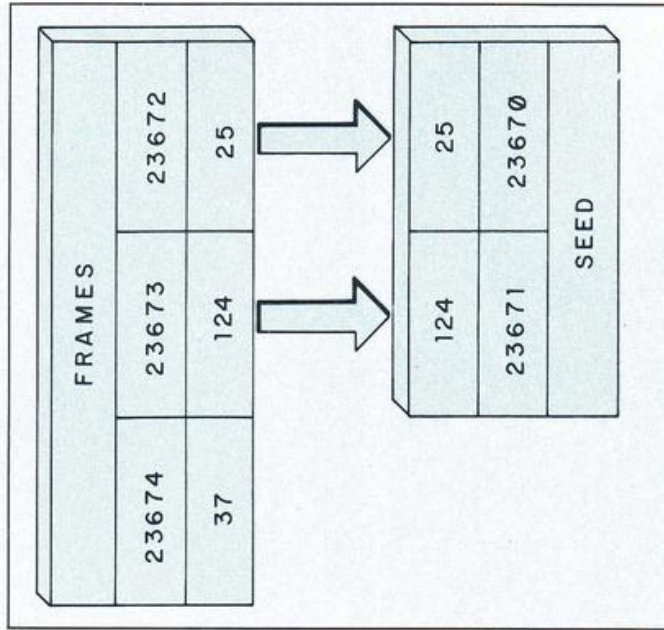
```
5 INPUT (tiempo >); X
32 PAUSE X
34 PRINT AT Y, X: ( )
```

la variable «X» controla el tiempo de visualización del asterisco. Introduzca, por ejemplo, el valor «5» y parecerá que hay una mosca revoloteando sobre la pantalla de su televisor.

Dentro del Sistema Operativo del Spectrum está implementado un algoritmo que calcula sucesivamente cual de los «65536» números aleatorios será el siguiente en ser presentado. El resultado de este algoritmo se almacena en dos posiciones consecutivas de memoria, éstas forman una de las variables del sistema conocida por el nombre de «SEED»; el contenido de esta variable es, por



Randomize «T».



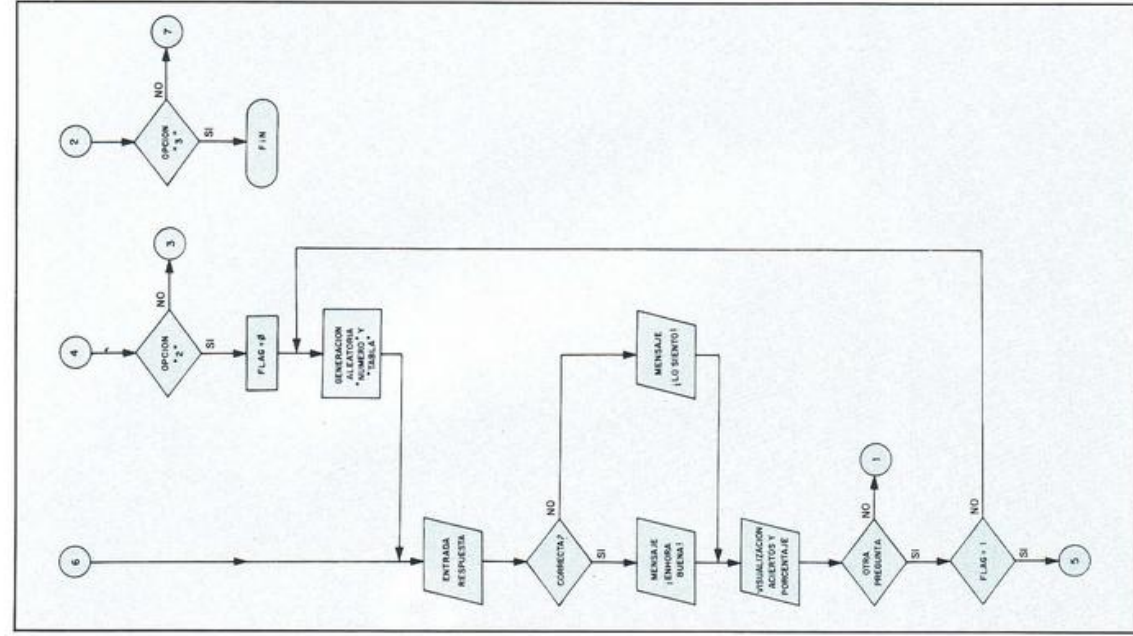
Randomize «0».

de «0» a «10», podríamos utilizar:

```
10 FOR n = 1 TO 44
20 PRINT RND * 10
30 NEXT n
```

pero si lo que se desea es obtener números aleatorios enteros, utilizaríamos:

```
10 FOR n = 1 TO 44
20 PRINT INT (RND * 10).
30 NEXT n
```



Programa «TABLA»

b) Cuando su argumento es cero.

Cada vez que se conecta el ordenador, se ejecuta una sentencia «NEW» o se activa el «RESET», la generación de números aleatorios se realiza siempre a partir del mismo valor.

```
10 REM *****
11 VARIABLE "FRAMES" *****
20 PRINT "Frames", "Seed", ...
30 PRINT *****
40 RANDOMIZE (23673, 25)
50 PRINT *****
60 GOTO 10
```

Edite el siguiente programa después de introducir «NEW» como comando directo:

```
10 PRINT RND
20 GOTO 10
```

anote alguno de los valores obtenidos. Vuelva a introducir «NEW» y edite otra vez el programa. Comprobará que los resultados son idénticos.

Para conseguir una función más aleatoria debe introducirse previamente:

```
RANDOMIZE 0
```

o en su defecto,

```
RANDOMIZE
```

Ejemplo:

```
10 RANDOMIZE
20 PRINT RND
30 GO TO 20
```

En este caso el valor retornado por «RND» está en función del tiempo que lleva conectado el ordenador. Existe otra variable del sistema conocida por el nombre de «FRAMES» que almacena indirectamente este tipo. La variable «FRAMES» ocupa tres posiciones de memoria, veamos un ejemplo de lectura de dicha variable:

```
10 PRINT
PEEK (23674+65536)+
PEEK (23673+256)+
PEEK 23672
20 GOTO 10
```

Cuando se ejecuta «RANDOMIZE 0» la variable «SEED» asume el valor de las dos posiciones de memoria menos significativas de «FRAMES». Veamos un ejemplo:


```

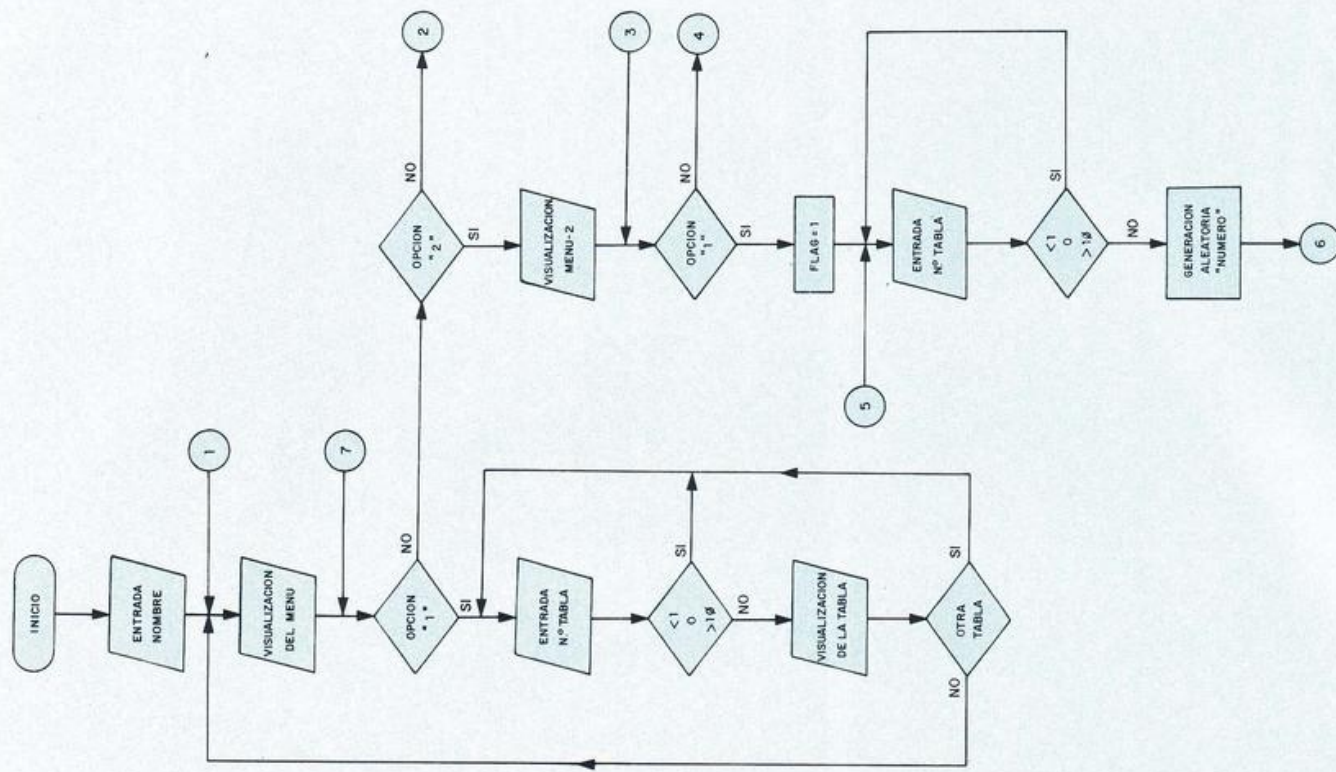
10 REM
*****
* CURSO BASIC *
*****
* LA TABLA *
*****
L20 BORDER 4: PAPER 4: INK 0: C
LS 99 REM
*****
* MENU *
*****
100 INPUT "Como te llamas >>> "
INKEY$=INKEY$
110 PRINT AT 1,1,$;" elige una
de las tres:"
120 PRINT AT 9,8,1;" REPARAR"
130 PRINT AT 12,8,2;" EXAMEN"
132 PRINT AT 15,8,3;" FIN"
140 IF INKEY$="1" THEN GO TO 140
150 IF INKEY$="2" THEN GO TO 150
160 IF INKEY$="3" THEN GO TO 160
162 IF INKEY$="3" THEN CLS : ST
OP
170 GO TO 140
1000 REM
*****
* REPARAR *
*****
1002 BORDER 1
1004 CLS
11010 INPUT "Que tabla deseas rep
asar >>> ";tabla
1020 IF tabla<1 OR tabla>10 THEN
1030 PRINT AT 1,2;"TABLA DE MULT
IPLES DEL " ,tabla,""
1038 LET n=1 TO 10
1040 LET resultado=tabla*n
1050 PRINT AT n+5,10;tabla," x "
n,
1060 IF n>10 THEN PRINT " ";
1070 PRINT "=" ;resultado
1080 NEXT n
1090 PRINT
1100 IF INKEY$="" THEN GO TO 110
1110 LET b$=INKEY$
1120 IF b$="S" OR b$="s" THEN GO
TO 1004
1130 IF b$="N" OR b$="n" THEN BO
RDER 4: CLS : GO TO 110
1140 GO TO 1100
2000 REM
*****
* MENU 2 *
*****
2010 BORDER 2
2020 CLS
2030 INPUT "n=1 TO 25: NEXT n
2032 PRINT AT 1,1,$;" elige el
2040
2050 PRINT AT 9,8,1;" PARCIAL"
2050 PRINT AT 12,8,2;" GENERAL"
2060 IF INKEY$="" THEN GO TO 206

```

```

2070 LET b$=INKEY$
2080 IF b$="1" THEN LET flag=1:
GO TO 2200
2090 IF b$="2" THEN LET flag=0:
GO TO 2200
2100 GO TO 2070
2200 REM
*****
* PREGUNTAS *
*****
CLS
2210 RANDOMIZE
2220 PRINT AT 2,1;"PREGUNTAS "
2230 PRINT AT 4,1;"ACIERTOS "
2240 PRINT AT 6,1;"PORCENTAJE"
2250 LET pregunta=1: LET acierto
=0: LET porcentaje=0
2260 IF flag=0 THEN GO TO 2350
2270 INPUT "Que tabla deseas >
";n
2280 IF tabla<1 OR tabla>10 THEN
GO TO 2270
2290 LET numero=INT (RAND*10)+1
2300 GO TO 2400
2350 LET tabla=INT (RAND*10)+1
2360 LET numero=INT (RAND*10)+1
2370 LET resultado=tabla+numero
2400 PRINT AT 10,1;tabla," x ";n
2410 PRINT AT 10,1;resultado," =
";resultado
2420 INPUT "¿ Cuántas son ";tabla
";x ";n,numero;">";respuesta
2440 PRINT AT 10,9;respuesta
2450 REM
*****
* ANALISIS *
*****
2460 IF respuesta=resultado THEN
GO TO 2480
2470 GO TO 2510
2480 PRINT AT 15,7;"!! ENHORABU
ENA !!
2490 LET acierto=acierto+1
2500 GO TO 2530
2510 PRINT AT 15,7;"!! LO SIENT
O !!
2520 PRINT AT 19,1;a$;" ;tabla;
";numero;" son ";resultado;
";acierto
2530 LET porcentaje=INT (acierto
/pregunta*100)/100
2550 PRINT AT 6,13;"
";
"%
2570 PRINT #0;" Deseas continuar
";
2580 INPUT #0;
2590 IF b$=INKEY$
2600 LET b$="S" OR b$="s" THEN GO
TO 2630
2610 IF b$="N" OR b$="n" THEN BO
RDER 4:CLS: GO TO 110
2620 GO TO 2580
2630 LET pregunta=pregunta+1
2640 FOR n=15 TO 21
2650 PRINT AT n,0;" "
2660 NEXT n
2670 IF flag=1 THEN GO TO 2290
2680 GO TO 2350

```



tanto, la dirección dentro de la tabla de números aleatorios donde se encuentra el siguiente a presentar.

Edite las siguientes líneas de programa; al ejecutarlas, aparecerá en una columna el contenido de la variable SEED y en la otra el número aleatorio correspondiente.

La línea «50» se encarga de visualizar el contenido de dicha variable.

El programa número «1» está basado en el conocido juego de los *barquitos*, pero en esta ocasión en vez de tener que hundir una completa flota enemiga compuesta por varias unidades, tan solo será necesario acertar la posición de una lancha que está situada en una retícula de «10» por «10».

Las instrucciones del juego son sencillas, deberá introducir primero la coordenada vertical (y), posteriormente la horizontal (x). Cuando una de las coordenadas coincide aparece un mensaje de «alerta» indicando que vamos por buen camino.

La estructura general del programa es la siguiente:

```
10 : Presentación del programa.
20 : Asignación de los colores azul para el borde, verde para el fondo y negro para los caracteres.
30 : Inicialización de la variable (record).
60-140 : Bucles utilizados para dibujar la retícula. Se han utilizado las sentencias «FOR» y «DRAW», éstas serán explicadas en el capítulo dedicado al dibujo.
150-180 : Visualización de las posiciones de la retícula.
220-250 : Cálculo de la posición del barco. Se utiliza «RANDOMIZE», de esta forma, la posi-
```

ción depende del tiempo que lleve conectado el ordenador.

260 : Comienzo del juego.

270-320 : Inicialización y visualización de variables.

330-400 : Introducción del valor de las coordenadas, éste tiene que ser un número entero y estar comprendido dentro de los márgenes (0 a 9).

410 : Dibujo de la posición disparada.

425-427 : Incremento y visualización del número de intentos.

430 : Comprobación si las coordenadas han sido acertadas.

440 : Comprobación si una de las coordenadas ha sido averiguada.

450-500 : Indicación de disparo fallado.

510 : Petición de nuevas coordenadas.

520-580 : Indicación de acierto, visualización de la puntuación y actualización del record.

670-730 : ¿Continuación del juego?

RANDOMIZE

Acceso al teclado

RND



MODO K

MERGE

Tipo de sentencia

Comando de programación.

Definición

Si intenta buscar la palabra

clave «RANDOMIZE» no la encontrará, ya que en su lugar está «RAND», forma simplificada de la anterior.

Su estructura general es:

SENTENCIA	ARGUMENTO
RANDOMIZE	expresión numérica

Ejemplos:

```
- RANDOMIZE 70
- RANDOMIZE dato
- RANDOMIZE 10 * alta/3
- RANDOMIZE
```

«RANDOMIZE» sin argumento toma, por defecto, el valor «0».

El argumento debe estar comprendido entre «0» y «65535», de lo contrario aparecerá el mensaje:

B Integer out of range

Existen dos formas de utilizar «RANDOMIZE»:

a) Cuando su argumento tiene un valor comprendido entre «1» y «65535».

El argumento se utiliza para definir en qué número de la secuencia va a comenzar la generación de números aleatorios.

Ejemplo:

```
10 RANDOMIZE 50
20 FOR n = 1 TO 44
30 PRINT RND
40 NEXT n
```

Observará que cada vez que se ejecuta el programa la secuencia de número aleatorio es exactamente la misma, ya que siempre se inicializa con el valor «50».

Una forma de poder visualizar la lista completa de números aleatorios, podría ser la siguiente:

PROGRAMA 1

```
10 REM
** ***** **
** CURSO BASIC **
** ***** **
** !! AGUA !! **
** ***** **
LS 20 BORDER 1: PAPER 4: INK 0: C
30 LET record=0
60 REM
** ***** **
** DIBUJO **
** ***** **
70 FOR n=72 TO 152 STEP 8
80 PLOT 24,n
90 DRAW 80,0
100 NEXT n
110 FOR n=24 TO 104 STEP 8
120 PLOT n,72
130 DRAW 0,80
140 NEXT n
150 REM
** ***** **
** COORDENADAS **
** ***** **
160 FOR n=1 TO 12
170 PRINT AT n,1:n-3
180 NEXT n
220 REM
** ***** **
** SITUACION **
** ***** **
330 RANDOMIZE
340 barcoy=INT (10*RND)
350 LET barcox=INT (10*RND)
360 REM
** ***** **
** COMIENZO **
** ***** **
270 LET intentos=0
272 LET puntuacion=0
280 PRINT AT 3,16;"RECORD "
290 PRINT AT 5,16;"PUNTUACION "
300 PRINT AT 7,16;"INTENTOS "
310 PRINT AT 10,16;"COORD Y:"
320 PRINT AT 12,16;"COORD X:"
330 REM
** ***** **
** DISPARO **
** ***** **
10 FOR n = 1 TO 65535
20 RANDOMIZE n
30 PRINT n, RND
40 NEXT n
```

```
340 PRINT AT 10,25;" ";AT 12,25
350 INPUT "Coordenada y >>>";P
360 IF (INT posy>posy) OR (posy<0) THEN GO TO 350
370 PRINT AT 10,25;posy
380 INPUT "Coordenada x >>>";P
390 IF (INT posx>posx) OR (posx<0) OR (posx>9) THEN GO TO 370
400 PRINT AT 12,25;posx
410 PRINT AT posy+3,posx+3;" "
420 REM
** ***** **
** COMPROBACION **
** ***** **
425 LET intentos=intentos+1
427 PRINT AT 7,27;intentos
430 IF (posy=barcoy) AND (posx=barcox) THEN GO TO 520
440 IF (posy=barcoy) OR (posx=barcox) THEN PRINT "ASI:";AT 14,3;".COX:";THEN PRINT "ASY:";AT 14,3;".COY:";GO TO 460
450 ALERTA AT 14,3;"!!! AGUA !!!"
460 PRINT AT 18,10;"!!! AGUA !!!"
490 PAUSE 50
500 PRINT AT 18,10;" "
510 GO TO 340
520 REM
** ***** **
** ACIERTO **
** ***** **
530 PRINT AT 14,3;" "
540 PRINT AT 18,9;"!!! ACERTO !!!"
550 PRINT AT posy+3,posx+3;FLA
SH 1;" "
560 IF intentos=100 THEN LET puntuacion=0: GO TO 580
570 LET puntuacion=puntuacion+1
580 PRINT AT 5,27;puntuacion
590 IF puntuacion=record THEN LET record=puntuacion
600 PRINT AT 3,27;record
670 REM
** ***** **
** CONTINUACION **
** ***** **
680 PRINT AT 0,AT 1,2;"Desea jugar otra vez (S/N)";
690 LET a=INKEY$
700 IF a="S" OR a="s" THEN CL
710 GO TO 50
720 IF a="N" OR a="n" THEN CL
730 GO TO 690
```

es el número de secuencia que hace que la función «RND» retorne el valor «0», pero si no, compruébelo con:

```
10 RANDOMIZE 45438
20 PRINT RND
```

Si tiene la suficiente paciencia podrá averiguar cual

Cuando se ejecuta una sentencia del tipo «RANDOMIZE n» lo que en realidad sucede es que la variable «SEED» asume el valor del argumento; el siguiente programa lo demuestra:

FUNCIONES DE CADENA

En capítulos anteriores se explicaron las funciones matemáticas y la pseudoaleatoría; en éste, por el contrario, van a ser estudiadas las *funciones de cadena*.

Conviene, antes de leer este capítulo, si no se tiene el concepto de *cadena* lo suficientemente claro, repasar el dedicado al «Código ASCII» (pag. 37) y «operaciones con

cadena» (pag. 42), en este último se tratan temas tales como:

- CONCATENACION
- SUBCADENAS
- FRAGMENTACION
- ORDENACION

Spectrum son:

- LEN
- STR\$
- VAL
- VAL\$
- CHR\$
- CODE

también pueden definirse funciones de cadena por el usuario con la sentencia «DEF FN».

Las funciones de cadena implementadas dentro del juego de sentencias del

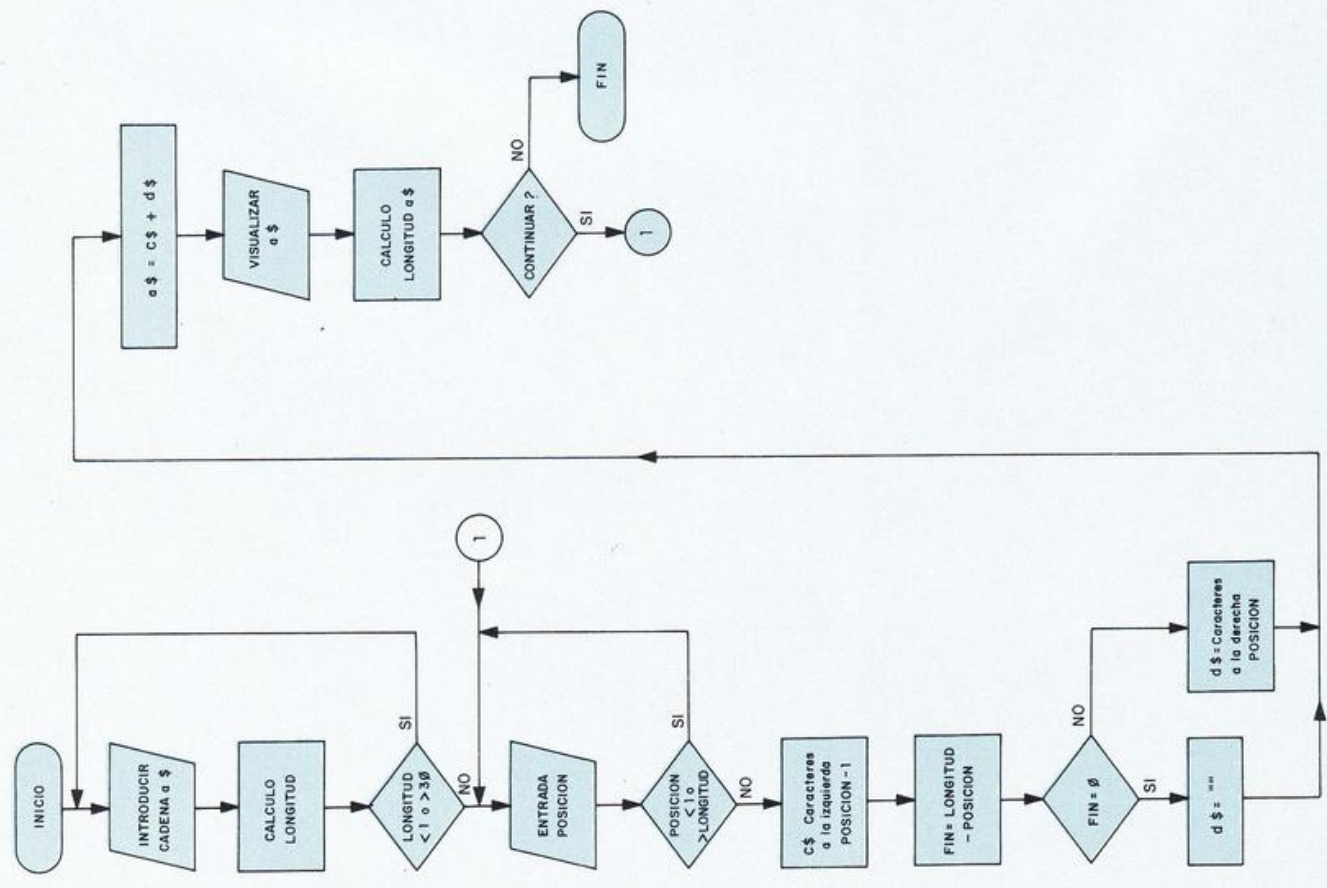
PROGRAMA 1

```
10 REM *****  
* CURSO/BASIC *  
*****  
* BUSQUEDA *  
*****  
20 PRINT "Introduce una cadena  
(max. 30)."; LINE a$  
30 INPUT " "; LINE a$  
40 LET longitud=LEN a$  
50 IF longitud<1 OR longitud>30  
  THEN GO TO 30  
60 CLS  
70 PRINT "CADENA: ";  
80 PRINT a$;  
90 PRINT "LONGITUD: "; longitud  
100 PRINT "CARACTER A BUSCAR: "  
110 INPUT " "; LINE b$  
120 IF b$="" THEN GO TO 110  
130 IF LEN b$>30 THEN GO TO 110  
140 PRINT b$;30 THEN GO TO 110  
150 LET contador=0  
160 FOR x=1 TO longitud  
170 LET c$=a$(x)  
180 IF c$=b$ THEN PRINT c$; 0  
  TO 205  
190 LET contador=contador+1  
200 PRINT "FLASH.1: c$;  
205 PAUSE 10  
210 NEXT x  
220 IF contador=0 THEN PRINT "C  
aracter no encontrado."; GO TO 2  
230 PRINT "Caracter encontrado  
"; contador; " veces."  
240 PRINT "¿Quiere buscar otr  
o caracter S/N"  
250 PAUSE 0  
260 LET i$=INKEY$  
270 IF i$="S" OR i$="s" THEN CL  
S  
280 GO TO 70  
290 IF i$="N" OR i$="n" THEN ST  
OP  
300 GO TO 260
```

PROGRAMA 2

```
10 REM *****  
* CURSO/BASIC *  
*****  
* INSERTAR *  
*****  
20 PRINT "Introduce una cadena  
(max. 30)."; LINE a$  
30 INPUT " "; LINE a$  
40 LET longitud=LEN a$  
50 IF longitud<1 OR longitud>30  
  THEN GO TO 30  
60 CLS  
70 PRINT "CADENA: ";  
80 PRINT a$;  
90 PRINT "LONGITUD: "; longitud  
100 PRINT "CADENA A INSERTAR: "  
110 INPUT " "; LINE b$  
120 IF b$="" THEN GO TO 110  
130 IF LEN b$>30 THEN GO TO 110  
140 PRINT b$;30 THEN GO TO 110  
150 LET posicion=VAL b$  
160 IF posicion<1 OR posicion>l  
ongitud THEN GO TO 110  
170 LET c$=a$(1 TO posicion)  
180 LET d$=a$(posicion+1 TO )  
190 LET e$=c$+b$+d$  
200 LET f$=LEN e$  
210 IF f$>30 THEN GO TO 110  
220 LET g$=e$;  
230 LET h$=LEN g$;  
240 PRINT c$; b$; d$;  
250 PRINT " "; LINE a$;  
260 PAUSE 10  
270 NEXT x  
280 PRINT d$;  
290 LET longitud=LEN a$  
300 LET longitud=LEN a$  
310 INSERT "S/N". Has caracteres a  
insertar S/N";  
320 IF i$="S" OR i$="s" THEN GO  
TO 70  
330 IF i$="N" OR i$="n" THEN ST  
OP  
340 GO TO 300
```

Estructura programa «ANULAR».



Acceso al teclado

LEN



SCREEN \$

Definición

La función «LEN» retorna un valor equivalente al número de caracteres de una cadena; hay que tener presente que los espacios en blanco también cuentan.

La estructura general de esta sentencia es:

FUNCION	ARGUMENTO
LEN	expresión de cadena

Ejemplos:

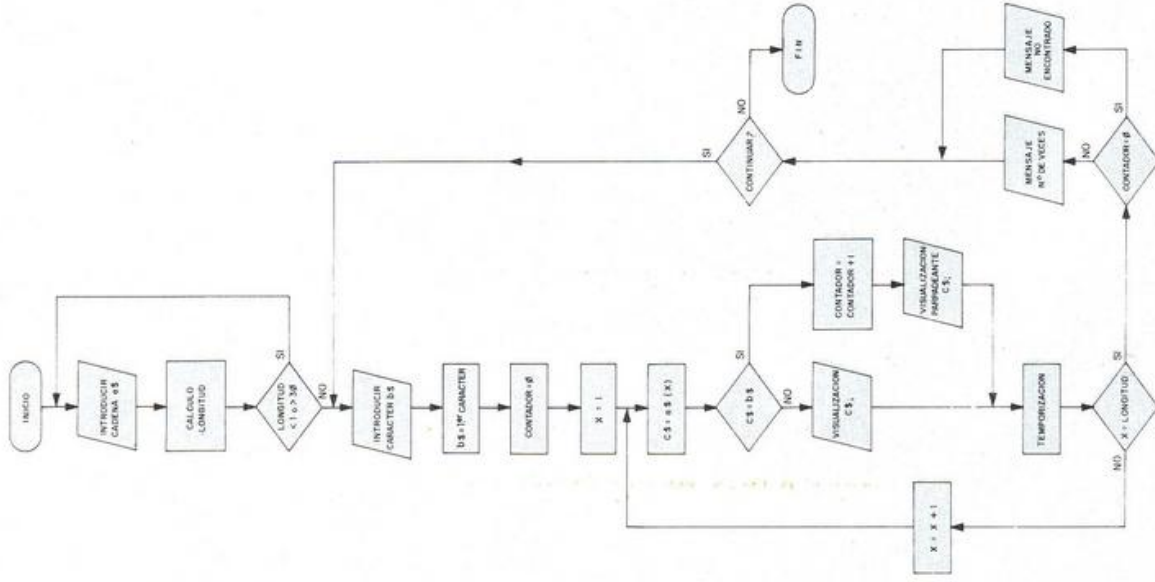
- PRINT LEN "BASIC"
- PRINT LEN a\$
- LET C\$ = LEN (b\$ + j\$)
- LET X\$ = LEN a\$ (3 TO)

El siguiente programa es un ejemplo de utilización de la sentencia «LEN»:

```

10 REM *****
20 PRINT "Introduce tu nombre"
30 INPUT "Nombre: "; N$
40 LET longitud=LEN N$
50 FOR i=1 TO longitud
60 LET longitud=1
70 NEXT i
80 PRINT "Tu nombre es: " & N$
90 PRINT "y tiene " & longitud & " caracteres."

```



Estructura programa «BUSQUEDA».

FUNCION	ARGUMENTO
CODE	expresión de cadena

Ejemplos:

- PRINT CODE "A"
- LET n = CODE a\$
- PRINT CODE b\$ (X)
- LET X 3 CODE "RAIZ"

Al retornar «CODE» el código del primer carácter, los siguientes comandos directos darán el mismo resultado (65):

- PRINT CODE "A"
- PRINT CODE "Alazan"
- PRINT CODE "Almeria"

La palabra clave «CODE» también es utilizada junto con «LOAD», «SAVE» y «VERIFY», pero esta particularidad será vista en otro capítulo.

En este programa se visualizan las letras mayúsculas de la «A» a la «Z» con su correspondiente código decimal. Observe que la función «CODE» se utiliza para controlar el bucle:

```

10 FOR n = CODE "a" TO CODE "z"
20 PRINT n, " "; CHR$(n);
30 NEXT n

```

En este otro, debe introducirse una cadena como máximo de «15» caracteres: la instrucción 70 permite centrar la visualización de dicha cadena. El bucle compuesto por las líneas «80» a «100» permite representar el código decimal de cada carácter, ya que se utiliza la función «CODE» junto con la fragmentación de la cadena.

Cuando el argumento es una cadena vacía ("") la función «CODE» retorna el valor «0».

```

10 REM *****
20 PRINT "Introduce un"
30 INPUT "Cadena: "; A$
40 LET longitud=LEN A$
50 IF longitud>15 THEN
60 PRINT "Error: cadena"
70 PRINT "demasiado larga"
80 FOR i=1 TO longitud/2
90 LET longitud=longitud-1
100 NEXT i
110 PRINT "Cadena correcta: " & A$

```

Funciones definidas de cadena

De la misma manera que el usuario puede definir sus propias funciones numéricas, también puede hacerlo con las de cadena.

La estructura general es:

SENTENCIA	ARGUMENTO
DEF FN	letra \$ (variables) = función

Ejemplos:

- DEF FN a\$ (b) = CHR\$(b) + "a"
- DEF FN b\$ (a\$, C\$) = a\$ + C\$
- DEF FN c\$ (a\$, n) = a\$ (n)
- DEF FN d\$ (X) = STR\$(X)

La llamada de las funciones sigue la estructura:

FN letra\$ (parámetros)

Ejemplos:

- PRINT FN a\$ (59)
- PRINT FN b\$ ("Pepe", "Juan")
- PRINT FN c\$ ("Camión", 3)
- PRINT FN d\$ (4 * 56)

Errores

Dos errores típicos en el

manejo de cadenas son:

a) C Nonnense in BASIC

Ocorre en las siguientes situaciones:

```
LET a$ = "1374a2"
PRINT VAL a$
```

ya que la variable «a\$» contiene un carácter no numérico. Sin embargo

```
LET a$ = "a13742"
PRINT VAL a$
```

da el error:

```
2 variable not found
```

ya que la función «VAL» al suprimir las comillas y detectar que el primer carácter es una letra, el sistema operativo interpreta que se trata de una variable numérica y al no encontrarla visualiza el error anterior. Observe la diferencia con las siguientes instrucciones:

```
LET valor = 30
LET a$ = "valor"
PRINT VAL a$
```

Con la función «VAL\$» también se produce el error «C»:

```
LET c$ = "oasis"
PRINT VAL$ c$
```

ya que o bien se modifica:

```
LET c$ = "oasis"
PRINT VAL$ c$
```

o por el contrario

```
PRINT VAL$ "C$"
```

b) 3 Subscript wrong.

Sucede cuando el índice

CADENA:
MICROHOBBY SEMANAL
LONGITUD: 18
CARACTER A BUSCAR: M
MICROHOBBY SEMANAL
Caracter encontrado 2 veces.

CADENA:
MICROHY SEMANAL
CADENA A INSERTAR:
OBB
POSICION: 6
MICROHOBBY SEMANAL

Búsqueda de caracteres.

Definición

Evalúa una cadena cuyo contenido son números y operadores matemáticos, y el resultado lo asigna a una variable numérica.

Su estructura general es:

FUNCION	ARGUMENTO
VAL	expresión de cadena

Ejemplos:

- PRINT VAL "23"
- LET a = VAL a\$
- PRINT VAL ("24" + "/" 2)
- LET n = VAL b\$ + c\$

Para realizar la evaluación, la función «VAL» suprime las comillas de la cadena.

El siguiente programa realiza las cuatro operaciones básicas entre dos números, éstos están asignados a las variables «a\$» y «b\$». El signo de la operación depende de la letra minúscula introducida por teclado (s - suma, r - resta, m - multiplicación y d - división) y queda asignada a la variable «C\$».

Para calcular el resultado se utiliza la función «VAL» en la línea 100.

```

10 REM *****
11 FUNCTION VAL :
12 *****
30 INPUT "numero 1 >>> ";a$
30 INPUT "numero 2 >>> ";b$
30 INPUT "operación (s/r/m/d) >>> ";c$
30 IF c$="s" THEN LET c$="+"
30 IF c$="r" THEN LET c$="-"
30 IF c$="m" THEN LET c$="*"
30 IF c$="d" THEN LET c$="/"
30 LET resultado=a$ c$ b$
30 GO TO 100
30 IF c$="+" THEN LET c$="+"
30 IF c$="-" THEN LET c$="-"
30 IF c$="*" THEN LET c$="*"
30 IF c$="/" THEN LET c$="/"
30 LET resultado=a$ c$ b$
30 GO TO 100
110 PRINT a$; " ";c$; " ";b$; " = ";resultado
120 GO TO 20

```

Insertión de cadenas.

Ejemplos:

- PRINT VAL\$ "" "BASIC""
- LET a\$ = VAL\$ "b\$"
- PRINT VAL\$ "a\$ + c\$"
- LET b\$ = VAL\$ "C\$" + "PE"

Conversiones de código

Hay dos funciones de cadena que permiten realizar cualquier conversión entre los códigos «ASCII» y «DECIMAL», éstas son:

— CHR\$
— CODE

Ver tabla del Código ASCII en la pag. 41.

CHR\$

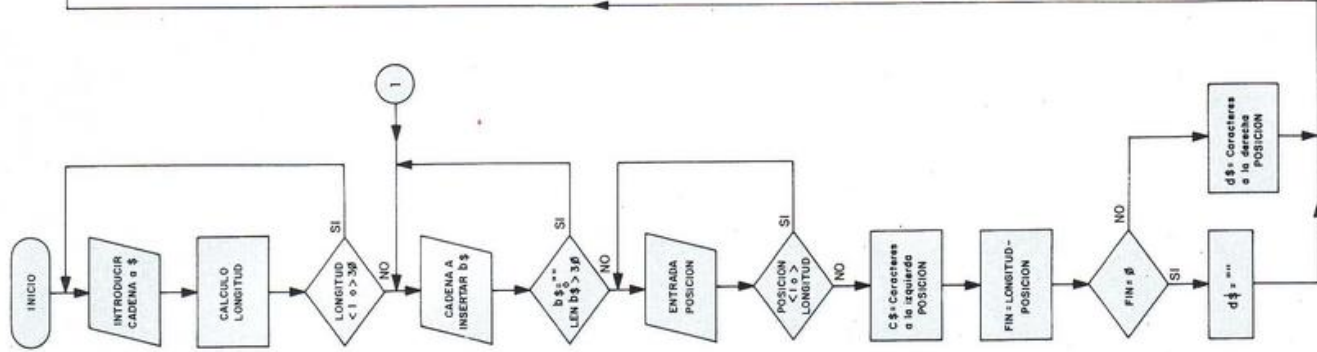
Acceso al teclado

CHR\$



Definición

La función «CHR\$» transforma un número decimal en



Estructura programa «INSERTAR».

205 : Contador en una unidad y se visualiza parpadeante (FLASH) el carácter.

210 : Temporización.

230 : Incremento de la variable de control del bucle.

240 : Si la variable «contador» es igual a cero se visualiza el mensaje: «Carácter no encontrado».

250-300 : Visualización del contenido de la variable «contador» en el caso de que sea distinto de cero.

255-300 : Análisis de las teclas «S» y «N» para buscar otro carácter o no.

El programa n.º 2 (INSERTAR) inserta un carácter o una serie de ellos dentro de otra cadena a partir de la posición que se quiera.

La estructura general del programa es:

10 : Comentario con el nombre del programa.

20-30 : Entrada de la cadena (a\$).

40 : Cálculo de su longitud.

50 : Verifica si está dentro de los límites.

70-80 : Visualización de la cadena.

100-110 : Entrada de la cadena a insertar (b\$).

120 : Comprueba que b\$ no es una cadena vacía.

130 : Comprueba que no tiene más de treinta caracteres.

140 : Visualización de la cadena a insertar.

150-170 : Entrada de la posición.

180 : Comprueba que la posición no es cero, ni mayor que la longitud de la cadena a\$.

185 : Visualización de la posición.

190 : Asignación a la variable c\$ de los caracteres situados a la izquierda de posición.

200 : Calcula cuantos caracteres hay situados a la derecha de posición.

210 : Si lo es, asigna a la variable d\$ el valor de una cadena vacía.

220 : Si no lo es, asigna a d\$ los caracteres situados a la derecha de posición.

230 : Si no hay ninguno, asigna a la variable d\$ una cadena vacía.

240 : Si hay, asigna a d\$ los caracteres situados a la derecha de posición.

250-330 : Concatenación de las variables c\$, b\$ y d\$. Visualización, a continuación, de los caracteres de la variable b\$ (temporizadores).

280 : Visualización de la variable d\$. Nuevo cálculo de la longitud de la cadena.

285 : Análisis de las teclas «S» y «N» para insertar otra cadena o no.

290-330 : El programa n.º 3 (ANULAR) borra el carácter indicado por su posición, dentro de una cadena que tenga «30» caracteres como máximo.

La estructura del programa es:

10 : Comentario con el nombre del programa.

20 : Entrada del valor numérico asignado a una variable de cadena (a\$).

30 : Comprueba de que la longitud no excede de 10 caracteres.

40 : Comienzo del bucle de análisis.

50 : Si el código del carácter es un «quinto» continúa el análisis de caracteres.

60 : Si el código pertenece a un número comprendido entre «0» y «9» se continúa con el análisis.

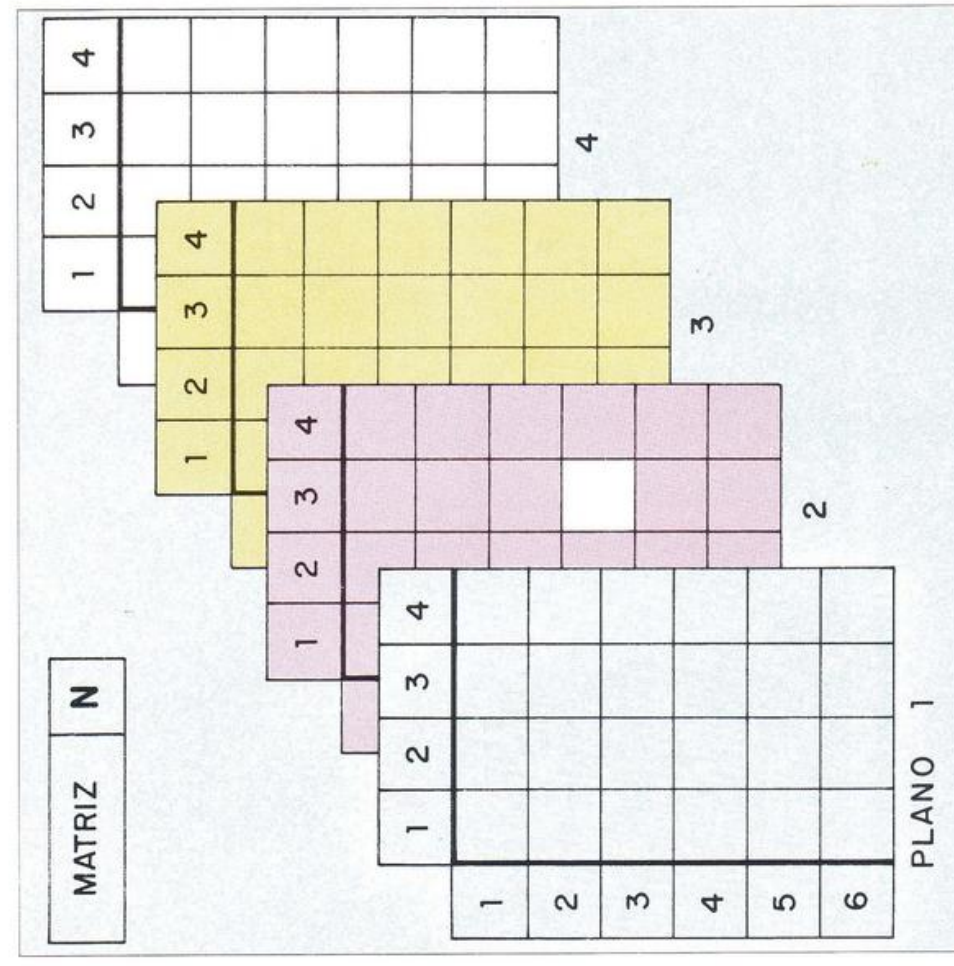
70 : El carácter al no ser ni un punto ni un número hace que la ejecución del bucle se interrumpa.

80 : Incremento de la variable de control del bucle.

90-100 : Si todos los caracteres son correctos se evalúa la variable valor y se visualiza su contenido.

120-130 : Visualización de un mensaje de error y del primer carácter no numérico introducido en la cadena.

140-190 : Análisis de las teclas «S» y «N» para introducir un nuevo valor o no.



Matriz tridimensional N(4,6,4), elemento N(2,4,3).

Recíprocamente, puede asignarse a una variable numérica el contenido de un determinado elemento. Ejemplos:

a) Asignar a la variable «número» el contenido del elemento 3 de la matriz «h».

```
LET número = h (3)
```

b) Asignar a la variable «cálculo» la raíz cuadrada del elemento situado en las coordenadas 7 y 2 de la matriz «p».

Programa 1. Búsqueda.

de sus elementos queda inicializado el valor cero.

Ejemplo:

```
10 DIM X(21)
20 FOR n = 1 TO 21
30 PRINT n, X(n)
40 NEXT n
```

Asignación y visualización

Para asignar un valor a un elemento determinado de una matriz, es necesario *posicionar* correctamente los subíndices o punteros. Por ejemplo:

a) Asignar el valor 30 al elemento 7 de una matriz «Z».

```
LET Z(7) = 30
```

b) Asignar al elemento situado en las coordenadas «y = 3» y «x = 5» de una matriz «J», el valor de la variable «patron».

```
LET J (3,5) = patron
```

Hay que tener presente que los punteros deben estar comprendidos dentro de los valores especificados al dimensionar la matriz.

```
1 > 50 > 50 > 50 > 50 > 50
2 > 20 > 20 > 20 > 20 > 20
3 > 10 > 10 > 10 > 10 > 10
4 > 10 > 10 > 10 > 10 > 10
5 > 10 > 10 > 10 > 10 > 10
```

```
3 > 4 > 5 > 6 > 7 > 8
4 > 5 > 6 > 7 > 8 > 9
5 > 6 > 7 > 8 > 9 > 10
6 > 7 > 8 > 9 > 10 > 11
7 > 8 > 9 > 10 > 11 > 12
```

```
3 > 2 > 3 > 4 > 5 > 6
4 > 3 > 4 > 5 > 6 > 7
5 > 4 > 5 > 6 > 7 > 8
6 > 5 > 6 > 7 > 8 > 9
7 > 6 > 7 > 8 > 9 > 10
```

Matriz bidimensional C(9,5), elemento C(4,2).

MATRICES

La matriz, tabla o array, es una estructura utilizada en programación que permite almacenar los datos (constantes o variables) de un programa.

Los elementos de una matriz tienen en común el mismo nombre y se diferencian en el subíndice, de esta manera se tiene el acceso directo a ellos.

Utilizando correctamente los subíndices pueden asignarse o leerse los contenidos de los elementos de una matriz.

Las matrices pueden ser de dos tipos:

- NUMERICAS
- DE CADENA

Dimensionado de matrices

Para reservar una serie de posiciones de memoria (tabla) es necesario, previamente, *dimensionar* una matriz; tarea que consiste en definir la cantidad de elementos de que va a constar así como su distribución; es decir, podremos dimensionar una matriz, de 40 elementos, de varias formas:

- Una, en que los elementos están consecutivos.
- También podrían estar distribuidos como en una especie de «tablero de ajedrez», que tuviera «5» filas por «8» columnas o «10» filas por «4»

- columnas, etc.
- Otra solución sería distribuirlos en dos planos formados cada uno por dos «tablas» de 4 * 5.
- etc.

Dependiendo de la distribución las matrices, reciben diversos nombres:

- UNIDIMENSIONALES
- BIDIMENSIONALES
- TRIDIMENSIONALES
- MULTIDIMENSIONALES

ADVERTENCIA

Las matrices antes de ser utilizadas deben estar «dimensionadas»

PROGRAMA 1

```
10 REM *****
* MANEJO DE TABLAS *
*****
20 INPUT "Numero de elementos"
30 IF elementos < 1 THEN PRINT " "
40 IF elementos < 10 THEN PRINT " "
50 DIM a(elementos)
60 REM *****
* ASIGNACION VALORES *
*****
65 RANDOMIZE
70 FOR n=1 TO elementos
80 LET contenido=INT (RAND*100)
90 LET a(n)=contenido
100 NEXT n
110 REM *****
```

```
*****
* VISUALIZACION *
*****
120 FOR n=1 TO elementos
130 PRINT n;
140 IF n<10 THEN PRINT " ";
150 NEXT n
*****
* BUSQUEDA *
*****
160 INPUT "Valor a encontrar >>"
170 IF valor < 1 THEN PRINT " "
180 IF valor < 10 THEN PRINT " "
190 IF valor < 100 THEN PRINT " "
200 LET veces=0
210 FOR n=1 TO elementos
220 IF a(n)=valor THEN veces=veces+1
230 NEXT n
240 PRINT "Se encontró el valor " valor " en " veces veces
```



```

*****
* EXPLORACION *
*****
220 FOR n=1 TO elementos
600 IF a(n) <> valor THEN GO TO 2
240 LET veces=veces+1
250 LET b(veces)=n
260 NEXT n
270 REM

*****
* VISUALIZACION *
*****
280 CLS
282 IF veces=0 THEN PRINT "¡Ualo
o valor, no encontrado!"; GO T
300 PRINT "El valor: ";valor
302 PRINT "ha sido encontrado:
veces: ";veces;
310 PRINT "en las posiciones: "
320 FOR n=1 TO veces
330 PRINT n; " ";b(n)
340 NEXT n
350 PRINT #0;"Otro valor (S/N) "
360 PAUSE 0
370 LET z$=INKEY$
380 IF z$="S" OR z$="s" THEN GO
TO 400
390 IF z$="N" OR z$="n" THEN GO
TO 410
400 CLS TO 360
410 FOR n=1 TO elementos
412 FOR n=1 TO elementos
414 PRINT n;
416 IF n<10 THEN PRINT " ";
420 PRINT " ";a(n);
430 NEXT n
440 REM

*****
* PAR/IMPAR *
*****
450 PRINT #0;"Pulsa una tecla p
ara continuar";
460 PAUSE 0
470 CLS
480 LET par=0
490 LET impar=0
500 DIM c(elementos)
510 DIM d(elementos)
520 REM

*****
* BUSQUEDA *
*****
530 FOR n=1 TO elementos
540 LET division=a(n)/2
550 IF INT (division)*2=a(n) TH
EN GO TO 590
560 LET impar=impar+1
570 LET d(impar)=a(n)

```

DIM

Tipo de sentencia

Comando de programa-
ción.

Acceso al teclado

DATA



MODO K

Definición

La sentencia «DIM» permite
dimensionar tanto las matri-
ces de tipo numérico como
las de cadena.

```

580 GO TO 610
590 LET par=par+1
600 LET c(par)=a(n)
610 NEXT n
*****
* VISUALIZACION *
*****
625 CLS
630 PRINT "Numeros pares encont
rados: ";par; "TO par";
640 FOR n=1 TO par
650 PRINT n; " ";c(n);
660 NEXT n
670 PRINT #0;"Pulsa una tecla p
ara continuar";
680 PAUSE 0
690 CLS
692 PRINT "Numeros impares enco
ntrados: ";impar; "TO impar";
700 FOR n=1 TO impar
710 PRINT n; " ";d(n);
720 NEXT n
730 PRINT #0;"Vuelvo a visualiz
ar (S/N)";
740 PAUSE 0
750 LET z$=INKEY$
760 IF z$="S" OR z$="s" THEN GO
TO 620
770 IF z$="N" OR z$="n" THEN GO
TO 790
780 CLS
790 FOR n=1 TO elementos
800 FOR n=1 TO elementos
802 PRINT n; THEN PRINT " ";
804 IF n<10 THEN PRINT " ";
810 PRINT " ";a(n);
820 NEXT n
830 REM

*****
* ORDENA *
*****
840 PRINT #0;"Pulsa una tecla p
ara continuar";
850 PAUSE 0
860 CLS
870 FOR x=1 TO elementos-1
880 LET menor=x
890 FOR y=x+1 TO elementos
900 IF a(menor)>a(y) THEN LET m
enor=y
910 NEXT y
920 LET cambio=a(x)
930 LET a(x)=a(menor)
940 LET a(menor)=cambio
950 PRINT x;
960 IF x<10 THEN PRINT " ";
970 PRINT " ";a(x);
980 NEXT x
990 PRINT "Elementos: ";
1000 IF elementos<10 THEN PRINT
1010 PRINT " ";a(elementos)
1020 PAUSE 0

```

Matrices numéricas

La estructura del dimensio-
nado de matrices numéricas
es:

SENTENCIA	ARGUMENTO
DIM	letra (lista de valores)

Ejemplos:

a) Matriz «unidimensional».

DIM X (20)

La matriz «X» se dimensiona
de 20 elementos consecuti-
vos. El subíndice varía, por
tanto, entre 1 y 20:

X (1), X (2)... X (20)

b) Matriz «bidimensional».

DIM J (5, 7)

La matriz «J» se dimensiona
de 35 elementos distribuidos
en 5 filas por 7 columnas. Los
subíndices varían desde:

J (1, 1), J (1, 2)... J (5, 6), J (5, 7)

c) Matriz «tridimensional».

DIM N (3, 4, 2)

Matriz llamada «N» consti-
tuida por 24 elementos distri-
buidos en tres planos de 4 fi-
las por 2 columnas. Los
subíndices varían entre:

N (1, 1, 1), N (1, 1, 2)... N (3, 4, 2)

Los subíndices también
pueden estar constituidos
por variables y expresiones
de tipo numérico.

Ejemplos:

- DIM C (7 * 3, 6)
- DIM P (Valor)
- DIM F (SIN cr, 5)
- DIM J (20 * X, SQR 9)

Como habrá podido obser-
var, el nombre de una matriz
está formado por una sola le-
tra que la distingue de las de-
más, por tanto, no puede ha-
ber dos matrices con el mis-
mo nombre aunque estén di-
mensionadas de distinta for-
ma, ya que al dimensionar
una matriz se borra cualquier
otra que tuviera el mismo

Matriz unidimensional B(18), elemento B(10).

MATRIZ "B"
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

SUBINDICE

nombre. Por el contrario, pue-
de haber una variable numé-
rica con el mismo nombre que
una matriz, ya que, el sistema
operativo distingue una de la
otra, por que, en su caso se
utilizan los subíndices y en el
otro no.
Cuando se dimensiona una
matriz numérica el contenido

Manejo de tablas

Las tablas de datos o matrices son quizás uno de los recursos más utilizados a la hora de elaborar un programa, por este motivo es imprescindible que el programador conozca a fondo su manejo.

El programa n.º «1» realiza diversos tratamientos con las matrices, por este motivo va a ser explicado con bastante detalle.

Las funciones básicas que realiza el programa son:

- Dimensionar una matriz en función de una variable.
- Asignar a la matriz valores aleatorios.
- Buscar un número determinado y visualizar cuántas veces aparece y en qué posiciones.
- Calcular cuáles son los números pares e impares y visualizarlos en dos listas separadas.
- Ordenar la tabla, generada aleatoriamente, de menor a mayor.

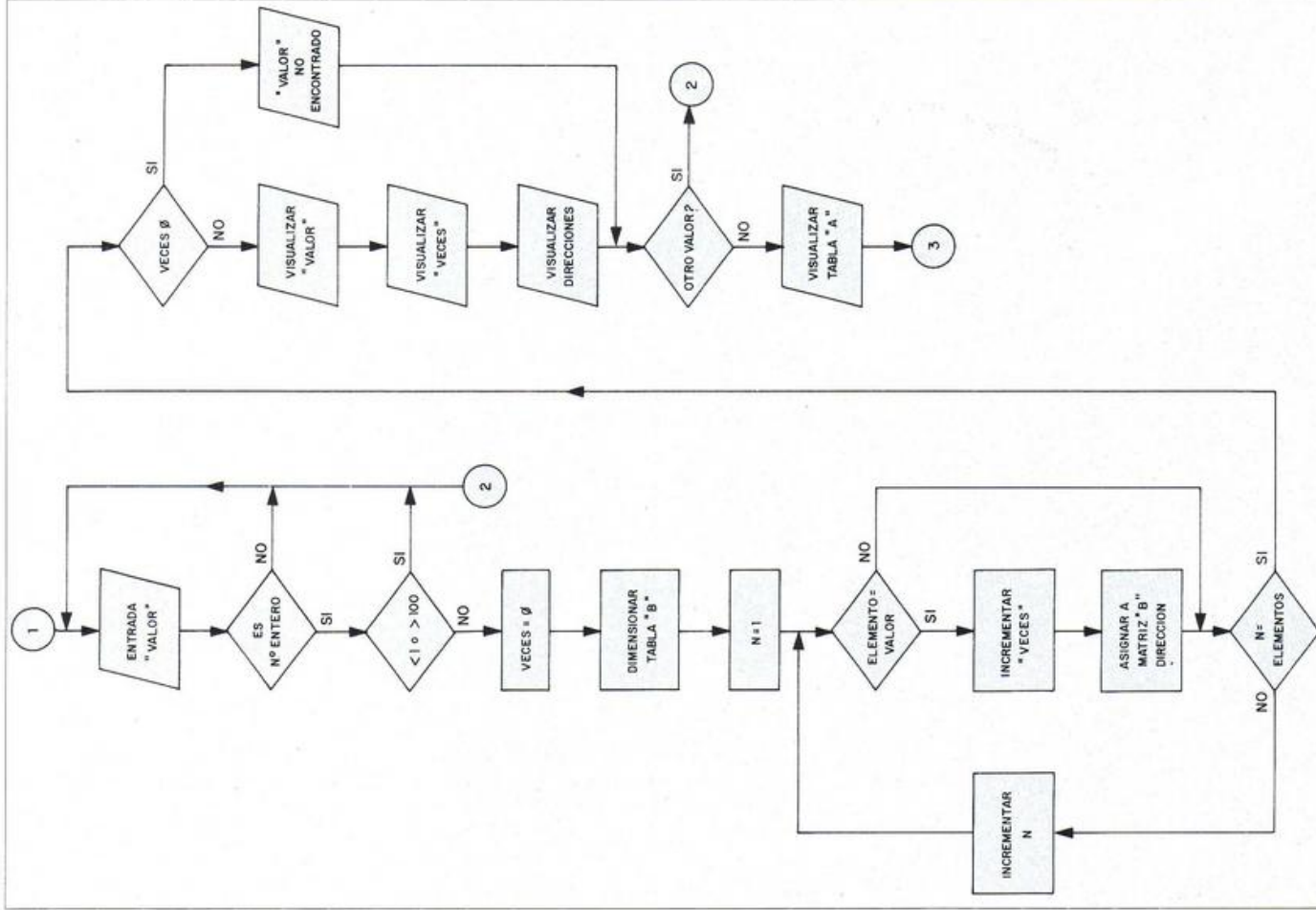
a) Dimensionado y asignación:

La línea 20 del programa permite introducir un número por teclado, éste queda asignado a la variable «elementos»; este número hace referencia a la cantidad de elementos de que va a constar nuestra tabla.

La línea 30 comprueba que el número introducido es entero, es decir, que no tiene decimales, de no ser así, el programa vuelve a pedir que introduzcamos otro número.

En la 40 se comprueba que el número está comprendido dentro del rango «1» a «40».

El dimensionado de la tabla se realiza en la línea 50. El nombre asignado es «a» y con



Manejo Tablas. Búsqueda.

```

10 REM *****
* MANEJO DE TABLAS *
* *****
* *****
20 INPUT "Numero de elementos
>>>"; elementos
30 IF elementos <> INT elementos
THEN GO TO 20
40 IF elementos < 1 OR elementos
> 40 THEN GO TO 20
50 DIM a (elementos)
60 REM *****
* ASIGNACION VALORES *
* *****
65 RANDOMIZE
70 FOR n=1 TO elementos
+1 80 LET contenido=INT (RND*100)
90 LET a(n)=contenido
100 NEXT n
110 REM *****
* VISUALIZACION *
* *****
120 FOR n=1 TO elementos
130 PRINT n;
132 IF n < 10 THEN PRINT " ";
134 PRINT ">"; a(n),
140 NEXT n

```

una dimensión definida en la variable «elementos».

En la línea 65 se realiza la inicialización de la función aleatoria, ésta depende del tiempo que lleve conectado el ordenador (variable FRAMES).

Dentro del bucle comprendido entre las líneas 70 a 100 se realiza la asignación de valores aleatorios.

La generación del número aleatorio se realiza en la línea 80. Debido al algoritmo empleado, la función retorna valores comprendidos entre los números 1 y 100. El valor queda asignado a la variable «contenido».

En la línea 90 se realiza la asignación del valor de la variable «contenido», al elemento apuntado por la variable de control del bucle (n).

La visualización del contenido de la matriz lo lleva a cabo en el bucle formado por las líneas 120 a 140.

La línea 130 visualiza el n.º de subíndice. La 132 visualiza a continuación un espacio en blanco, si el número de subíndice es menor que 10; esto permite que la visualización de los números quede alineada.

La visualización de los contenidos de la matriz, seleccionados por la variable de con-

tro] «n», lo realiza la línea 134. El formato de salida es a dos columnas.

b) Búsqueda:

En la línea 160 se realiza la entrada del valor a buscar.

La línea 170 comprueba si el número introducido es entero o no, y en la 180 si está comprendido dentro del rango 1 a 100.

En la 190 se inicializa la variable «veces» con el valor «0». En la línea 200 se dimensiona la matriz «b», ésta va a contener las direcciones de la matriz «a» correspondientes al valor buscado.

El bucle de exploración de los elementos de la matriz «a» está localizado entre las líneas 220 y 260.

En la línea 230 se comprueba si el elemento direccionado por la variable de control de bucle (n) es igual al contenido de la variable «valor».

Si son iguales en la línea 240 se incrementa la variable «veces», que contiene el número de ocasiones en que el valor es encontrado.

En la 250 se asigna a elemento de la matriz «b», seleccionado por la variable «veces», el valor de la variable de control del bucle (n), que indica la dirección de la tabla «a» donde el valor ha sido encontrado.

La línea 280 «limpia» la pantalla para poder visualizar los resultados de la búsqueda.

En la 282 se comprueba si el número ha sido encontrado. En caso negativo se visualiza un mensaje indicando el hecho.

Si al menos en una ocasión el valor se encuentra, las líneas 290 a 300 se encar-

```

150 REM
***
* BUSQUEDA *
*
*
*
160 INPUT "Valor a encontrar >>" ; valor
170 IF valor<>INT valor THEN GO TO 160
180 IF valor<1 OR valor>100 THEN GO TO 160
190 LET veces=0
200 DIM b(veces)
210 REM
***
* EXPLORACION *
*
*
220 FOR n=1 TO elementos
230 IF a(n)<>valor THEN GO TO 240
240 LET veces=veces+1
250 LET b(veces)=n
260 NEXT n
270 REM
***
* VISUALIZACION *
*
*
280 CLS
282 IF veces=0 THEN PRINT "Valor "; valor; " no encontrado": GO TO 350
290 PRINT "El valor: "; valor
300 PRINT "ha sido encontrado: "; veces; " veces"
310 PRINT "en las posiciones: "
FOR n=1 TO veces
330 PRINT n; "; b(n)"
340 NEXT n
350 PRINT #0; "Otro valor (S/N)"
360 PAUSE 0
370 LET z$=INKEY$
380 IF z$="S" OR z$="s" THEN GO TO 150
390 IF z$="N" OR z$="n" THEN GO TO 410
400 GO TO 360
410 CLS
412 FOR n=1 TO elementos
414 PRINT n;
416 IF n<10 THEN PRINT " ";
420 PRINT n; "; a(n)";
430 NEXT n

```

```

05 ordenador=valor carta
1550 GO TO 1300
1560 REM
***
* TANTOS ORDENADOR *
*
*
1570 PRINT AT 0,0;"ME QUEDO"
1580 LET p(1)=STR$ puntos orden
1590 PRINT AT 0,0; INVERSE 1;"HECIDO UN TOTAL DE p(1)"
1592 PRINT #0;"Pulse una tecla para continuar."
1594 PAUSE 0
1600 REM
***
* COMPARACION *
*
*
1610 IF puntos ordenador=puntos jugador THEN LET ganador=0
1620 IF puntos ordenador<puntos jugador THEN LET ganador=1
1630 REM
***
* GANADOR *
*
*
1635 CLS
1640 IF ganador=0 THEN PRINT AT 0,0;"Ganado";
1650 IF ganador=1 THEN PRINT AT 0,0;"Has ganado";
1660 PRINT AT 0,0;"LA PROXIMA GANARE YO"
1670 PRINT AT 0,4;"Puntos Jugado"; puntos jugador
1680 PRINT AT 0,4;"Puntos Ordenador"; puntos ordenador
1690 IF puntos ordenador=0 THEN PRINT "Puntos ordenador para continuar."
1700 PRINT #0;"Pulse una tecla para continuar."
1710 PAUSE 0
1720 REM
***
* CALCULOS *
*
*
1730 IF ganador=0 THEN GO TO 1790
1740 IF puntos jugador=7.5 THEN LET doble=2; GO TO 1760
1750 LET doble=1
1760 LET cred ordenador=cred ordenador-apuesta+doble
1770 LET cred jugador=cred jugador+apuesta+doble
1780 GO TO 1810
1790 LET cred ordenador=cred ordenador-apuesta
1800 LET cred jugador=cred jugador+apuesta
1810 REM
***
* RESULTADOS *
*
*

```

```

1815 CLS
1820 PRINT AT 7,3;"Manos jugadas"; manos
1830 PRINT AT 12,3;"Credito Juguador"; cred jugador
1840 PRINT AT 17,3;"Credito Ordenador"; cred ordenador
1850 IF cred jugador<0 THEN PRINT AT 3,0;"HAS ROTADO LA BANCA";
1860 IF cred jugador=0 THEN PRINT AT 3,0;"SE ACABO TU DINERO";
1870 PRINT AT 3,0;"CREDITOS ";
1875 REM

```

```

*****
* CONTINUAR? *
*
*****
1880 PRINT #0;"Quieres jugar ot"; manos/n;
1890 PAUSE 0
1900 LET z$=INKEY$
1910 IF z$="S" OR z$="s" THEN GO TO 1940
1920 IF z$="N" OR z$="n" THEN IN PUT 0; PAUSE 0; STOP
1930 GO TO 1890
1940 INPUT 0
1950 LET manos=manos+1
1960 FOR a=1 TO 10
1970 LET b(a,b)=0
1980 NEXT b
1990 NEXT a
2000 GO TO 380
2100 REM

```

```

*****
* INSTRUCCIONES *
*
*****

```

```

2110 PRINT #0; AT 1,5;"Instruccio (S/N)";
2120 PAUSE 0
2130 LET z$=INKEY$
2140 IF z$="S" OR z$="s" THEN GO TO 2165
2150 IF z$="N" OR z$="n" THEN RE TURN GO TO 2120
2160 CLS
2170 PRINT AT 0,0;"INST"

```

```

2180 PRINT "Este programa est a basado en... el conocido jueg a de naipes... Las 7 1/2 manos...
2190 PRINT AT 10,1;"Tienes un cr edito inicial de: 12,5";
2200 PRINT AT 15,1;"Las apuestas 000 pesetas";
2210 PRINT AT 20,1;"Si tienes 7 1/2 manos al día, un benefi cio de 200 ptas";
2220 PRINT #0;"Pulse una tecla para continuar";
2230 PAUSE 0
2240 RETURN

```



```

730 LET puntos_jugador=0
740 LET carta=1
750 PRINT AT 3,0;" ";
760 GO SUB 800
770 LET puntos_jugador=valor_c
780
790 LET carta=2
800 GO TO 860
810 REM
820
830 PRESENTACION *
840 *
850 *
860 *
870 *
880 *
890 *
900 *
910 LET numero=c(carta,1)
920 LET palo=c(carta,2)
930 PRINT AT 1,0;"Palo: ";palo;
940 LET valor_carta=numero)
950 RETURN
960
970 *
980 *
990 *
1000 *
1010 *
1020 *
1030 *
1040 *
1050 *
1060 *
1070 *

```

```

870 PRINT #0; AT 1,3;"Quieres ot
880 carta (S/N)"; THEN GO TO 880
890 IF INKEY$="" THEN GO TO 880
900 LET Z$=INKEY$
910 IF Z$="S" OR Z$="s" THEN GO
TO 930
920 IF Z$="N" OR Z$="n" THEN LE
T indice=carta; GO TO 1140
930 GO TO 880
940 INPUT #0
950 LET x=1 TO 35; NEXT x
960 LET carta=carta+1
970 PRINT AT 1,0;"Palo: ";
980 GO SUB 800
990 LET puntos_jugador=puntos_j
ugador+valor_carta
1000 IF puntos_jugador>7.5 THEN
GO TO 1000
1010 IF puntos_jugador=7.5 THEN
GO TO 1070
1020 GO TO 870
1030 PRINT AT 0,0;"LO SIENTO,"
1040 PRINT AT 0,0;"
1050 LET P$(1)=STR$ puntos_jugad
or
1060 PRINT AT 21,0; INVERSE 1;"
1070 HECHO UN TOTAL DE ";P$(1);
1080 LET ganador=0
1090 LET puntos_ordenador=0
1100 PRINT #0;"Pulsa una tecla
para continuar"
1110 GO TO 1530
1120 PAUSE 0
1130 GO TO 1530
1140 REM
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
2050
2060
2070
2080
2090
2100
2110
2120
2130
2140
2150
2160
2170
2180
2190
2200
2210
2220
2230
2240
2250
2260
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420
2430
2440
2450
2460
2470
2480
2490
2500
2510
2520
2530
2540
2550
2560
2570
2580
2590
2600
2610
2620
2630
2640
2650
2660
2670
2680
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
2980
2990
3000
3010
3020
3030
3040
3050
3060
3070
3080
3090
3100
3110
3120
3130
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
3270
3280
3290
3300
3310
3320
3330
3340
3350
3360
3370
3380
3390
3400
3410
3420
3430
3440
3450
3460
3470
3480
3490
3500
3510
3520
3530
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720
3730
3740
3750
3760
3770
3780
3790
3800
3810
3820
3830
3840
3850
3860
3870
3880
3890
3900
3910
3920
3930
3940
3950
3960
3970
3980
3990
4000
4010
4020
4030
4040
4050
4060
4070
4080
4090
4100
4110
4120
4130
4140
4150
4160
4170
4180
4190
4200
4210
4220
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490
4500
4510
4520
4530
4540
4550
4560
4570
4580
4590
4600
4610
4620
4630
4640
4650
4660
4670
4680
4690
4700
4710
4720
4730
4740
4750
4760
4770
4780
4790
4800
4810
4820
4830
4840
4850
4860
4870
4880
4890
4900
4910
4920
4930
4940
4950
4960
4970
4980
4990
5000
5010
5020
5030
5040
5050
5060
5070
5080
5090
5100
5110
5120
5130
5140
5150
5160
5170
5180
5190
5200
5210
5220
5230
5240
5250
5260
5270
5280
5290
5300
5310
5320
5330
5340
5350
5360
5370
5380
5390
5400
5410
5420
5430
5440
5450
5460
5470
5480
5490
5500
5510
5520
5530
5540
5550
5560
5570
5580
5590
5600
5610
5620
5630
5640
5650
5660
5670
5680
5690
5700
5710
5720
5730
5740
5750
5760
5770
5780
5790
5800
5810
5820
5830
5840
5850
5860
5870
5880
5890
5900
5910
5920
5930
5940
5950
5960
5970
5980
5990
6000
6010
6020
6030
6040
6050
6060
6070
6080
6090
6100
6110
6120
6130
6140
6150
6160
6170
6180
6190
6200
6210
6220
6230
6240
6250
6260
6270
6280
6290
6300
6310
6320
6330
6340
6350
6360
6370
6380
6390
6400
6410
6420
6430
6440
6450
6460
6470
6480
6490
6500
6510
6520
6530
6540
6550
6560
6570
6580
6590
6600
6610
6620
6630
6640
6650
6660
6670
6680
6690
6700
6710
6720
6730
6740
6750
6760
6770
6780
6790
6800
6810
6820
6830
6840
6850
6860
6870
6880
6890
6900
6910
6920
6930
6940
6950
6960
6970
6980
6990
7000
7010
7020
7030
7040
7050
7060
7070
7080
7090
7100
7110
7120
7130
7140
7150
7160
7170
7180
7190
7200
7210
7220
7230
7240
7250
7260
7270
7280
7290
7300
7310
7320
7330
7340
7350
7360
7370
7380
7390
7400
7410
7420
7430
7440
7450
7460
7470
7480
7490
7500
7510
7520
7530
7540
7550
7560
7570
7580
7590
7600
7610
7620
7630
7640
7650
7660
7670
7680
7690
7700
7710
7720
7730
7740
7750
7760
7770
7780
7790
7800
7810
7820
7830
7840
7850
7860
7870
7880
7890
7900
7910
7920
7930
7940
7950
7960
7970
7980
7990
8000
8010
8020
8030
8040
8050
8060
8070
8080
8090
8100
8110
8120
8130
8140
8150
8160
8170
8180
8190
8200
8210
8220
8230
8240
8250
8260
8270
8280
8290
8300
8310
8320
8330
8340
8350
8360
8370
8380
8390
8400
8410
8420
8430
8440
8450
8460
8470
8480
8490
8500
8510
8520
8530
8540
8550
8560
8570
8580
8590
8600
8610
8620
8630
8640
8650
8660
8670
8680
8690
8700
8710
8720
8730
8740
8750
8760
8770
8780
8790
8800
8810
8820
8830
8840
8850
8860
8870
8880
8890
8900
8910
8920
8930
8940
8950
8960
8970
8980
8990
9000
9010
9020
9030
9040
9050
9060
9070
9080
9090
9100
9110
9120
9130
9140
9150
9160
9170
9180
9190
9200
9210
9220
9230
9240
9250
9260
9270
9280
9290
9300
9310
9320
9330
9340
9350
9360
9370
9380
9390
9400
9410
9420
9430
9440
9450
9460
9470
9480
9490
9500
9510
9520
9530
9540
9550
9560
9570
9580
9590
9600
9610
9620
9630
9640
9650
9660
9670
9680
9690
9700
9710
9720
9730
9740
9750
9760
9770
9780
9790
9800
9810
9820
9830
9840
9850
9860
9870
9880
9890
9900
9910
9920
9930
9940
9950
9960
9970
9980
9990
10000

```

```

1030 PRINT AT 0,0;"ARRIBA SIET
1040 PRINT AT 21,0;"ESPERA A C
1050 PRINT AT 21,0;"
1060 LET indice=carta
1070 PRINT #0;"Pulsa una tecla
para continuar"
1080 GO TO 1200
1090 REM
1100
1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
2050
2060
2070
2080
2090
2100
2110
2120
2130
2140
2150
2160
2170
2180
2190
2200
2210
2220
2230
2240
2250
2260
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420
2430
2440
2450
2460
2470
2480
2490
2500
2510
2520
2530
2540
2550
2560
2570
2580
2590
2600
2610
2620
2630
2640
2650
2660
2670
2680
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
2980
2990
3000
3010
3020
3030
3040
3050
3060
3070
3080
3090
3100
3110
3120
3130
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
3270
3280
3290
3300
3310
3320
3330
3340
3350
3360
3370
3380
3390
3400
3410
3420
3430
3440
3450
3460
3470
3480
3490
3500
3510
3520
3530
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720
3730
3740
3750
3760
3770
3780
3790
3800
3810
3820
3830
3840
3850
3860
3870
3880
3890
3900
3910
3920
3930
3940
3950
3960
3970
3980
3990
4000
4010
4020
4030
4040
4050
4060
4070
4080
4090
4100
4110
4120
4130
4140
4150
4160
4170
4180
4190
4200
4210
4220
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490
4500
4510
4520
4530
4540
4550
4560
4570
4580
4590
4600
4610
4620
4630
4640
4650
4660
4670
4680
4690
4700
4710
4720
4730
4740
4750
4760
4770
4780
4790
4800
4810
4820
4830
4840
4850
4860
4870
4880
4890
4900
4910
4920
4930
4940
4950
4960
4970
4980
4990
5000
5010
5020
5030
5040
5050
5060
5070
5080
5090
5100
5110
5120
5130
5140
5150
5160
5170
5180
5190
5200
5210
5220
5230
5240
5250
5260
5270
5280
5290
5300
5310
5320
5330
5340
5350
5360
5370
5380
5390
5400
5410
5420
5430
5440
5450
5460
5470
5480
5490
5500
5510
5520
5530
5540
5550
5560
5570
5580
5590
5600
5610
5620
5630
5640
5650
5660
5670
5680
5690
5700
5710
5720
5730
5740
5750
5760
5770
5780
5790
5800
5810
5820
5830
5840
5850
5860
5870
5880
5890
5900
5910
5920
5930
5940
5950
5960
5970
5980
5990
6000
6010
6020
6030
6040
6050
6060
6070
6080
6090
6100
6110
6120
6130
6140
6150
6160
6170
6180
6190
6200
6210
6220
6230
6240
6250
6260
6270
6280
6290
6300
6310
6320
6330
6340
6350
6360
6370
6380
6390
6400
6410
6420
6430
6440
6450
6460
6470
6480
6490
6500
6510
6520
6530
6540
6550
6560
6570
6580
6590
6600
6610
6620
6630
6640
6650
6660
6670
6680
6690
6700
6710
6720
6730
6740
6750
6760
6770
6780
6790
6800
6810
6820
6830
6840
6850
6860
6870
6880
6890
6900
6910
6920
6930
6940
6950
6960
6970
6980
6990
7000
7010
7020
7030
7040
7050
7060
7070
7080
7090
7100
7110
7120
7130
7140
7150
7160
7170
7180
7190
7200
7210
7220
7230
7240
7250
7260
7270
7280
7290
7300
7310
7320
7330
7340
7350
7360
7370
7380
7390
7400
7410
7420
7430
7440
7450
7460
7470
7480
7490
7500
7510
7520
7530
7540
7550
7560
7570
7580
7590
7600
7610
7620
7630
7640
7650
7660
7670
7680
7690
7700
7710
7720
7730
7740
7750
7760
7770
7780
7790
7800
7810
7820
7830
7840
7850
7860
7870
7880
7890
7900
7910
7920
7930
7940
7950
7960
7970
7980
7990
8000
8010
8020
8030
8040
8050
8060
8070
8080
8090
8100
8110
8120
8130
8140
8150
8160
8170
8180
8190
8200
8210
8220
8230
8240
8250
8260
8270
8280
8290
8300
8310
8320
8330
8340
8350
8360
8370
8380
8390
8400
8410
8420
8430
8440
8450
8460
8470
8480
8490
8500
8510
8520
8530
8540
8550
8560
8570
8580
8590
8600
8610
8620
8630
8640
8650
8660
8670
8680
8690
8700
8710
8720
8730
8740
8750
8760
8770
8780
8790
8800
8810
8820
8830
8840
8850
8860
8870
8880
8890
8900
8910
8920
8930
8940
8950
8960
8970
8980
8990
9000
9010
9020
9030
9040
9050
9060
9070
9080
9090
9100
9110
9120
9130
9140
9150
9160
9170
9180
9190
9200
9210
9220
9230
9240
9250
9260
9270
9280
9290
9300
9310
9320
9330
9340
9350
9360
9370
9380
9390
9400
9410
9420
9430
9440
9450
9460
9470
9480
9490
9500
9510
9520
9530
9540
9550
9560
9570
9580
9590
9600
9610
9620
9630
9640
9650
9660
9670
9680
9690
9700
9710
9720
9730
9740
9750
9760
9770
9780
9790
9800
9810
9820
9830
9840
9850
9860
9870
9880
9890
9900
9910
9920
9930
9940
9950
9960
9970
9980
9990
10000

```

gan de visualizar el «valor» y el número de «veces»; y el bucle formado por las líneas 320 a 340 de visualizar las posiciones.

Las líneas 350 a 400 comprueban si se desea buscar otro valor o no.

En caso negativo se «limpia» la pantalla y se vuelve a visualizar el contenido de la matriz de origen.

c) Par-Impar:

Las líneas 450 a 460 temporizan hasta que se pulsa una tecla y la 470 «limpia» la pantalla.

La 480 y 490 asignan a las variables «par» e «impar» el valor inicial 0.

La matriz «C» se dimensiona en la línea 500 y se utiliza para almacenar los números pares encontrados. La 510 dimensiona la tabla «D» que hace lo mismo con los números impares.

El bucle de búsqueda de números pares o impares está implementado entre las líneas 530 y 610.

La línea 540 asigna a la variable «division» el cociente entre el elemento de la matriz «a», direccionado por la variable de control (n), y el número dos.

La forma de calcular si el número par o no se realiza en la línea 550. Un número es par cuando es divisible por dos, por lo tanto, si se multiplica la parte entera del cociente («division») por el divisor (2) y el resultado es igual al dividendo (a(n)), es que el resto es igual a cero, y por tanto, par.

Si el número es impar en las líneas 560 y 570 se incrementa la variable «impar», que almacena el número total de impares, y se asigna el número al elemento de la matriz «d», direccionada por el puntero

```

440 REM
450 PRINT #0;"Pulsa una tecla para
continuar"
460 PAUSE 0
470 CLS
480 LET par=0
490 LET impar=0
500 DIM c(elementos)
510 DIM d(elementos)
520 REM
530 FOR n=1 TO elementos
540 LET division=a(n)/2
550 IF INT (division)*2=a(n) TH
EN GO TO 590
560 LET impar=impar+1
570 LET d(impar)=a(n)
580 GO TO 610
590 LET par=par+1
600 LET c(par)=a(n)
610 NEXT n
620 REM
630 CLS
640 PRINT "Números pares encont
rados: ";par
650 FOR n=1 TO par
660 PRINT " ";c(n),
670 NEXT n
680 PRINT #0;"Pulsa una tecla para
continuar"
690 PAUSE 0
700 CLS
710 PRINT "Números impares enco
ntrados: ";impar
720 FOR n=1 TO impar
730 PRINT " ";d(n),
740 NEXT n
750 PRINT #0;"Vuelvo a visualiz
ar los (S/N)"
760 PAUSE 0
770 LET Z$=INKEY$
780 IF Z$="S" OR Z$="s" THEN GO
TO 620
790 IF Z$="N" OR Z$="n" THEN GO
TO 790
800 CLS
810 FOR n=1 TO elementos
820 PRINT n; THEN PRINT " ";
830 IF n<10 THEN PRINT " ";
840 PRINT " ";a(n),
850 NEXT n

```

```

625 CLS
630 PRINT "Números pares encont
rados: ";par
640 FOR n=1 TO par
650 PRINT " ";c(n),
660 NEXT n
670 PRINT #0;"Pulsa una tecla para
continuar"
680 PAUSE 0
690 CLS
700 PRINT "Números impares enco
ntrados: ";impar
710 FOR n=1 TO impar
720 PRINT " ";d(n),
730 NEXT n
740 PRINT #0;"Vuelvo a visualiz
ar los (S/N)"
750 PAUSE 0
760 LET Z$=INKEY$
770 IF Z$="S" OR Z$="s" THEN GO
TO 620
780 IF Z$="N" OR Z$="n" THEN GO
TO 790
790 CLS
800 FOR n=1 TO elementos
810 PRINT n; THEN PRINT " ";
820 IF n<10 THEN PRINT " ";
830 PRINT " ";a(n),
840 NEXT n

```


PROGRAMA 2

```

0000 REM
0010 *****
0020 ** ORDENA **
0030 **
0040 *****
0050 PRINT #0;"Pulsa una teclap
0060 continuar";
0070 CLS
0080 x=1 TO elementos-1
0090 LET menor=x
0100 FOR y=x+1 TO elementos
0110 IF a(menor)>a(y) THEN LET m
0120     color=y
0130 NEXT y
0140 cambio=a(x)
0150 LET a(x)=a(menor)
0160 LET a(menor)=cambio
0170 PRINT x;
0180 IF x<10 THEN PRINT " ";
0190 NEXT x
0200 PRINT elementos;
0210 IF elementos<10 THEN PRINT
0220     " ";a(elementos)
0230 PRINT #0

```

valor correspondiente de la variable de control del bucle (x).

El bucle de comparación se encuentra implementado en las líneas 890 a 910.

Al final del bucle la variable «menor» contiene el menor número de los comparados.

En las líneas 920 a 940 se realiza el intercambio entre los elementos direccionados por los subíndices «x» y «menor»; este intercambio se apoya en la variable «cambio» para poder realizarlo.

La visualización de la posición y del valor ordenado lo ejecutan las líneas 950 a 970.

Para finalizar la visualización del último elemento (mayor), lo llevan a cabo las líneas 990 a 1010.

[illegible]

Programa 1. Ordenación.

«impar». En las líneas 590 y 600 se realiza la misma operación para los números pares.

La línea 625 «limpia» la pantalla. En la línea 630 se visualiza el número de pases en contrados. El bucle formado por los números 640 a 660 visualiza los valores de la matriz «C».

Las líneas 670 a 685 realizan la función de pausa y borrado.

La visualización del total de
impares y sus valores, está
implementado en las líneas
690 a 720.

Las líneas 730 a 780 comprueban si se desea volver a visualizar los números. En caso negativo, las líneas 790 a 820 nos vuelven a visualizar la tabla original.

d) Ordenación:

El último paso que queda es la ordenación de la tabla, para ello se ha utilizado el siguiente procedimiento.

Se compara el primer número de la tabla con el resto, si resulta que es el más pequeño, se queda donde está; pero si no, el valor más pequeño se sitúa en la primera posición y el primer número en su lugar. En el siguiente ciclo se realiza la misma operación con el número dos, este ciclo se repite tantas veces como elementos tenga la tabla menos uno, ya que el último valor queda automáticamente ordenado.

Las líneas 840 a 860 temporizan y posteriormente borran la pantalla cuando se pulsa una tecla.

El bucle formado por las líneas 870 a 980 es el dedicado a «barrer» todos los números menos uno.

La variable «menor» se inicializa en la línea 880 con el

[illegible]

```

905 GO SUB 2100
910 REM *****
    * BARAJEAR *
    * *****
370 RANDOMIZE
380 CLS
390 PRINT AT 3,0;" ESPERA "
BARAJEAR
400 PRINT AT 13,0;" VOY A BA
RAJEAR "
410 FOR x=1 TO 40
420 GO SUB 490
430 IF b(numero,palo)=1 THEN GO
TO 420
440 LET m(x,1)=numero
450 LET m(x,2)=palo
460 LET b(numero,palo)=1
470 NEXT x
480 GO TO 530
490 REM *****
    * CARTA ALEATORIA *
    * *****
500 LET numero=INT (RAND*10)+1
510 LET palo=INT (RAND*4)+1
520 RETURN
530 REM *****
    * CORTAR EL MAZO *
    * *****
quiere cortar?"
570 INPUT n$; LINE n$
572 IF n$="" THEN GO TO 570
574 IF x=1 TO LEN n$
576 FOR n$(x)="" OR n$(x)="9" T
HEN GO TO 570
578 NEXT x
580 LET numero=VAL n$
590 LET numero<1 OR numero>40 TH
EN GO TO 570
592 GO TO 570
594 PRINT AT 3,0;" ESPERA "
600 LET resto=40-numero
610 FOR x=1 TO numero
620 LET c(x+resto,1)=m(x,1)
630 LET c(x+resto,2)=m(x,2)
640 NEXT x
650 FOR x=numero+1 TO 40
660 LET c(x-numero,1)=m(x,1)
670 LET c(x-numero,2)=m(x,2)
675 NEXT x
680 PRINT AT 3,0;" PODEMOS E
MPEZAR A JUGAR "
690 PRINT AT 13,0;" !! BUEN
SUETE !!!
700 PRINT #0;" Pulsa una tecla
para continuar."
710 CLS
715 GOTO 720
720 REM *****
    * INICIAL JUGADOR *
    * *****
TU JUEGAS
722 PRINT AT 0,0;"

```


2 Variable not found

b) Cuando uno de los subíndices es negativo o mayor que «65535» aparece:

8 Integer out of range

Ejemplo:

– PRINT a\$(12,-1)
– PRINT a\$(67000,3)

c) Cuando un subíndice está fuera de los límites del dimensionado de una matriz, se visualiza

3 Subscript wrong

Por ejemplo, en la matriz

«b\$(20,10)»
– LET b\$(0,5)
– PRINT b\$(7,11)

d) Cuando se dimensiona una tabla demasiado grande que ocupa toda la memoria destinada a los programas «BASIC», aparece el error

4 Out of memory

Grabación de datos

Con el comando «SAVE» es posible almacenar o grabar en cinta las tablas generadas en alguno de nuestros programas; también pueden ser cargadas, posteriormente en el ordenador, con el comando «LOAD».

La sintaxis es la siguiente:

SAVE "nombre" DATA letra ()

en el caso de matrices numéricas, y

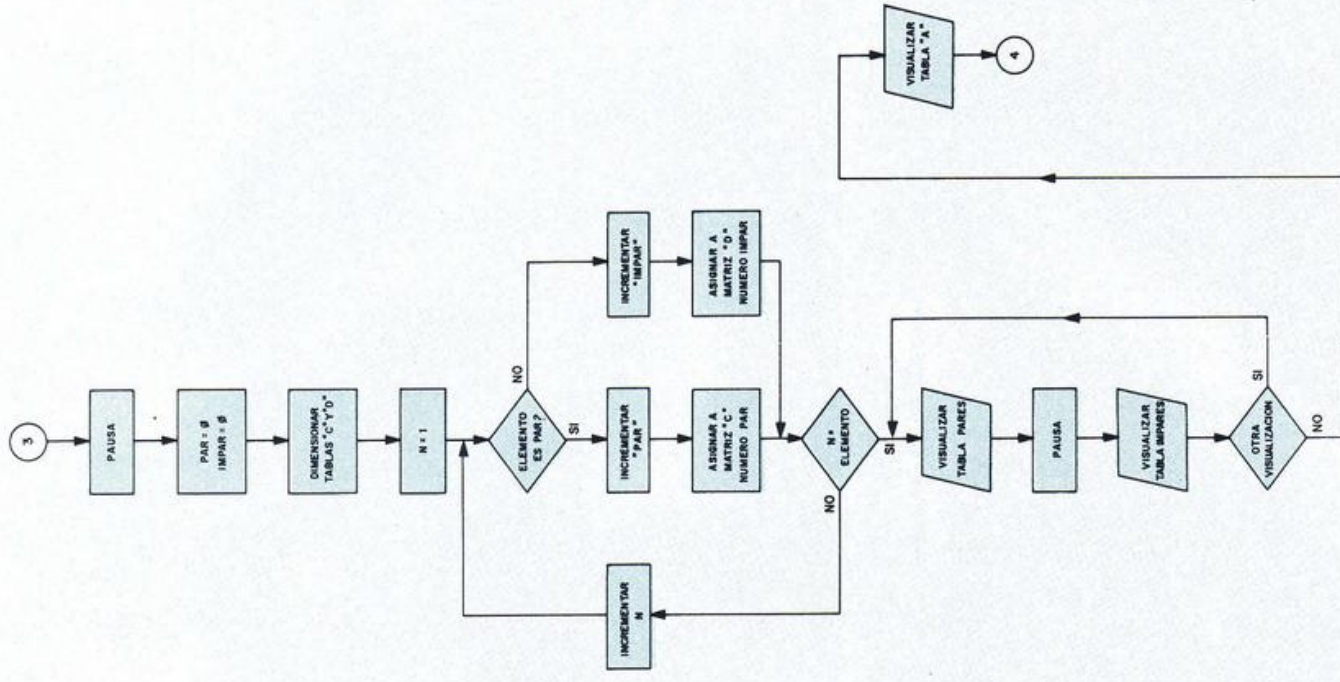
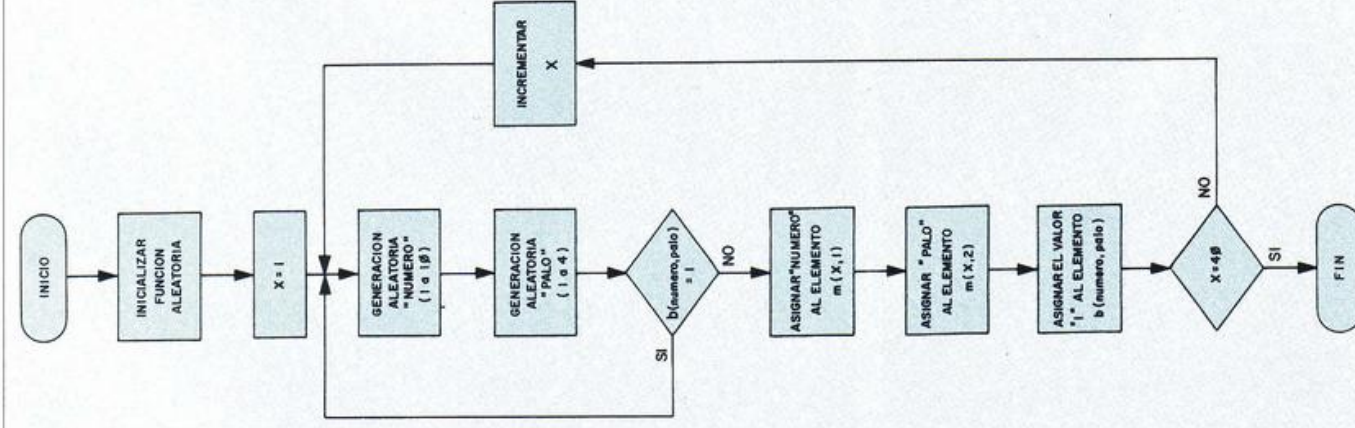
SAVE "nombre" DATA letra \$ ()

en el de cadena.

El *nombre*, entre comillas, es el que se asigna a la tabla de datos en la cinta, y la *letra* corresponde al asignado al dimensionar la matriz.

Si desea verificar o cargar una tabla, previamente salvada, sustituya el comando «SAVE» por «VERIFY» o «LOAD».

Rutina «barajar» los naipes.



Manejo tablas. PAR/IMPAR.

Con estos conceptos, interesante confeccionar un programa que maneje esta «tabla», es decir, que permita la introducción de los datos, que visualiza los datos de un programa determinado o que permita visualizarla completamente.

tener o una «A» o una «B», y el campo «TIPO» una de estas siglas «J» (juego), «U» (utilidad), o «D» (didáctico), etc.

La forma de dimensionar la matriz, suponiendo que el número máximo de programas es 100, sería:

`DIM P$(100, 22)`

¿Cómo con un dimensionado de este tipo se puede acceder a cada campo? En total, la cantidad de información que necesitamos por programa, es de 22 caracteres ($10 + 10 + 1 + 1$). La forma de acceder a cada campo, por ejemplo del elemento 1, sería la siguiente:

a) Campo «PROGRAMA».

`LET P$(1, 10) = (RANDAX)`

b) Campo «CINTA».

`LET P$(1, 11 TO 20) = (MICROHOBBY)`

c) Campo «CARA».

`LET P$(1, 21) = "A"`

d) Campo «TIPO».

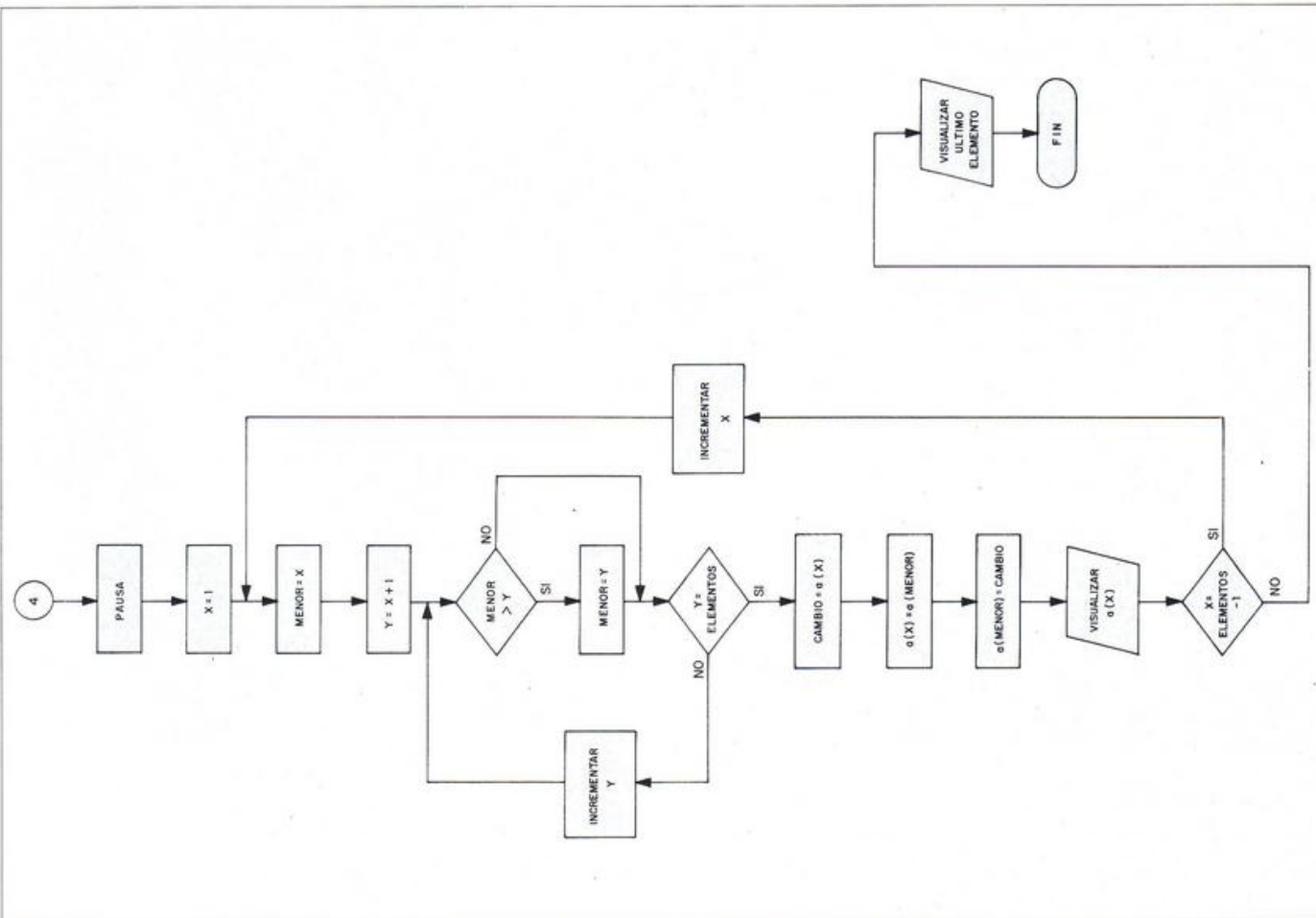
`LET P$(1, 22) = "J"`

Para visualizar el contenido del elemento 1 podríamos utilizar la sentencia:

`PRINT P$(1)`

pero una forma más racionalizada podría ser:

`PRINT "Programa : "; P$(1, 10 TO 10)
PRINT "Cinta : "; P$(1, 11 TO 20)
PRINT "Cara : "; P$(1, 21)
PRINT "Tipo : "; P$(1, 22)`




```
60 LET A$(5, 10, 4) = "TRIN"
70 PRINT A$(5)
```

c) Asignar a partir del quinto carácter la cadena "CHAR".

```
80 LET A$(5, 5, 10) = "CHAR"
90 PRINT A$(5)
```

d) Asignar al carácter número 9 el valor "E".

```
100 LET A$(5, 9) = "E"
110 PRINT A$(5)
```

La fragmentación permite utilizar tablas bidimensionales con diversos campos. Supongamos que deseamos crear una tabla para almacenar los datos referentes a nuestros programas, ésta podría tener cuatro campos:

```
- PROGRAMA
- CINTA
- CARA
- TIPO
```

En el primer campo almacenaríamos el nombre del programa; en el segundo, el nombre de la cinta donde se encuentra; en el tercero en qué cara de la cinta, y en el último, el tipo de programa, es decir, si es un juego, utilidad, etc.

La cantidad de caracteres de cada campo va a ser la siguiente:

CAMPO	CARACT.
PROGRAMA	10
CINTA	10
CARA	1
TIPO	1

El campo "CARA" es de un solo carácter ya que va a con-

Matrices de cadena

La estructura del dimensionado de matrices de cadena es el siguiente:

SENTENCIA	ARGUMENTO
DIM	letra \$ (lista de valores)

Ejemplos:

- DIM a\$(30)
- DIM x\$(2, 20)
- DIM n\$(3, 7, 5)
- DIM z\$(8, 4, 5, 10)

El nombre de una matriz de cadena está formado por una sola letra seguida del símbolo «dolar» (\$). A diferencia de las numéricas, no pueden comenzar el mismo nombre una matriz de cadena y una variable del mismo tipo.

Al dimensionar una matriz de cadena, el contenido de sus elementos queda inicializado con espacios en blanco.

Ejemplo:

```
10 DIM a$(10, 7)
20 FOR n = 1 TO 10
30 PRINT INVERSE 1; a$(n)
40 NEXT n
```

El argumento «INVERSE 1» permite visualizar los espacios, ya que intercambia el color de «tinta» por el de «papel».

Existen ciertas diferencias entre el dimensionado de matrices numéricas y el de cadenas, debido a que es preciso indicar de cuantos caracteres va a constar cada elemento.

Ejemplos:

- a) Dimensionar una matriz de un elemento de 20 caracteres.

```
DIM n$(20)
```

MICROHOBBY SEMANAL

JUEGO

LAS SIETE Y MEDIA

Por Rafael Prades

Programa 2. Carátula.

INSTRUCCIONES

Este programa está basado en el conocido juego de naipes: "Las 7 1/2"

Tienes un crédito inicial de:

10.000 pesetas

Las apuestas son de 100 ptas.

Si tienes 7 1/2 y ganas, obtienes un beneficio de 200 ptas.

Programa 2. Ilustraciones.

Las características de esta matriz son similares a las de una variable de cadena.

- b) Matriz «unidimensional» de 100 por 20 elementos, con 7 caracteres cada uno.

```
DIM J$(100, 20, 7)
```

- d) Matriz «tridimensional» formada por tres planos de 10 por 4; el máximo de caracteres por elemento es cinco.

```
DIM T$(3, 10, 4, 5)
```

- c) Matriz «bidimensional»

```
DIM n$(20)
```

Rutina asignación de naipes.

- > AS de BASTOS
- > SOTA de ESPADAS
- > TRES de COPAS
- > CABALLO de BASTOS
- > CUATRO de BASTOS

Programa 2. Juegos.

En la asignación de valores a los elementos de una matriz de cadena, no es necesario hacer referencia al último subíndice dimensionado, es decir, el que indica el número de caracteres.

```

10 DIM W$(10, 5)
20 LET W$(3) = "PEPE"
30 PRINT W$(3)

```

Quando se indica dicho subíndice, es que se hace referencia a un determinado carácter del elemento seleccionado. Siguiendo con el mismo ejemplo:

```
40 LET W$ (3, 4) = "A"
50 PRINT W$ (3, 4)
60 PRINT W$ (3)
```

La asignación de la línea 40 quiere expresar: Asignar al carácter cuatro del elemento tres, perteneciente a la matriz «W\$, el valor de cada una «A»; los demás caracteres

196 MICROBASIC

Ejemplos:
a) Rellenado con espasios.

[illegible]

Los elementos de la matriz «M\$» están dimensionados de 10 caracteres, todos los

b) Recortado.

```

10 REM *****
   RECORRADO
*****
20 G=INT(ABS(7.5)
FOR N=1 TO 7
LEND=LEN(G)
PRINT INVERSE 1;G$;INVERSE
75 PRINT INVERSE 1;$(N)
80 REM *****
      DATOS
*****
90 DATA "LINES","HARTES","INTER"
DATA "CARGO","MATERIALES","CARGO"

```

En este otro ejemplo, los elementos de la matriz «d» tienen una longitud fija de cinco caracteres, cualquier dato (días) que tenga una *mayor* longitud queda recortado a este valor.

De la misma manera que las variables de cadena, las matrices pueden fragmentarse (VER pag. 43).

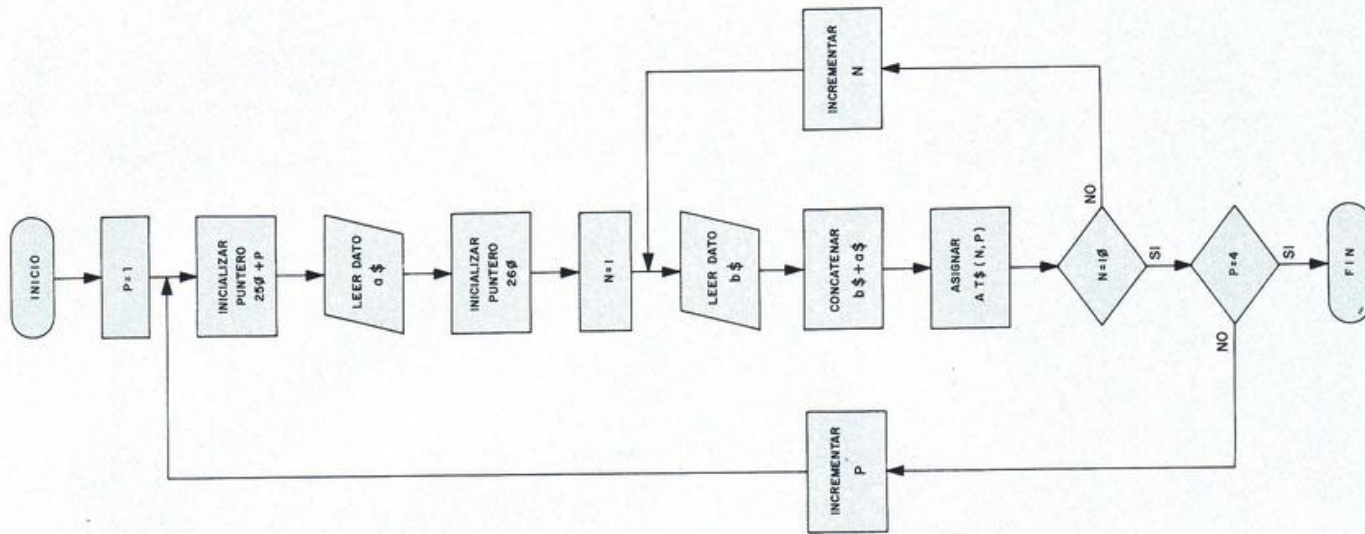
Asignemos primero un valor a un elemento de la matriz «A\$», y veamos posteriormente unos ejemplos:

```
10 DIM A$(10,10)
20 LET A$(5) = "ORDENADOR"
30 PRINT A$(5)
```

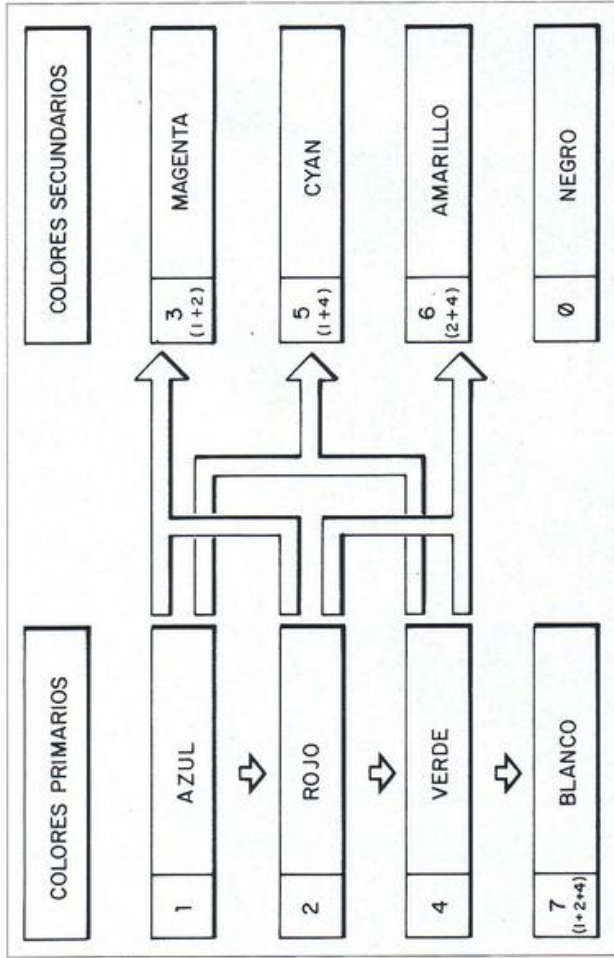
a) Asignar a los caracteres 3 a 5 el valor «POS».

40 LET A\$ (5, 3 TO 5)
5 PRINT A\$ (5)

b) Asignar a los cuatro primeros caracteres la cadena "TRIN".



Rutina asignación de naipes.



Síntesis aditiva de colores.

al ejecutarse.

Tres cosas que facilitan bastante la depuración son:

- DIAGRAMAS DE FLUJO
- LISTADOS
- USO DE SUBROUTINAS

El tener confeccionado el diagrama de flujos nos permite ver las funciones que realiza el programa paso a paso y los diversos caminos que debe seguir según se cumplan ciertas condiciones. Podemos localizar, con su ayuda,

la zona o el bloque de instrucciones donde falla el programa.

Si se dispone de una impresora, podemos obtener un listado y así localizar con mayor facilidad las sentencias que consideremos oportunas.

El uso de subrutinas facilita la depuración, ya que éstas pueden ser comprobadas individualmente, de esta manera se pueden tener ciertas garantías de que funcionan cuando se depura el programa final.

Es importante tener localizada la zona a partir de la cual se desea comenzar la depuración, por ejemplo, en el siguiente programa, que tiene dos errores, empezáramos a depurar a partir de la línea 40, ya que si lo ejecuta observará que se visualizan las consignes de cadena de las líneas 20 y 30.

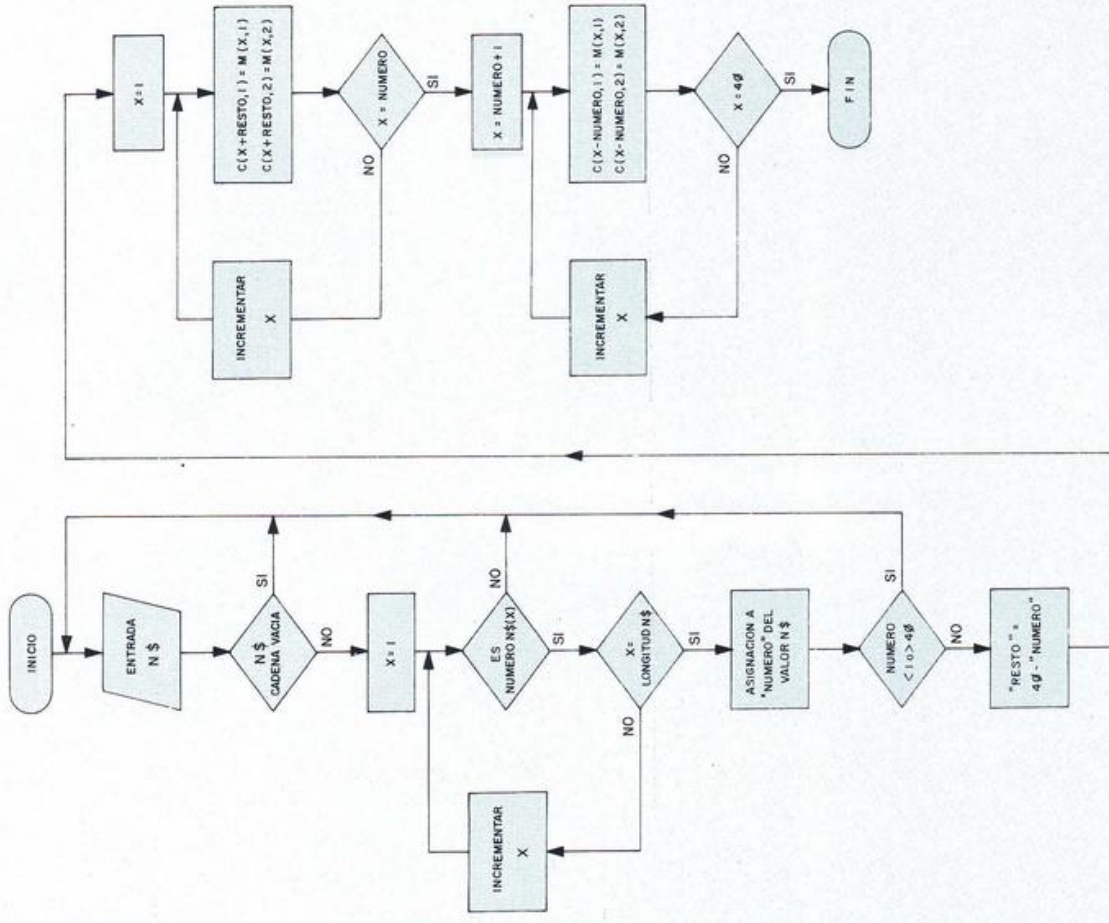
Para corregir el ejemplo, deberá modificar la línea 60

```
60 PRINT AT n + 2,0; a$
```

PROGRAMA 1

```
10 REM *****
   * ERRORES *
   * *****
20 PRINT "ARTICULO"; "PRECIO";
30 PRINT "_____"; "_____";
```

```
40 FOR n=1 TO 5
50 INPUT "Artículo > "; a$
60 PRINT AT 3,0; a$ " "; p
70 INPUT "Precio > "; p
80 NEXT n
90 PRINT AT n+2,15; p
```



Rutina «cortar» el mazo.

Programa

El programa número «2» permite jugar con el ordenador, al conocido juego de cartas: «Las Siete y Media».

El ordenador hace el papel de «banca» y por tanto, baraja y reparte las cartas; nosotros tenemos la posibilidad de indicar por qué número deseamos «cortar» la baraja.

Después de repartir las dos primeras cartas, una para él y otra para nosotros, nos irá preguntando sucesivamente si deseamos otra carta o no.

Al final de la partida se presentan los tantos obtenidos por cada uno de los dos jugadores y si deseamos jugar otra «mano».

Para dar mayor emoción a la partida, al principio de ésta, tenemos un crédito de «10.000» ptas. Las apuestas son de «100» ptas., ganamos el doble si obtenemos «siete y media» y el ordenador no, ya que en igualdad de puntos gana la banca.

¡BUENA SUERTE!

Hay una serie de matrices utilizadas en la confección del programa, que se encuentran dimensionadas en las líneas 90 a 125.

La matriz «T\$» almacena el nombre de cada carta (as deoros, dos de..., etc.). Es una tabla bidimensional de «10» por «4», ya que la baraja consta de 4 «palos» de diez cartas. La tercera dimensión (20) indica el número máximo de caracteres de cada elemento, y se encuentra ligeramente sobredimensionada ya que con dieciocho hubiera bastado, por que el contenido de mayor longitud va a ser «caballo de espadas» (16 letras + 2 espacios).

10	BLOQUE 1
20	
30	
40 STOP	
50	BLOQUE 2
60	
70	
80 STOP	
90	BLOQUE 3
100	
110 STOP	
120	
130	BLOQUE 4
140	
150 STOP	

«Depurado por partes».

Los blucles anidados «p» y «n» (líneas 140-200 y 160-190) son los encargados de asignar a la matriz «T\$» los valores de cadena especificados en las sentencias «DATA» (líneas 251 a 260).

En los juegos de naipes, cada carta tiene un valor relativo distinto; en el caso de las siete y media, las cartas del «as» al «siete» tienen un valor

idéntico al que figura en una de sus esquinas. Por el contrario, las «figuras» (sota, caballo y rey) tienen un valor de medio punto. Con estos requisitos, los blucles 310-330 y 340-360 asignan a los elementos de la matriz «V» el valor correspondiente.

«V» es una matriz unidimensional de diez elementos, uno por cada valor de carta. La re-

AZUL (1)

CYAN (5)

MAGENTA (3)

BLANCO (7)

VERDE (4)

ROJO (2)

AMARILLO (6)

Estrella de colores.

«mes» era el nombre de una variable, solucionaremos el problema asignando a esta un valor en la zona del programa que corresponda, por ejemplo:

25 LET mes = 8

para encontrar el error, ya que indica el tipo y la línea en que lo ha detectado (ver pag. 96). El número de línea no siempre significa que el error se encuentre en la misma, sino que está relacionado con ella, por ejemplo, un error con mensaje:

2 Variable not found, 70:1

suponiendo que la línea 70 fuera:

70 PRINT mes

significa que el argumento de «PRINT» al no ir entrecomillado, lo interpreta como variable y no encuentra su asignación inicial. Si efectivamente

Depuración

La tarea de depurar no es muy complicada, pero si laboriosa, aunque a veces basta con echar una ojeada al listado, para descubrir el error.

La complejidad de la depuración depende del número de bifurcaciones que tenga el programa, ya que puede ser bastante elevado el número de caminos distintos que siga

que nuestro programa no realice las funciones que en un principio teníamos previstas. Al corregir este tipo de errores, de estructura, están concebidas las técnicas de depuración.

DEPURACION DE PROGRAMAS

En capítulos anteriores se ha ido explicando la mayor parte del juego de sentencias del Spectrum, así como una serie de conceptos: Bucles, Subrutinas, Funciones, Matrices, etc. Con todo este material el iniciado en programación puede ya confeccionar un volumen importante de programas; por este motivo es necesario que conozca una serie de técnicas destinadas a la depuración o puesta a punto de programas; también son conocidas por el término inglés *debugging*.

Errores

Es muy difícil que un programa, medianamente complicado, funcione a la primera, ya que hay muchos factores que deben tenerse en cuenta.

Los tipos de error que surgen al elaborar un programa, pueden clasificarse, principalmente, en tres apartados:

- SINTACTICOS
- DE DEFINICION
- DE ESTRUCTURA

Los errores *sintácticos* son aquellos que se producen al teclear un argumento no compatible con una sentencia determinada, por ejemplo:

```

- NEW 32
- LET "A$" = "abc"
- CLS fin
- PRINT $
    
```



Zonas de la pantalla.

Este tipo de errores son detectados inmediatamente por el intérprete BASIC, al pulsar «ENTER». Una interrogación parpadeante se posiciona en las proximidades del error, indicando que no entiende dicha sentencia.

Revise la línea introducida; si no encuentra el fallo, consulte, en el capítulo correspondiente del libro, las posibles estructuras que puede adoptar dicha sentencia.

Como puede observar, los errores *sintácticos* son detectados y corregidos en la

fase de edición de programas, a diferencia de otros ordenadores en los que se realiza posteriormente.

Otro tipo de errores se manifiestan al ejecutarse el programa, presentando el correspondiente mensaje de error. Estos errores son originados principalmente por no haber definido previamente una variable, por utilizar un subíndice fuera de rango, por utilizar un parámetro erróneo en una función, etc.

El mensaje presentado sirve únicamente de referencia

lación entre el subíndice y la carta es el siguiente:

- 1 - AS
- 2 - DOS
- 3 - TRES
- 4 - CUATRO
- 5 - CINCO
- 6 - SEIS
- 7 - SIETE
- 8 - OCHO
- 9 - CABALLO
- 10 - REY

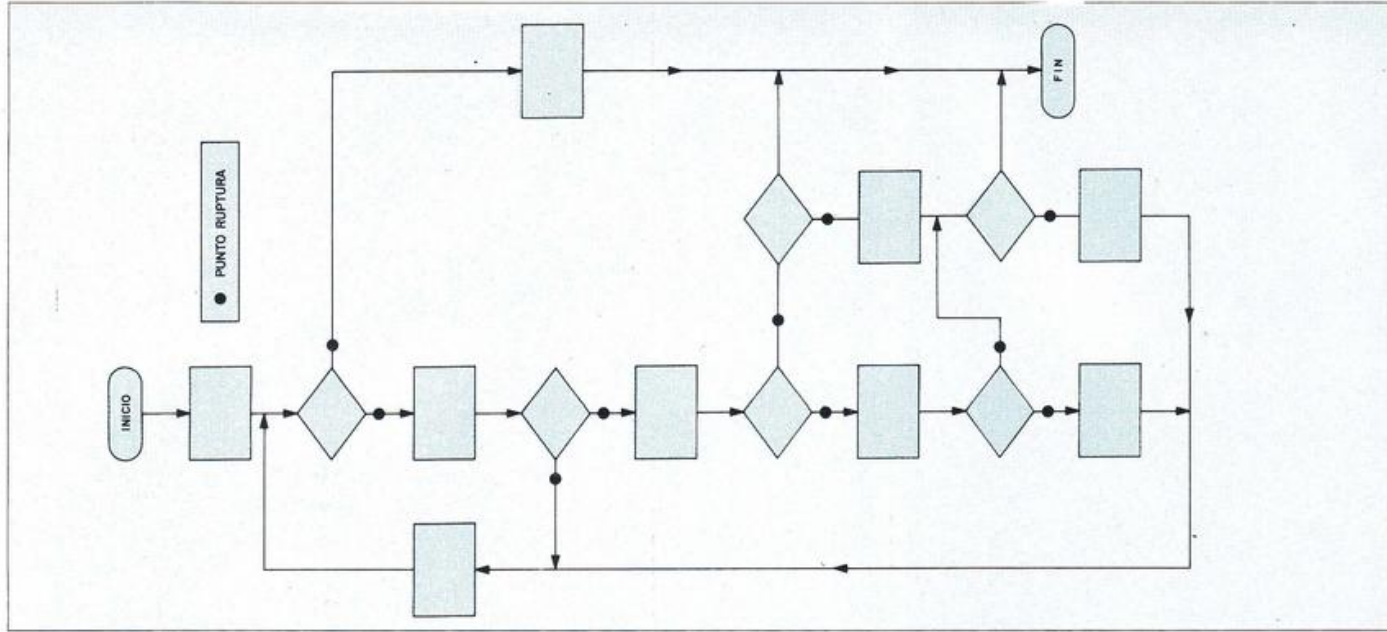
Para barajar las cartas se utiliza la función pseudoaleatoria «RND». La subrutina localizada en la línea 490, reordena los valores aleatorios asignados a las variables «número» y «palo». La variable «número» tiene un valor comprendido entre «1» y «10»; la correspondencia, con los números de carta, es idéntica a la utilizada por la matriz «V». La variable «palo» puede tener valores entre «1» y «4»; la relación entre dicho valor y el palo de la carta es:

- 1 - OROS
- 2 - COPAS
- 3 - ESPADAS
- 4 - BASTOS

Las cartas generadas aleatoriamente son asignadas sucesivamente a los elementos de la matriz «M». La primera dimensión indica la posición de la carta dentro de la baraja, y la segunda, el número y el palo.

Al utilizar una función aleatoria pueden repetirse varios valores; para que la matriz «M» no contenga varias cartas idénticas, se ha utilizado la matriz «B». Esta matriz bidimensional de 10 por 4 está inicializada en un principio a cero.

Los valores proporcionan-



Puntos estratégicos de ruptura.

dos por la subrutina de generación de cartas aleatorias son utilizados como subíndices de la matriz «B»; si el elemento direccionado tiene valor «0», es que la carta no se ha generado anteriormente; si, por el contrario, tiene valor «1», indica que la carta se repite. En el primer caso, los valores de las variables «número» y «palo» se asignan al elemento correspondiente de la matriz «M» y el valor «1» al elemento direccionado de la matriz «B». En el segundo caso, no se asigna ningún valor a la matriz «M» y el programa genera dos nuevos valores que vuelven a ser comprobados.

La rutina encargada de «cortar» la baraja se encuentra localizada en las líneas 610 a 675. En la matriz «C» quedan las cartas ordenadas y listas para poder comenzar la partida.

La transferencia de cartas entre la matriz «M» y «C» se realiza en dos fases. En la primera, se transfieren los elementos situados en las primeras posiciones, es decir, desde el número uno hasta el direccionado por la variable «número»; en la segunda, se realiza la transferencia de las restantes. En la matriz «C» estos dos bloques quedan intercambiados.

La estructura general del programa es:

- 10 : Comentario con el nombre del programa.
- 15 : Asignación del color azul para el borde, verde para el fondo y negro para los caracteres.
- 20-26 : Carátula del programa.
- 40-75 : Asignación de valores a las variables «crédito» y «manos». Esta última contabiliza el número de partidas jugadas.

- 90-125 : Dimensionado de las matrices explicadas anteriormente.
- 140-200 : Bloque de asignación de los nombres de las cartas a la matriz «I».
- 251-260 : Datos relativos a los nombres de las cartas.
- 310-360 : Bucles para la asignación de valores.
- 365 : Llamada a la subrutina de visualización de instrucciones.
- 372 : Inicialización de la función aleatoria.
- 380-400 : Mensaje de espera.
- 410-470 : Asignación aleatoria de valores, de carta, a la matriz «M».
- 490-520 : Generación aleatoria de número y palo.
- 540-570 : Introducción del número de carta por donde se desea cortar la baraja.
- 572 : Comprobación de que el valor introducido no es una cadena vacía.
- 574-578 : Bucle para detectar si hay algún valor no numérico dentro de la variable «N\$».
- 580 : Evaluación de la cadena «N\$».
- 590 : Comprobación de que el valor está dentro de los márgenes (1) a «40».
- 592-594 : Mensaje de espera.
- 600 : Cálculo del número de cartas que hay desde la cortada hasta el final.
- 610-675 : Transferencia de valores entre las matrices «M» y «C» (cortar la baraja).
- 680-690 : Pantalla de ánimo.
- 700-715 : Temporización hasta que se pulsa una tecla y borrado de la pantalla.
- 722 : Invitación a jugar.
- 730 : Inicialización de los tantos del jugador.
- 740 : Asignación de la primera carta del jugador.
- 760 : Llamada a la subrutina de presentación de cartas.
- 770 : Asignación de los tantos

- 1220 : Inicialización de los tantos del ordenador.
- 1230 : Asigna la segunda carta al ordenador.
- 1240 : Indicación de que juega el ordenador.
- 1252 : Coordinada «Y» de la siguiente carta a visualizar.
- 1260 : Llamada a la subrutina de presentación de cartas.
- 1270 : Asignación de los tantos del ordenador.
- 1280 : Asignación de la siguiente carta a escoger.
- 1300 : Comprueba si el ordenador se ha «pasado».
- 1310 : Comprueba si el ordenador ha obtenido $(7/2)$.
- 1320 : Comprueba si no se ha pasado.
- 1330 : Asignación del valor (1) a la variable «ganador». El ordenador pierde.
- 1340-1360 : Mensaje de presentación de los tantos del ordenador, en el caso de «pasarse».
- 1370-1380 : Temporiza hasta que se pulsa una tecla.
- 1410 : Asignación del valor «B» a la variable «ganador». El ordenador gana.
- 1420 : Mensaje de visualización $(7/2)$, por parte del ordenador.
- 1430-1432 : Temporiza hasta que se pulsa una tecla.
- 1460 : Comprueba si el jugador ha obtenido $(7/2)$, para seguir pidiendo cartas hasta que gane o se pase.

- 1470 : Si el jugador se ha quedado, pide cartas mientras que él tenga menos de seis puntos.
- 1490 : Mensaje pidiendo otra carta.
- 1492 : Duración del mensaje.
- 1494 : Borrado del mensaje.
- 1500 : Temporización entre dos jugadas.
- 1510 : Asignación de la siguiente carta.
- 1522 : Incrementar, en dos, la siguiente posición.
- 1530 : Llamada a la subrutina de presentación de cartas.
- 1540 : Incrementar los tantos del ordenador.
- 1570-1590 : Mensaje de presentación de los tantos del ordenador, en el caso de quedar «SE».
- 1592-1594 : Temporiza hasta que se pulsa una tecla.
- 1610 : Si el ordenador tiene más puntos o igual que el jugador, el ordenador gana.
- 1620 : Si tiene menos, el ordenador pierde.
- 1640 : Mensaje de que el ordenador gana.
- 1650 : Mensaje de que el jugador gana.
- 1660 : Visualización de los tantos del ordenador.
- 1680 : Si el jugador se pasó, el ordenador no indica cuál era su carta.
- 1690 : Visualización de los tantos del ordenador.
- 1700-1710 : Temporización hasta que se pulsa una tecla.
- 1740 : Si el jugador gana y tiene $(7/2)$, la apuesta se duplica.
- 1760-1770 : Cálculo de los nuevos créditos si gana el ordenador.
- 1790-1800 : Idem. si gana el ordenador.
- 1820 : Visualización de las manos jugadas hasta ese momento.
- 1830 : Visualización del nuevo crédito del jugador.
- 1840 : Idem. del ordenador.
- 1850 : Comprueba si el ordenador se ha quedado sin crédito.
- 1860 : Idem. del ordenador.
- 1880 : Mensaje de invitación a jugar otra mano.
- 1890-1930 : Comprobación de la respuesta elegida.
- 1940 : Borrado del mensaje de la zona inferior.
- 1945 : Incremento de las manos jugadas.
- 1950-1990 : Inicialización a cero de los elementos de la matriz «B»; permite volver a barajar las cartas.
- 2000 : Salto a la rutina de barajar.
- 2100 : Comienzo de la subrutina «INSTRUCCIONES».
- 2110 : Invitación a visualizar las instrucciones.
- 2120-2160 : Comprobación de la operación elegida.
- 2170-2210 : Visualización de las instrucciones.
- 2220-2230 : Pausa hasta que se pulsa una tecla.


```

10 REM *****
  * BORDE/PAPEL/TINTA *
  * *****
30 FOR X=0 TO 7
40 FOR Y=0 TO 7
50 BORDER X
60 INK Y
70 NEXT Y
80 PRINT AT 10,7;"MICROHOBBY 5"
90 PRINT AT 10,7;"PULSE UNA TECLA"
100 GOTO 30
110 GOTO 30
120 NEXT X
130 NEXT Y
140 INK 0
150 INK 0

```

Los colores, por defecto, que presenta el Spectrum al conectarlo, son:

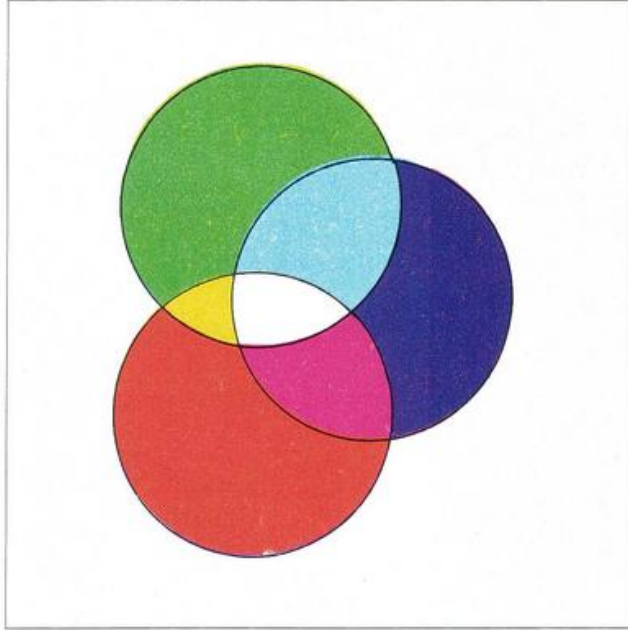
Borde blanco
Papel blanco
Tinta negra

Atributos permanentes y temporales

Como se comentó en la página 71, la zona de representación, destinada al usuario, está constituida por 22 líneas y 32, esto hace un total de «704» caracteres. Cada posición, de estos caracteres, posee unos *atributos* que determinan, entre otras cosas, el color del «papel» y el de «tinta»; estos atributos son fijados mediante ciertos parámetros que pueden ser:

— PERMANENTES
— TEMPORALES

Los atributos *permanentes* son fijados con las sentencias «BORDER», «PAPER», «INK», etc. éstos permanecen en la memoria de *presentación visual* hasta que se vuelven a modificar, de manera que todas las sentencias que tengan que ver con la visualización de caracteres o gráficos lo harán en los colores a tinta y papel especificados. Ejemplo:



Demostración síntesis aditiva

```

10 REM *****
  * PERMANENTES *
  * *****
12 LET azul=1: LET rojo=2: LET
verde=4
14 LET mensaje=5: LET cyan=5
16 LET negro=0: LET blanco=7
18 INPUT "Color borde ";bord
20 INPUT "Color fondo ";fond
22 INPUT "Color tinta ";tint
24 BORDER borde
26 PAPER fond
28 INK tint
30 INPUT "Mensaje ";msg
100 PRINT msg
110 GOTO 30

```

PRINT PAPER 2: "HOLA"

que visualiza la cadena alfanumérica "HOLA" sobre fondo verde, independientemente de los atributos permanentes que tuviera. Veamos otro ejemplo aclaratorio:

```

10 REM *****
  * TEMPORALES *
  * *****
12 LET azul=1: LET rojo=2: LET
verde=4
14 LET mensaje=5: LET cyan=5
16 LET negro=0: LET blanco=7
18 INPUT "Color borde ";bord
20 INPUT "Color fondo ";fond
22 INPUT "Color tinta ";tint
24 BORDER borde
26 PAPER fond
28 INK tint
30 INPUT "Mensaje 1 ";msg
100 PRINT msg
110 INPUT "Color de mensaje 2 ";
120 INPUT "Mensaje 2 ";msg
130 PRINT INK temporal,msg
140 GOTO 30

```

Los atributos *temporales* se utilizan para visualizar uno o varios caracteres con unos atributos distintos de los especificados.

Las sentencias «PRINT», «INPUT», etc. deben ir acompañadas en este caso, de las conocidas «PAPER» e «INK». Ejemplo:

e introducir la línea 90 dentro del bucle, asignándole un nuevo número de línea comprendido entre 71 y 79.

Lamentablemente el Spectrum no dispone de dos sentencias muy potentes utilizadas en otros ordenadores para depurar. Estas son:

— TRON
— TROFF

que permiten habilitar o deshabilitar la facilidad de traza, con ella se puede visualizar en pantalla la secuencia de instrucciones ejecutadas por el ordenador, paso a paso y, comprobar si es correcta.

El Spectrum utiliza para depurar las sentencias «STOP» y «CONTINUE», utilizando la primera como *breakpoint* o punto de ruptura.

STOP y CONTINUE

La utilización de estas sentencias es bastante sencilla, pero si tiene alguna duda consulte las páginas 91 a 96.

Básicamente la depuración

con estas sentencias consiste en colocar en lugares estratégicos del programa, diversos puntos de ruptura con la sentencia «STOP».

Al ejecutarse deberá pararse en el primer punto, visualizando el manejo correspondiente, si no lo encuentra es que el fallo está localizado entre la línea de arranque y dicho «STOP». Si por el contrario se para, podremos mediante comandos directos conocer el valor de las variables utilizadas; si su contenido es correcto introduciremos también como comando directo la sentencia «CONTINUE» y el programa continuará su ejecución hasta el próximo punto de ruptura, donde haremos las mismas operaciones.

De esta manera iremos ejecutando por partes el programa hasta que localicemos el fallo.

Algunos de los puntos estratégicos para la colocación de los «STOP» son las bifurcaciones, es decir, donde el programa pregunta si se cumple una condición im-

puesta en un «IF ... THEN ...». Una vez parado un programa si se desea continuar con su ejecución, en una línea de terminada, debe utilizar «GO TO» o «GO SUB», si se trata de una subrutina, ya que si utiliza «RUN n» todas las variables que tenía definidas se borran y posiblemente aparezcan el error:

2 Variable not found

por que la definición se encontraba en las líneas anteriores.

Cuando el programa esté corregido deben suprimirse la líneas con las sentencias «STOP», utilizadas como puntos de ruptura.

Programa «Depurador»

Como alternativa al uso «STOP» y «CONTINUE» el programa «1» realiza las mismas funciones, pero tiene la ventaja de una mayor facilidad de uso.

El programa que usted quiera depurar deberá estar

```

150 REM "CONTINUE"
160 PRINT #0;"Desea continuar ("
$ /N)";
170 PAUSE 0:INKEY$
180 IF LEN$=" " OR LEN$="s" THEN GO
TO 170
190 IF LEN$="n" OR LEN$="n" THEN CL
S: PAUSE 0: STOP
200 GO TO 170
210 REM LOGO
1000 RESTORE
1010 FOR N=1 TO 32
1020 PRINT INVERSE 1;AT 0,N;CHR$(
DATA 32,77,73,99,82,79,72,7
9,65,66,89,32,63,66,77,55,7,6,5,
7,32,32,32,32,32,32,32,32,32,32
1040 NEXT N
1050 DATA 32,77,73,99,82,79,72,7
9,65,66,89,32,63,66,77,55,7,6,5,
7,32,32,32,32,32,32,32,32,32,32
1060 RETURN AT 7,0;"Cadenas #1+ y
#2+ distintas"
1100 RETURN AT 7,0;"Cadenas #1+ y
#2+ distintas"
1210 RETURN

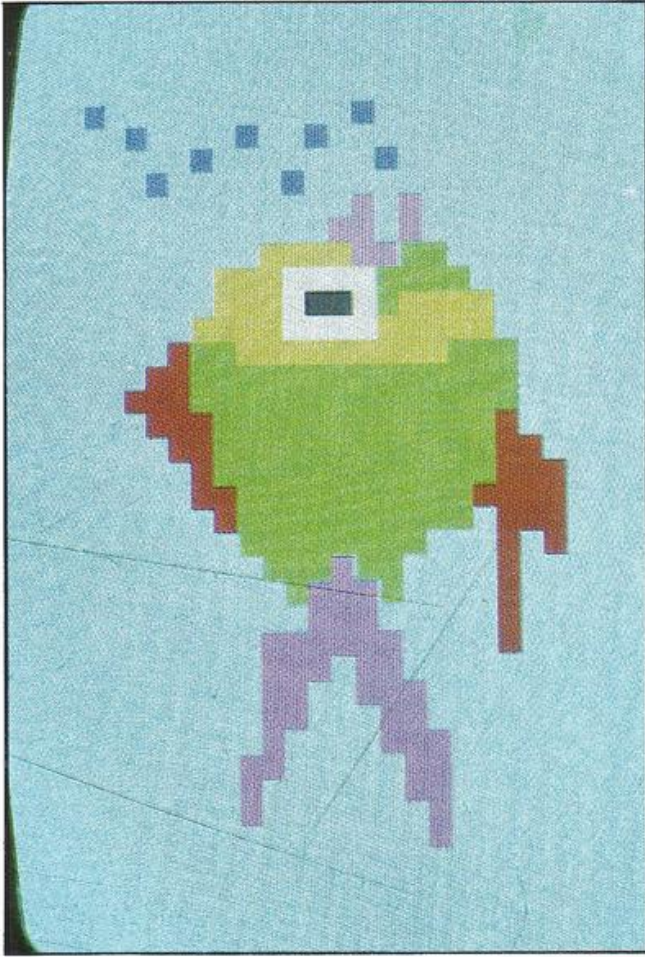
```

PROGRAMA 2

```

10 REM *****
  * CURSO/BASIC *
  * *****
  * 11 ERRORES *
  * *****
20 BORDER 4: PAPER 4: INK 0: C
L5
30 REM "VARIABLES": LET logo=1
000
35 LET igual=1100
40 GO TO logo
50 INPUT "Cadena 1 ";a$
60 IF a$="" THEN GO TO 1000
70 IF LEN a$ > 20 THEN GO TO 50
80 PRINT AT 3,0;"Cadena 1: ";a
$
90 INPUT "Cadena 2 ";b$
100 IF b$="" THEN GO TO 90
110 INPUT "Cadena 2 ";b$
120 PRINT AT 10,0;"Cadena 2: ";
b$
130 IF a$=b$ THEN GO SUB igual
140 GO SUB distinto

```

Las posibilidades del color

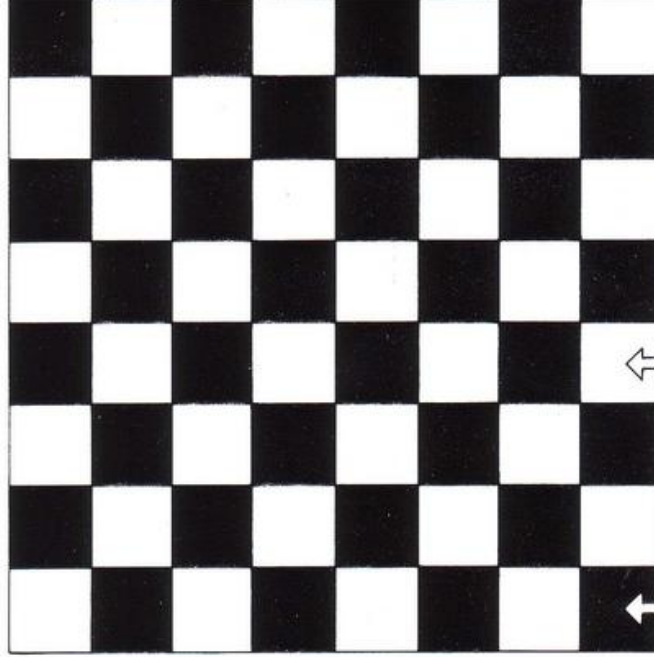
COLOR

El color es actualmente una de las características a tener en cuenta cuando se adquiere un ordenador personal, principalmente si lo piensa destinar a programas de juegos, aunque en otro tipo de aplicaciones tampoco está de más añadir algo de color, bien sea por estética o para destacar algún mensaje o zona de pantalla.

El «ZX Spectrum», hermano mayor del conocido «ZX-81», debe precisamente su nombre a la capacidad que tiene

GRAFICO DEFINIDO POR USUARIO "A"

CODIGO DECIMAL



FONDO (PAPER) ROJO
TINTA (INK) AMARILLA

85
170
85
170
85
170
85
170

Acceso al teclado

BIN



MODO E

BRIGHT
Definición

Con este comando puede alterarse el color del borde. La estructura general es:

SENTENCIA	ARGUMENTO
BORDER	código color

Ejemplos:

- BORDER 2 (rojo)
- BORDER 3 (magenta)
- BORDER 0 (negro)
- BORDER 7 (blanco)

Zonas de pantalla

Existen dos zonas que pueden ser modificadas por los comandos de control de color:

- BORDE
- PAPEL

El **borde**, como su propio nombre indica, es la zona formada por el margen exterior de la pantalla del televisor, en este no se pueden imprimir caracteres.

El **papel** o fondo es el rectángulo central, rodeado por el «borde», donde se imprimen los caracteres o gráficos.

Las siguientes instrucciones muestran estas zonas:

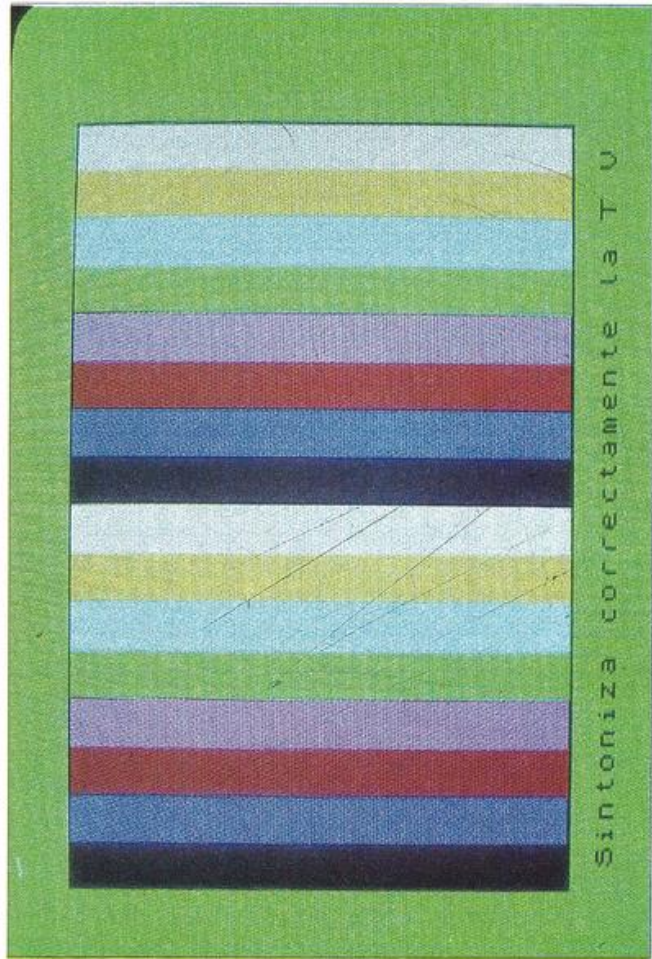
Comandos

BORDER

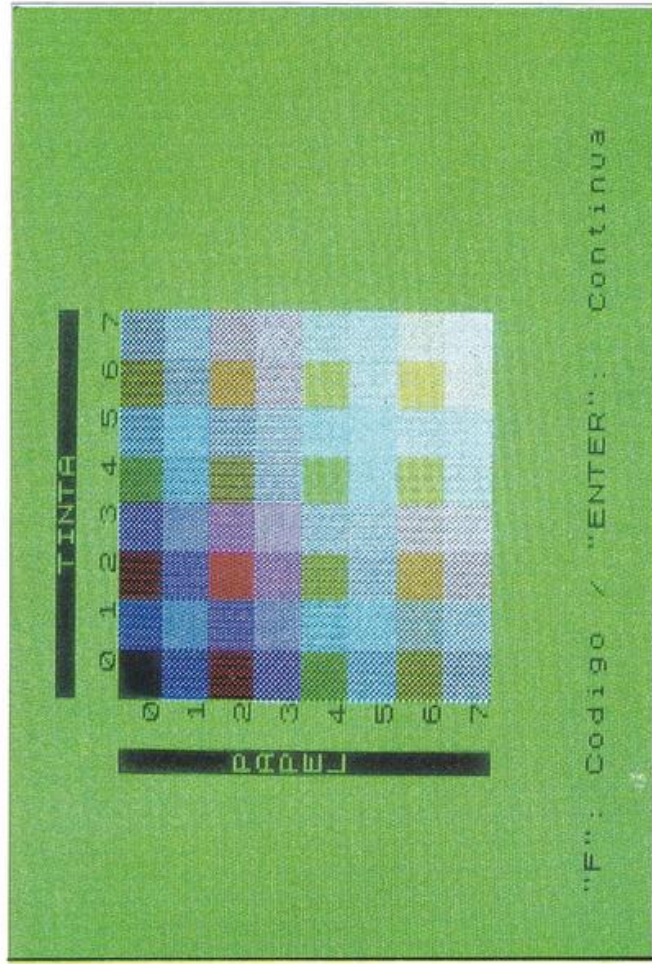
El mensaje «BORDE rojo» está visualizado en la línea 23 (zona reservada), ésta al igual que la zona 22, toman el color del «borde».

El color en que visualizamos los caracteres se denomina **tinta**.

Rejilla utilizada en la simulación de colores (Ej. naranja)



Carta de ajuste



«64» Colores

para generar los colores del espectro luminoso.

El Spectrum dispone de ocho colores que pueden conseguirse en dos gamas de brillo; cada uno tiene asignado un número que lo identifica a la hora de programar.

CODIGO	COLOR
0	NEGRO
1	AZUL
2	ROJO
3	MAGENTA
4	VERDE
5	CYAN
6	AMARILLO
7	BLANCO

Este código está en función de la luminosidad del color, así el negro o ausencia de luz, tiene el código cero y según va progresando la luminosidad aumenta el valor hasta llegar al blanco que tiene el

código siete. La luminosidad de los colores es difícil de apreciar ya que el ojo humano no es más sensible a ciertos colores, los cuales, nos parecen más claros.

El programa n.º «1» muestra unas barras verticales con diversos colores, éstos están ordenados según su código. Si dispone de un televisor en blanco y negro podrá comprobar la luminosidad de cada color ya que cada uno toma un valor distinto de gris; el resultado es una escala degradada de grises. Obtendrá el mismo efecto en un televisor de color, si éste es anulado, el programa también sirve para poder sintonizar correctamente el televisor.

Teoría del color

Antes de explicar la utiliza-

ción de los comandos que afectan sobre el color de las diversas zonas de la pantalla, es conveniente tener algunos conocimientos sobre la teoría del color.

La descomposición de la luz blanca, por un prisma, se reduce básicamente a tres colores:

- AZUL
- ROJO
- VERDE

éstos son llamados «primarios». A partir de éstos, y por *síntesis aditiva*, pueden obtenerse los llamados «secundarios».

- MAGENTA
- CYAN
- AMARILLO

Se denominan colores

complementarios aquéllos que al mezclarse, por *síntesis aditiva*, dan origen, de nuevo, a la luz blanca. En la estrella de los colores, representada en la figura, pueden observarse los colores complementarios, ya que éstos se encuentran enfrentados:

AZUL - AMARILLO
ROJO - CYAN
VERDE - MAGENTA

Lógicamente, al recomponer los tres colores primarios se obtiene también la luz blanca.

Síntesis aditiva

Esta se demuestra con la utilización de tres proyectos de luz blanca, en el primero insertamos un filtro de color

rojo, en el segundo uno verde y en el tercero uno azul.

Los proyectores se colocan de manera que sus haces coincidan según la figura adjunta.

Se comprueba que en el área iluminada por los tres proyectores se obtiene luz blanca; en el área de intersección de la luz roja con la verde, el amarillo; en el caso de la luz roja y del azul, resulta el magenta y por último, de la mezcla de la luz verde y azul, de cyan.

ROJO + VERDE + AZUL	BLANCO
ROJO + VERDE	AMARILLO
ROJO + AZUL	MAGENTA
VERDE + AZUL	CYAN

Comprobemos que en el Spectrum se cumple la *síntesis aditiva*, sumando los respectivos códigos de color.

a) Blanco:

Azul	1
Rojo	2
Verde	4
Blanco	7

b) Magenta:

Azul	1
Rojo	2
Magenta	3

c) Cyan:

Azul	1
Verde	4
Cyan	3

d) Amarillo:

Rojo	2
Verde	4
Amarillo	6

ceder a cualquiera de estas combinaciones.

Entre las aplicaciones de acceso directo, puede destacar la utilidad que tiene el editar un programa con diversos colores, para resaltar u ocultar alguna de sus zonas.

El acceso directo a las funciones de VIDEO fue revisado en la página 70.

Errores

Si se especifica un argumento erróneo en alguna de estas sentencias, se producen los siguientes errores:

- a) Color no válido.

X Invalid colour

Ejemplos:

BORDER 8
PAPER 10
INVERSE 2
FLASH 9

PROGRAMA 4

```

10 REM *****
* ATRIBUTOS *
*****
20 BORDER 4: PAPER 4: INK 0: C
25 REM ENTRADA PRINCIPAL
30 INPUT "A(tributo) = ", LINE a
$
40 IF a$="" THEN GO TO 30
50 FOR n=1 TO LEN a$
60 IF a$(n) < "0" OR a$(n) > "9" T
HEN GO TO 30
70 NEXT n
80 LET atributo=VAL a$
90 IF atributo<0 OR atributo>2
55 THEN GO TO 30
100 PRINT atributo," CODIGO DECIMA
102 REM CONVUELTO BINARIO
104 LET numco=atributo
106 LET b$=""
110 DIM i(8)
120 FOR n=7 TO 0 STEP -1
130 IF numero=INT (2^n): LET
i(n+1)=1
140 LET b$=b$STR$ i(n+1)
150 NEXT i
160 PRINT "CODIGO BINARIO ... "
170 REM IDENTIFICACION
180 IF b$(1)="1" THEN PRINT FLA
SH TO 200
190 IF b$(1)="0" THEN PRINT FLA
SH TO 200

```

224 MICROBASIC

CODIGO	DECIMAL	...	86
--------	---------	-----	----

CODIGO BINARIO ... 01010110

PARPADEO DESACTIVADO

BRILLO ACTIVADO

PAPEL ... 2

TINTA ... 6

Ejemplo programa 4.

b) Entero fuera de rango.

6 Integer out of range

Ocorre cuando el código de color es inferior a «0» o superior a «255».

Ejemplos:

INK -
BRIGHT

Una instrucción del tipo «BORDER 3,6» no es errónea ya que el argumento queda redondeado a «4», obteniéndose por tanto un borde de color verde. ■

Una instrucción del tipo «BORDER 3,6» no es errónea ya que el argumento queda redondeado a «4», obteniéndose por tanto un borde de color verde. ■

```

190 PRINT "PARPADEO DESACTIVADO"
200 IF b$(2)=""1" THEN PRINT BRI
GOTO 1
210 PRINT "BRILLO ACTIVADO" GOTO 1
220 PRINT "BRILLO DESACTIVADO"
212 LET pape1=0
214 LET peso=0
220 FOR n=5 TO 3 STEP -1
230 LET valor=VAL b$(n)
240 IF valor=1 THEN LET pape1=p
ape1+2*peso
244 LET peso=peso+1
250 NEXT n
260 PRINT "PAPEL, pape1; INK 9;"P
AGE
270 LET tinta=0
280 LET peso=0
290 FOR n=8 TO 6 STEP -1
300 LET valor=VAL b$(n)
310 IF valor=1 THEN LET tinta=t
inta+2*peso
320 LET peso=peso+1
330 NEXT n
340 PRINT "PAPEL 9; INK tinta;"T
INTA
350 REM PARPADEO
360 PRINT "0", otro atributo (S/
362 PAUSE 0
370 IF z$=""N" OR z$=""N" THEN ST
OP
380 IF z$=""s" OR z$=""s" THEN GO
TO 20
390 GO TO 362

```

Resolución del color

Cada carácter está formado por una matriz de ocho por ocho puntos, también conocidos por el término inglés «pixel»; por tanto hay 64. A pesar de que podemos activar individualmente cada punto, con las sentencias utilizadas en la realización de gráficos en alta resolución, no se puede, sin embargo, asignar un color de tinta distinto para cada uno de ellos, ya que cada carácter está controlado por unos atributos, bien permanentes o temporales. Estos afectan a cada matriz de 64 pixels, por tanto sólo puede haber dos colores distintos en cada posición de carácter. Esta configuración se denomina «color en baja resolución».

Transparencia y contraste

Las sentencias «PAPER» e «INK» pueden tener como argumento los códigos de color 8 y 9, teniendo un significado de *transparencia* y *contraste* respectivamente.

La transparencia consiste en conservar los atributos temporales de la pantalla al imprimir un nuevo carácter.

Ejemplos:

a) Papel transparente:

a) Papel transparente:

L 0 RCH

PAPER "B"

L 12 BORDER & PAPER C INK "C"

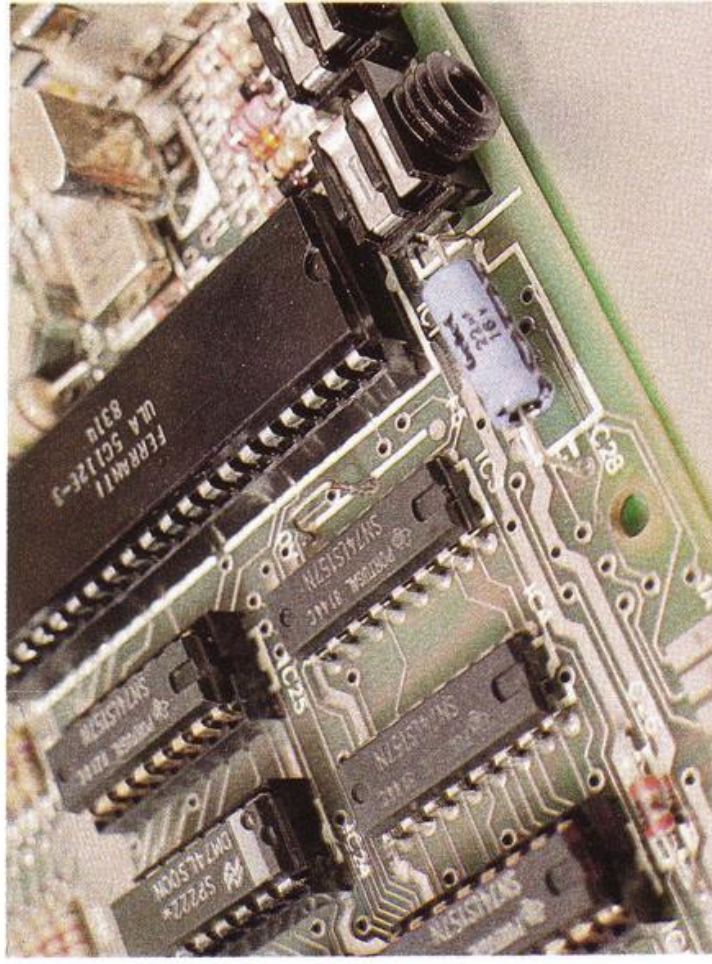
FOR A TO B
FOR A TO C
FOR A TO D
FOR NEXT
FOR NEXT
FOR NEXT
PRINT AT 7.7 "MICROHOBBY SE
NO PRINT AT 4.7, PAPER G-1HC
RONDOP SEMINAL

BORDER 2 : PAPER 2 : INK 0 : CLS

Tecllee el siquiente bucle:

FOR $n = 0$ TO 7 :
PRINT AT $n, 0$; INK n ; MICROHOBBY :
NEXT n

que visualiza la cadena «Mi-



Circuito generador de colores.

CROHOBBY» con diversos colores temporales de tinta. Introduzca ahora este otro bucle:

```
FOR n = 0 TO 10:
PRINT AT n, 0; INK 8;
MICROHOBby SEMANAL;
NEXT n
```

que visualiza la cadena «MICROHOBby» con diversos colores temporales de tinta. Introduzca ahora este otro bucle:

```
FOR n = 0 TO 10:
PRINT AT n, 0; INK 8;
MICROHOBby SEMANAL;
NEXT n
```

Observe como son respetados los atributos temporales de tinta.

El contraste es una característica que indica que el color correspondiente a papel o tinta va a ser en blanco o negro para destacar sobre el otro.

El color de contraste es «negro» cuando el otro es de tonalidad clara, y «blanco» cuando es oscura.

CONTRASTE	COLOR
BLANCO	NEGRO
	AZUL
	ROJO
NEGRO	MAGENTA
	VERDE
	CYAN
	AMARILLO
	BLANCO

El siguiente programa cambia el color de borde y papel; observe el color de la tinta.

```
10 REM *****
CONTRASTE:
*****
```

```
30 FOR n=0 TO 7:
35 BORDER n, 1; PAPER 0; CLS:
40 PRINT AT 10, 10; INK 9; "MICROHOBby SEMANAL";
45 PRINT AT 10, 10; PAPER 1; CLS:
50 PRINT AT 10, 10; INK 8; "MICROHOBby SEMANAL";
55 PRINT AT 10, 10; PAPER 0; CLS:
60 PRINT AT 10, 10; INK 9; "MICROHOBby SEMANAL";
65 PRINT AT 10, 10; PAPER 1; CLS:
70 GO TO 30
```

Tanto los mensajes que envía el ordenador como los que usted visualiza a través de los canales de comunicación cerebro y uno (H y I). tienen la característica de tinta de contraste, en relación con el color del borde.

Edite el siguiente programa que genera con colores temporales, una serie de barras verticales.

```
10 REM *****
TINTA "0":
*****
20 FOR x=0 TO 21:
30 FOR y=0 TO 7:
40 FOR z=0 TO 1:
50 PRINT AT x, y, z; INK 0; "MICROHOBby SEMANAL";
60 PRINT AT x, y, z; INK 1; "MICROHOBby SEMANAL";
70 NEXT z:
80 NEXT y:
90 NEXT x:
100 PRINT AT 14, 7; PAPER 0; CLS:
110 PRINT AT 14, 7; PAPER 1; CLS:
120 GO TO 20
```

Los mensajes de las líneas 90 y 100 están visualizados sobre papel transparente. El primer mensaje no se puede leer correctamente debido al poco contraste que existe en algunas zonas, entre el papel y la tinta; sin embargo, el segundo aprovecha la característica de contraste de tinta para ser legible.

En este otro ejemplo se demuestra la característica de papel de contraste:

```
10 REM *****
PAPEL "0":
*****
20 BORDER 2; PAPER 0; CLS:
30 PRINT AT 10, 10; INK 0; "MICROHOBby SEMANAL";
40 PRINT AT 10, 10; INK 1; "MICROHOBby SEMANAL";
50 PRINT AT 10, 10; PAPER 1; CLS:
60 PRINT AT 10, 10; PAPER 0; CLS:
70 GO TO 30
```

Con el papel de contraste, aparte de quedar el mensaje destacado, se visualiza la cadena «HOBby».

Simulación de colores

Aparte de los ocho colores de que dispone el Spectrum, pueden simularse también en más, disponibles también en dos gamas de brillo, por tanto pueden conseguirse en total hasta «72» tonalidades distintas [(28 + 8) * 2].

La forma de simularlos es bastante simple. Utilizando una rejilla pequeña, similar a un tablero de ajedrez, en que los cuadros blancos corresponden al fondo y los negros a la tinta, se observa que asignando diversos colores al papel y a la tinta, y situada la rejilla a cierta distancia, el ojo integra ambos colores dando como resultado uno distinto que es la mezcla; por ejemplo, con el rojo y el amarillo simularíamos el naranja.

Con el programa número «2», que utiliza los gráficos definidos, asignamos a la letra «A» una rejilla similar a la explicada.

No se preocupe si al ejecutarse la línea 60 el programa desaparece, visualizándose el conocido mensaje:

© 1982 Sinclair Research Ltd

ya que el gráfico permanece en memoria, podrá comprobarlo, seleccionando el modo [G] (gráficos) y pulsando la letra «A».

Retorne al modo anterior y ejecute el programa n.º «3». Dicho programa nos presenta, utilizando la rejilla, todas las combinaciones de tin-



Posibilidades de color en un juego.

Para introducir estos códigos es necesario utilizar la función «CHR\$».

Ejemplo:

```
PRINT CHR$ 16; CHR$ 5; "PEPE"
```

El primer carácter de control corresponde al de tinta y el segundo al código de color; la cadena «PEPE» será visualizada con tinta amarilla (5). Consignando el mismo resultado que:

```
PRINT INK 5; "PEPE"
```

El uso de estos caracteres puede tener utilidad en la asignación de atributos a variables de cadena.

Ejemplo:

```
LET a$ = CHR$ 16 + CHR 0 +
"MICRO"
LET b$ = CHR$ 16 + CHR$ 2 +
"HOBBY"
LET c$ = a$ + b$
PRINT c$
```

Una parte de la variable «c\$» se imprime en tinta negra y la otra en roja.

Se podría haber asignado directamente la totalidad de los caracteres de control a la variable c\$.

Acceso directo

Sin hacer uso de las sentencias «PAPER», «INK», de identificar la forma de ac-

«FLASH» o «BRIGHT» se pueden utilizar de forma directa, los colores.

Pase a modo [E] (extendido) y pulse, por ejemplo, la tecla «1»; ¿qué ocurre?, simplemente que a partir de ese instante escribe con fondo azul; y si vuelve al modo [E] y pulsa simultáneamente «CAPS SHIFT» y la tecla «7», la tinta será blanca.

En el modo extendido se tiene un acceso directo, con las teclas de la fila superior, a todas las combinaciones de color de papel y tinta y la activación o desactivación de las características de parpadeo y brillo.

En la figura adjunta se puede identificar la forma de ac-


```

10 REM *****
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****

```

- PRINT ATTR (0, 5)
- IF ATTR (3, 7) = 10 THEN...
- FOR n = 1 TO ATTR (10, 9)

El valor retornado es un número decimal que debe codificarse en su forma binaria para poder interpretar los atributos.

El significado de cada bit es el siguiente:

BIT	SIGNIFICADO
7	PARPADEO
6	BRILLO
3 a 5	PAPEL
0 a 2	TINTA

Con un ejemplo se comprenderá mejor. Supongamos que el valor retornado por la función «ATTR», correspondiente a una posición de la pantalla es «86». Este, codificado en binario es:

0 1 0 1 0 1 1 0

Atributos de pantalla

Como se explicó anteriormente, cada posición de carácter tiene unos atributos que determinan el color del papel, de la tinta, si tiene brillo o si parpadea. Para conocer estos atributos, el Spectrum dispone de una función denominada «ATTR».

ATTR

Acceso al teclado

USR

MODO E



ATTR

Definición

«ATTR» retorna los atributos de una posición determinada.

Su estructura general es la siguiente:

SENTENCIA	ARGUMENTO
ATTR	(línea, columna)

Ejemplos:

- LET a = ATTR (2, 3)

222 MICROBASIC

- Y por último, los bits «0» a «2» son «110» que al codificarlo también en decimal es «6», que corresponde a

TINTA AMARILLA

Compruébalo con las siguientes instrucciones, que seleccionan estos atributos y posteriormente los leen:

```

BRIGH 1
PAPER 2
INK 6
PRINT AT 0, 0 "A"
PRINT ATTR (0, 0)

```

Se comprueba también que sumando las potencias, de base dos, correspondientes a las posiciones donde hay un bit a «1», se obtiene el valor retornado por «ATTR».

64 + 16 + 4 + 2 = 86

El programa n.º «4» permite visualizar los atributos correspondientes a un código decimal introducido por teclado.

Caracteres de control

En el capítulo dedicado al código ASCII (pág. 37) se comentó que la zona comprendida entre el código 0 y el 31 era la constituida por el código transparente. Dentro de este código hay una serie de caracteres de control que tienen relación con el color, que son:

16	TINTA
17	PAPEL
18	PARPADEO
19	BRILLO
20	INVERTIDO

Los bits «3» a «5» tienen un valor de «010» este valor codificado en decimal es «2» que corresponde al código de color:

PAPEL ROJO

BITS							
7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

PAPEL				TINTA			
0	1	0	1	0	1	1	0



PARPADEO

BRILLO

TINTA

BINARIO	01010110
DECIMAL	86

ATRIBUTOS	
PARPADEO	DESACTIVADO
BRILLO	ACTIVADO
PAPEL	ROJO
TINTA	AMARILLO

Ejemplo función «ATTR».

ta y papel. Observará que algunos colores se encuentran repetidos ya que la combinación de papel «3» y tinta «5», es la misma que papel «5» y tinta «3».

Si pulsamos cualquier tecla, los colores se visualizan en una gama de brillo distinta. Si por el contrario pulsamos «CAPS SHIFT» simultáneamente con la tecla «F», accedemos a la opción «CODIGO».

El código debe ser introducido de la siguiente manera:

- Teclear el código del papel (0 - 7)
- Teclear el código de la tinta (0 - 7)

Pulsar «ENTER».

Este código selecciona el color simulado por la mezcla de ambos, el cual será visualizado con su brillo correspondiente, en la zona de pantalla

conocida como fondo.

Pulsando cualquier tecla se accede otra vez al «muestreo» de colores.

Control de impresión

Con el uso de ciertos comandos se pueden alterar las siguientes características de impresión:

- BRILLO
- INVERSION DE VIDEO
- PARPADEO

Estas, al igual que los colores, pueden ser permanentes o temporales.

El argumento de estas sentencias indica si se desea activar o desactivar cierta característica, utilizando para ello un código de control.

CODIGO	SIGNIFICADO
0	DESACTIVADO
1	ACTIVADO

BRIGH

Acceso al teclado

MODO E

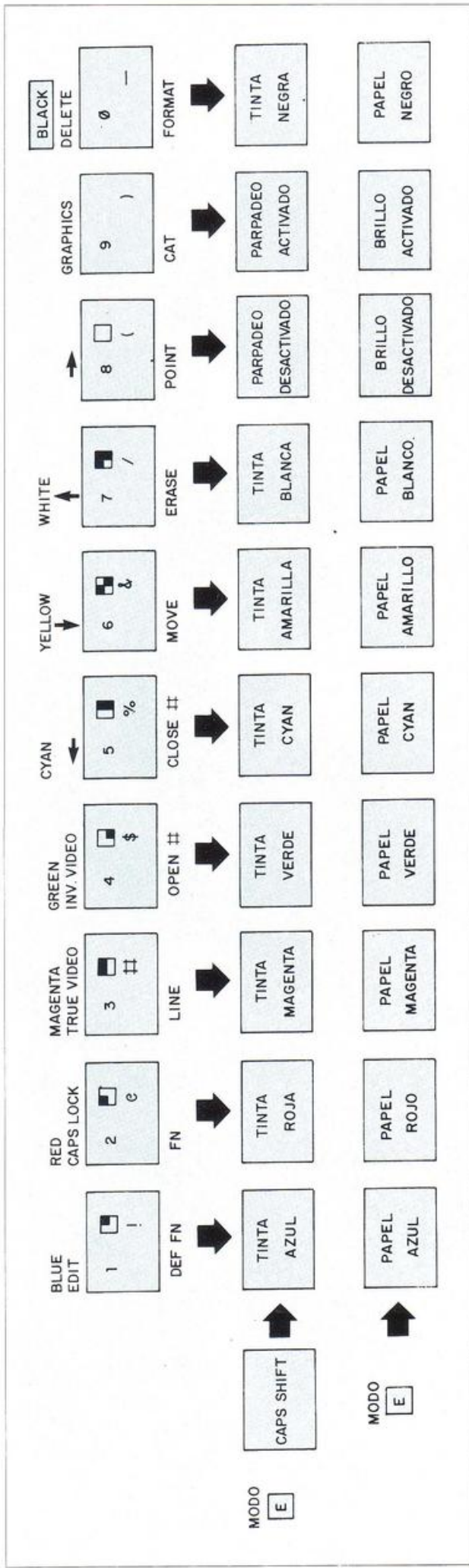
SYMBOL SHIFT

Definición

El comando «BRIGH» permite modificar el brillo de los colores.

La estructura general de esta sentencia es:

219 MICROBASIC



Acceso directo al color

SENTENCIA	ARGUMENTO
BRIGHT	código de control

Ejemplos:

- BRIGHT 0
- PRINT BRIGHT 1; "HOLA"
- BRIGHT 1
- INPUT BRIGHT 1; ">"; a

La opción por defecto es brillo desactivado. El brillo afecta tanto al color del papel como al de la tinta.

Ejemplo:



«BRIGHT» también puede tener como código de control el número «8», con el significado de transparente.

Ejemplo:



Acceso al teclado



Definición

Básicamente esta sentencia intercambia los colores de la tinta y papel. Véase en la página 70 la forma de acceder directamente a esta función y en la 81 una figura alusiva.

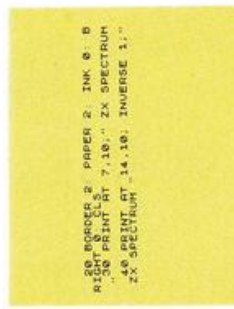
Su estructura general es:

SENTENCIA	ARGUMENTO
INVERSE	código de control

Ejemplos:

- INVERSE 1
- PRINT INVERSE 1; "JUAN"
- INPUT INVERSE 1;
- "edad?"; a
- INVERSE 0

También en este caso la opción por defecto es «VIDEO TRUE». Veamos un ejemplo:



Acceso al teclado



Definición

Con la sentencia «FLASH» se controla la característica

de parpadeo de los caracteres. Secuencialmente se van intercambiando los colores de tinta y papel.

Su estructura general es la siguiente:

SENTENCIA	ARGUMENTO
FLASH	código de control

Ejemplos:

- FLASH 1
- PRINT FLASH 1; "HOP"
- FLASH 0
- INPUT FLASH 1; "? "; a

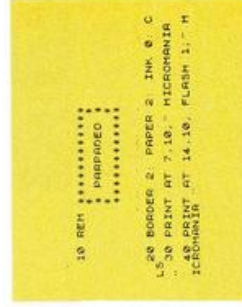
El atributo por defecto es: parpadeo desactivado. Las sentencias «FLASH» y «BRIGHT» al igual que «PA- PER» e «INK» controlan las características de impresión de caracteres completos; es decir, de bloques de 64 pixels (8 * 8).

Introduzca los siguientes comandos directos que harán parpadear a toda la pan-

talla, puesto que se modifican sus atributos.



Veamos otro ejemplo:



«FLASH» también tiene la posibilidad de conservar los atributos de pantalla con el código de control «8» (transparencia).


```

10 REM *****
  : RECTANGULOS
  : *****
  :
  : 20 BORDER 5: PAPER 5: INK 0: C
  : 30 INPUT "Incremento >>>";pas
  :
  : 40 LET base=235
  : 50 LET altu=3155
  : 60 FOR x=10 TO 50 STEP pas
  : 70 DRAW base;0
  : 80 DRAW base;0
  : 90 DRAW base;0
  : 100 DRAW 0;-altu;a
  : 110 LET altu=altu+a
  : 120 LET altu=altu+a
  : 130 NEXT x
  :
  : 140 NEXT n

```

La variable «paso» tiene el mismo significado que en el programa anterior.

Arcos de circunferencia

Como se explicó anteriormente, el parámetro «Z» de una sentencia «DRAW» era opcional y permitía dibujar arcos de circunferencia (líneas curvas). Este parámetro indica el ángulo de giro expresado en radianes.

Cuando «Z» es positivo, el giro se realiza en sentido contrario a las agujas del reloj (hacia la izquierda).

Ejemplo:

```

PLOT 30, 30
DRAW 100, 100, 3

```

PROGRAMA 3

```

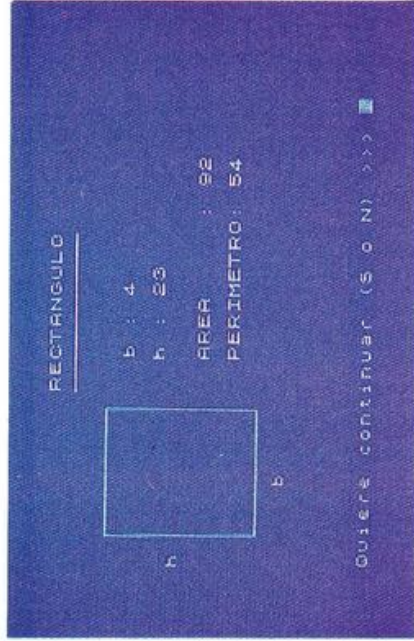
10 REM *****
  : CURSO/BASIC
  : *****
  :
  : 12 BORDER 5: PAPER 5: INK 9: C
  :
  : 14 RANDOMIZE
  : 20 FOR x=0 TO 31
  : 30 FOR y=0 TO 31
  : 40 LET caracter=INT (RND*2)

```

```

9 42 LET grafico=INT (RND*15)+12
  :
  : 44 LET color=INT (RND*8)
  : 50 IF caracter=1 THEN PRINT AT
  : y,x: PAPER color;CHR$ grafico:
  : GO TO 70
  :
  : 60 PRINT AT y,x: " "
  : 70 NEXT x
  : 80 NEXT y
  : 90 PRINT #0: " Pulsa una tecla
  : para continuar."
  : 100 PAUSE 0
  : 102 INPUT 0
  : 110 GO TO 20

```



Aplicaciones en matemáticas.



Gráfico de «baja resolución».

Cuando «Z» es negativo, el giro se realiza en el sentido de derecha).

PROGRAMA DEPURACION

```

9370 STOP
9380 REM DEPURADOR BASIC
9390 LET err_sp=(PEEK 23614)+256
9400 +PEEK 23613)
9410 POKE 23692,1
9420 LET linea=(PEEK (err_sp+3))
9430 +PEEK (err_sp+2))
9440 LET position=PEEK (err_sp+4)
9450
9460 OPEN #2:"K"
9470 PRINT AT 0,0:"Parada en ";l
9480 PRINT #2:position
9490 PAUSE 0
9500 IF INKEY$="Y" OR INKEY$="U"
  : THEN GO SUB 9993

```

Capítulo Depuración programas

PROGRAMA 5

```

10 REM *****
  : CURSO/BASIC
  : *****
  :
  : 20 BORDER 4: PAPER 4: INK 0: C
  :
  : 30 RESTORE
  : 40 FOR x=0 TO 31 STEP 2
  : 50 READ color;21
  : 60 FOR y=0 TO 31
  : 70 PRINT PAPER color;AT y,x:
  :
  :

```

PROGRAMA 6

```

10 REM *****
  : CURSO/BASIC
  : *****
  :
  : 20 FOR n=0 TO 6 STEP 2
  : 30 POKE USR "a"+n,85
  : 40 POKE USR "a"+n+1,170
  : 50 NEXT n
  : 60 NEW

```

PROGRAMA 7

```

100 REM *****
  : CURSO/BASIC
  : *****
  :
  : 102 BORDER 4: PAPER 4: INK 0: C
  :
  : 104 GO SUB 140
  : 106 LET p=CHR$ 144
  : 110 FOR y=3 TO 17 STEP 2
  : 120 FOR x=8 TO 22 STEP 2
  : 130 PRINT BRIGHT;BRILO; PAPER
  : (y-3)/2; INK (x-8)/2;AT y,x;a$;A
  : T y,x+1;a$;AT y+1,x;a$;AT y+1,x+
  : 1;a$
  : 140 NEXT x
  : 150 NEXT y
  : 160 PRINT #0: "Codigo / "
  : "ENTER para continuar"
  : 170 PAUSE 0
  : 180 INPUT 0

```

```

9991 CLOSE #2
9992 INPUT 0
9993 RETURN
9994 IF v$="" THEN INPUT 0: RETU
9995 PRINT AT 0,0:"Contenido ";v
9996
9997 IF LEN v$=2 THEN IF v$(2)="
  : $ THEN PRINT ".....";VAL$ v$;"....."
9998 PAUSE 0
9999 GO TO 9993

```

```

50 NEXT y
60 NEXT x
70 REM CONTORNO
80
90
100 PLOT 0,0
110 DRAW 95,175
120 DRAW 95,175
130 DRAW 95,175
140 PRINT #0: "Sintoniza correc
  : tamente la t
  : 150 PAUSE 0
  : 160 REM TABLA COLORES
  : 170 DATA 0,1,2,3,4,5,6,7,0,1,2,
  : 3,4,5,6,7

```

```

*****
  :
  : 20 FOR n=0 TO 6 STEP 2
  : 30 POKE USR "a"+n,85
  : 40 POKE USR "a"+n+1,170
  : 50 NEXT n
  : 60 NEW

```

```

182 IF INKEY$="F" THEN GO SUB 1
2090 IF BRILLO=0 THEN LET BRILLO
  : =1
110 PRINT AT 0,0,"BRILLO": GO TO
  : 200
200 PRINT AT 0,0: " "; GO T
  : 0
1006 REM NUMEROS
1010 PRINT INVERSE 1;AT 0,8;"
  :
1030 FOR x=9 TO 23 STEP 2
1040 NEXT x; (x-9)/2
1050 LET b$=""
1064 FOR y=3 TO 18 PAPEL
  :
1068 PRINT INVERSE 1;AT y,5;b$(y
  : -2)
1072 NEXT y
1090 FOR y=4 TO 18 STEP 2
1100 PRINT AT y,7;(y-4)/2
1120 NEXT y
1130 RETURN
1200 REM PANTALLA
1210 INPUT "Codigo >>> "; LINE c
1212 IF LEN c<>2 THEN GO TO 121

```



```

0
1200 IF VAL C$(1)<0 OR VAL C$(1)
1210 THEN GO TO 1230
1220 LET PAPER=VAL C$(1)
1230 IF VAL C$(2)<0 OR VAL C$(2)
1240 THEN GO TO 1210
1250 LET I=1:VAL C$(2)
1260 LET d$=""
1270 FOR n=1 TO 32
1280 LET d$=d$+a$
1290 NEXT n

```

```

1260 FOR n=0 TO 21
1270 PRINT PAPER:INK tint
1280 BRIGHT BRILLO,AT n,0;d$
1290 NEXT n
1300 PRINT "0;" Pulsa una tecla
1310 PAUSE 0
1320 CLS
1330 GO SUB 1000
1340 RETURN

```

GRAFICOS

La posibilidad de realizar gráficos con un ordenador como el Spectrum, es muy valioso para el usuario ya que entre las aplicaciones de éstos, unas veces como parte importante y otras como complemento a los programas, merecen destacarse:

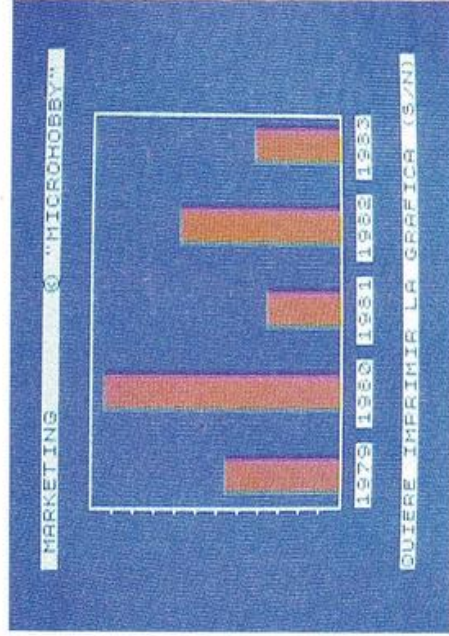
- Posibilidades gráficas en juegos.
- Presentación de gráficos de gestión (histogramas, gráficos de tarta, etc.).
- Utilización en el diseño industrial, esta aplicación es conocida por las siglas CAD-Computer Aided Design y Computer Aided Manufacturing, que traducido al español viene a significar: Diseño asistido por ordenador y fabricación asistida por ordenador.
- Matemáticas (Dibujo de Funciones).

Tipos de Gráficos

Atendiendo a la resolución gráfica de los dibujos, éstos se pueden clasificar en:

- Gráficos de baja resolución.
- Gráficos de alta resolución.

Para la realización de dibujos en baja resolución, nue-



Gráficos de gestión.

den utilizarse bloques de color o los *gráficos predeterminados*; en cambio para los gráficos de alta resolución el Spectrum dispone de las sentencias:

- PLOT
- DRAW
- CIRCLE

Estas sentencias pueden ser combinadas con «PAPER», «INK», «FLASH», «INVERSE» y «OVER».

Bloques de color

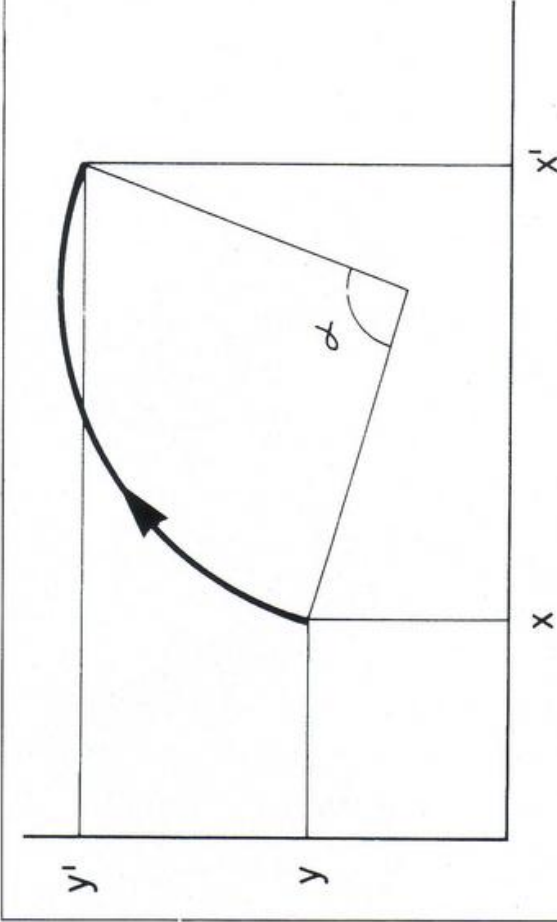
Utilizando el carácter «espacio», con diversos colo-

res de fondo (PAPER), se pueden conseguir gráficos a base de bloques coloreados de 8 por 8 pixel.

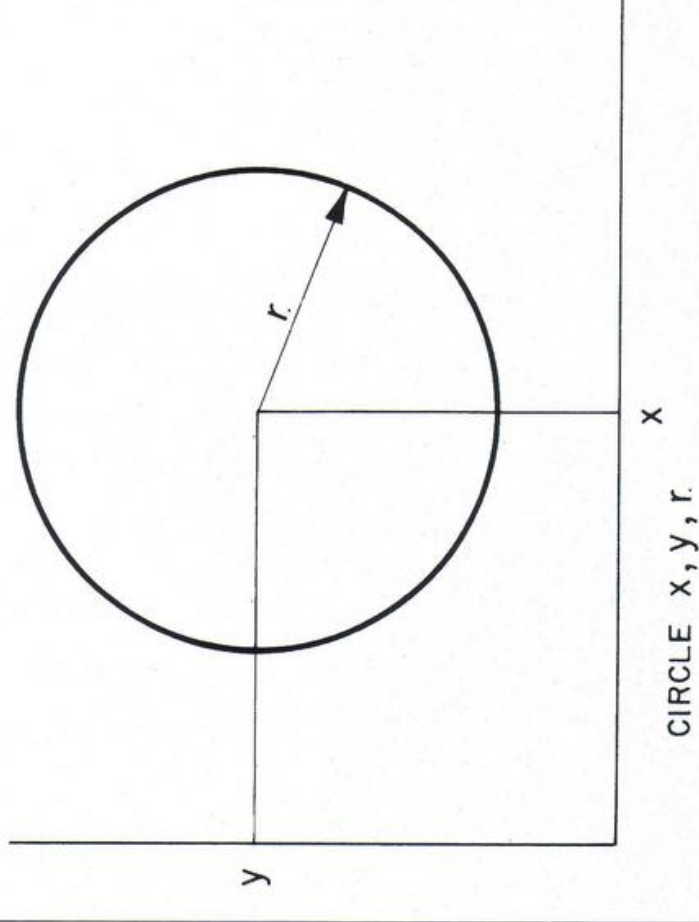
El programa número «1» dibuja en pantalla un simpático pez. El número «2» visualiza el nombre de la revista así como las bandas coloreadas características del Spectrum.

Gráficos predeterminados

Para realizar nuestros dibujos en baja resolución, también podemos utilizar los gráficos que se encuentran predeterminados en el juego de caracteres del Spectrum. Pa-



PLOT X,Y : DRAW X',Y',α



CIRCLE X,Y,r.

Draw y Circle.

PROGRAMA 2

```

10 REM
*****
* CURSOR/BASIC *
*****
* MICROHOBBY *
*****
12 BORDER 1: PAPER 1: INK 0:
14 REM
*****
* BORDE *
*****
30 FOR n=0 TO 31
40 PRINT AT 0,n: PAPER 0: " "
50 PRINT AT 21,n: PAPER 0: " "
60 NEXT n
70 FOR n=1 TO 20
80 PRINT AT 0,n: PAPER 0: " "
90 NEXT n
100 REM
*****
* MICRO *
*****
110 RESTORE
120 READ y,x
130 IF NOT y THEN GO TO 240
140 PRINT AT y,x: PAPER 0: " "
150 GO TO 120
160 REM
*****
* DATOS *
*****
170 DATA 7,5,6,5,5,5,6,5,7,6,
17,7,5,6,5,9,6,9,7,9
180 DATA 5,11,6,11,7,11,7,12,7,
3,3,11
190 DATA 5,17,5,16,5,15,6,15,7,
3,3,7
200 DATA 7,19,5,19,5,19,5,20,5,
11,1
210 DATA 7,23,5,23,5,23,5,24,5,
5,25,7,25,7,24
220 DATA 0,0
230 REM
*****
* HOBBY *
*****

```

rar) dejan de generarse, y con la opción «C» continúa.

Para comprobar la resolución gráfica de las rectas, edite el siguiente programa que «rota» una recta sobre un punto, con un incremento de cinco grados.

```

10 REM *****
   :
   : ROTACION
   :
   : *****
20 BORDER 4 : PAPER 4 : INK 0 : C
30 LET X=10:Y=120

```

230 MICROBASIC

```

350 READ Y,X
360 IF NOT Y,X THEN GO TO 360
370 PRINT PAPER 2,AT Y,X;" "
380 GO TO 360
390 REM
*****
* DATOS *
*****
300 DATA 15,5,14,5,13,5,12,5,11,
13,13,30,19,10,15,10,13,10,13,1,
1,13,19,14,15,12,15,11,
1,30,DATA 11,14,15,13,14,14,1,
4,15,14,15,15,16,14,16,13,16,
13,15
330 DATA 11,18,12,18,13,18,14,1,
8,15,18,15,13,15,20,14,20,13,20,
13,15
340 DATA 13,22,14,22,15,22,15,2,
3,15,27,14,24,13,24,16,24,17,24,
17,23,17,22
350 DATA 0,0
*****
* SPECTRO *
*****
300 LET position=24
310 LET color=2
320 GO SUB 1000
330 LET color=6
340 GO SUB 1000
350 LET color=4
360 GO SUB 1000
370 LET color=5
380 GO SUB 1000
390 GO SUB 0
400 STOP
410 REM
*****
* SUBROUTINA *
*****
300 LET y=20
310 FOR x=position TO 30
PRINT PAPER color,AT y,x;" "
NEXT x
330 LET y=y-1
340 LET position=position+1
350 RETURN

```

```

40 LET y.initial=00
50 LET long.tud=00
60 FOR n=0 TO 360 STEP 5
70 LET cad.in=PI/180
80 LET x=COS(cad.in*long.tud
90 LET y=SIN(cad.in*long.tud
100 PRINT x,y
110 NEXT n

```

Observará que la resolución depende de la inclinación de la recta sobre la horizontal.

El siguiente programa genera una serie de cuadrados crecientes a partir de la esquina inferior izquierda

```

10 REM *****
   : *****
   : CURDAROS :
   : *****
   : *****
20 BORDER 5: PAPER 5: INK 0: C
   :
30 INPUT "Incremento >>>";pas
   :
40 FOR n=1 TO 80 STEP pas
   :
50   PLOT n,n
   :
60   DRAW n,n
   :
70   DRAW n,0
   :
80   DRAW -n,0
   :
90   DRAW 0,-n
   :
100 NEXT n

```

La variable «paso» indica la diferencia entre dos cuadrados consecutivos.
Este otro genera unos rectángulos concéntricos.

PROGRAMA 1

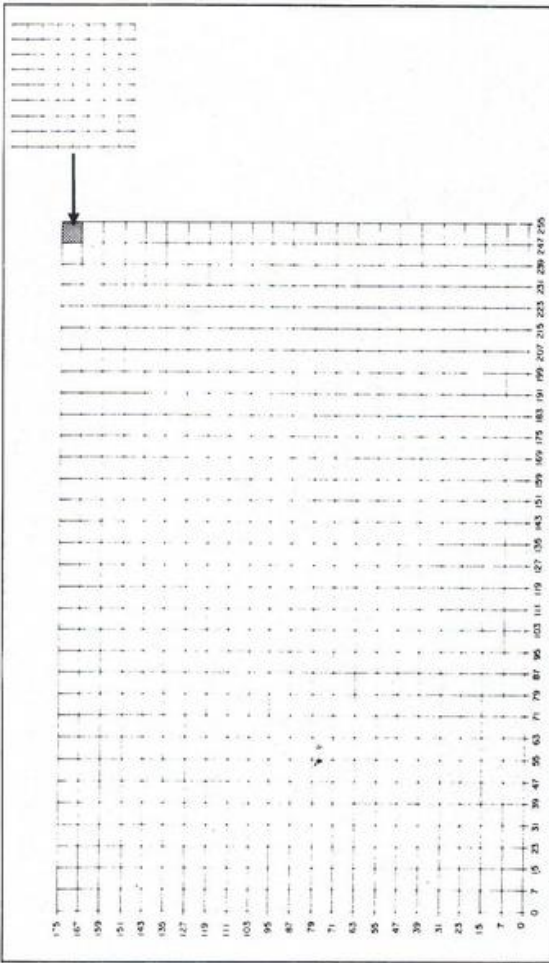
```

10 REM
*****
CURSO/BASIC
*****
***** PEZ
*****
20 BORDER 5: PAPER 5: INK 0: C
L5
30 REM
*****
PRINCIPAL
*****
32 RESTORE
40 LET COL=4
50 GO SUB 1000
60 LET COL=5
70 GO SUB 1000
80 LET COL=3
90 GO SUB 1000
100 LET COL=7
110 GO SUB 1000
120 LET COL=0
130 GO SUB 1000
140 LET COL=10
150 GO SUB 1000
160 PAUSE 100
170 REM
*****
*****
***** BURBUJAS
*****
*****
*****
FOR N=1 TO 9
100 PRINT "Y X:"
110 PRINT PAPER 1, AT Y, X:
120 PAUSE 25
130 NEXT N
140 PRINT "0"
150 PRINT "0"
160 PRINT "0"
170 PRINT "0"
180 PRINT "0"
190 PRINT "0"
200 PRINT "0"
210 PRINT "0"
220 PRINT "0"
230 PRINT "0"
240 PRINT "0"
250 PRINT "0"
260 PRINT "0"
270 PRINT "0"
280 PRINT "0"
290 PRINT "0"
300 PRINT "0"
310 PRINT "0"
320 PRINT "0"
330 PRINT "0"
340 PRINT "0"
350 PRINT "0"
360 PRINT "0"
370 PRINT "0"
380 PRINT "0"
390 PRINT "0"
400 PRINT "0"
410 PRINT "0"
420 PRINT "0"
430 PRINT "0"
440 PRINT "0"
450 PRINT "0"
460 PRINT "0"
470 PRINT "0"
480 PRINT "0"
490 PRINT "0"
500 PRINT "0"
510 PRINT "0"
520 PRINT "0"
530 PRINT "0"
540 PRINT "0"
550 PRINT "0"
560 PRINT "0"
570 PRINT "0"
580 PRINT "0"
590 PRINT "0"
600 PRINT "0"
610 PRINT "0"
620 PRINT "0"
630 PRINT "0"
640 PRINT "0"
650 PRINT "0"
660 PRINT "0"
670 PRINT "0"
680 PRINT "0"
690 PRINT "0"
700 PRINT "0"
710 PRINT "0"
720 PRINT "0"
730 PRINT "0"
740 PRINT "0"
750 PRINT "0"
760 PRINT "0"
770 PRINT "0"
780 PRINT "0"
790 PRINT "0"
800 PRINT "0"
810 PRINT "0"
820 PRINT "0"
830 PRINT "0"
840 PRINT "0"
850 PRINT "0"
860 PRINT "0"
870 PRINT "0"
880 PRINT "0"
890 PRINT "0"
900 PRINT "0"
910 PRINT "0"
920 PRINT "0"
930 PRINT "0"
940 PRINT "0"
950 PRINT "0"
960 PRINT "0"
970 PRINT "0"
980 PRINT "0"
990 PRINT "0"

```

[illegible]

Pantalla en alta resolución.




MICROBASIC 227


```

1 14,22,14,23,15,21,15,22,15,23,
16,22,16,23,17,22,17,23,0,0
2100 DATA 11,26,12,26,12,26,12,2
713,25,14,26,14,27,11,12,11,11
11,10,11,9,10,11,10,10,9,10,8
2110 DATA 12,10,12,10,12,9,12,8,1
27,13,9,13,5,6,9,5,6,8,7,5,6,8,
5,6,4
2120 DATA 14,7,14,6,14,5,14,4,14
3,7,4,7,3,7,2,15,5,15,4,15,3,15
2,15,1,0,0
2130 DATA 9,22,9,23,9,24,10,22,1

```

ra acceder a ellos debe se-
leccionarse el modo «gráfico»
(); en la página 7 de este
manual se explica con detalle
dicho procedimiento.

El programa número «3»
genera una serie de dibujos
aleatorios utilizando estos
gráficos; el color del «papel»
es también aleatorio y la «lin-
ta» tiene atributo de contraste.
La visualización de estos grá-
ficos se realiza con la función
«CHR\$».

Dentro del código ASCII del
Spectrum, los gráficos prede-
finidos tienen un código com-
prendido entre el 128 y el 143
en decimal.

El programa número «4»,
utilizando los gráficos pre-
definidos, visualiza un dibujo
infantil de nuestra redacción.

Pantalla en alta resolución

Cuando se utiliza la panta-
lla en la modalidad de alta re-
solución, el eje de abscisas (x)
se divide en 256 pixel y el de
ordenadas (y) en 176, esto
nos da un total de 45056 pi-
xel.

El origen del eje de coorde-
nadas de la pantalla en alta
resolución se encuentra en el
ángulo inferior izquierdo, a di-
ferencia del de baja resolu-
ción que se encuentra en el
ángulo superior izquierdo.

```

0,24,11,22,11,24,12,22,12,23,12,
24,0,0
2100 DATA 10,23,11,23,0,0
2150 DATA 2,19,3,18,3,19,3,20,4,
17,4,18,4,19,4,20,4,21,5,15,5,16
15,17,15,16,14,15,14,16,15,10,13,11
2160 DATA 17,15,16,9,18,10,13,11
18,12,18,13,18,14,16,15,16,16,1
8,17,16,16,10,19,14,19,15,19,15,19,1
2170 DATA 13,15,14,20,15,20,16,0,0,0
2180 DATA 13,29,12,31,10,30,9,28
7,7,30,5,29,3,28,2,30,0,31

```

PLOT

Acceso al teclado

SIN



ASN

Tipo de sentencia

Comando de dibujo.

Definición

La sentencia «PLOT» visu-
aliza un pixel, determinado por
sus coordenadas «X» e «Y»,
del color especificado de «lin-
ta».

La estructura general de la
sentencia es:

SENTENCIA	ARGUMENTO
PLOT	coord. x, coord. y

Ejemplos:

- PLOT 100, 100
- PLOT PAPER 4; 20, 30
- PLOT INK 5; 127, 30
- PLOT a, b

El siguiente programa ge-
nera una serie de puntos
aleatorios, de distinto color,
en la pantalla.

```

10 REM *****
: PUNTOS *****
: *****
20 BORDER 1 PAPER 5, CLS
30 FOR N=0 TO 100
40 LET X=INT (RND*256)
50 LET Y=INT (RND*176)
60 PLOT INK (COLOR(X,Y))
70 GO TO 30

```

Observará, cuando la pan-
talla tiene cierta cantidad de
puntos, que al visualizar uno
nuevo cambian de color los
de alrededor, esto es debido
a que la representación del
color se hace en baja resolu-
ción, por tanto todos los pixel
(64) de un bloque de caracte-
res deben tener el mismo co-
lor.

Con los siguientes progra-
mas se pueden dibujar punto
a punto las gráficas de las
funciones «SENO» y «COSE-
NO»; para que pueda ser vi-
sualizado un ciclo completo
(360°) ha sido necesario cal-
cular los «puntos» cada dos
grados.

a) Función «SENO»:

```

10 REM *****
: FUNCION SENO *****
: *****
20 DEF FN A TO 180N (X=PI/180)
30 FOR N=0 TO 180
40 LET Y=FN A (N+2)
50 PLOT INK (N+2)
60 NEXT N

```

b) Función «COSENO»:

```

10 REM *****
: FUNCION COSENO *****
: *****
12 LET origin=00
14 LET amplitude=60
16 FOR N=0 TO 180
20 LET Y=FN A (N+2)
30 PLOT INK (N+2)
40 NEXT N

```

DRAW

Acceso al teclado

COS



ACS

Tipo de sentencia

Comando de dibujo

Definición

Con «DRAW» se pueden di-
bujar líneas rectas y curvas
(arcos). Su estructura gene-
ral es:

SENTENCIA	ARGUMENTO
DRAW	coord. x, coord. y, z

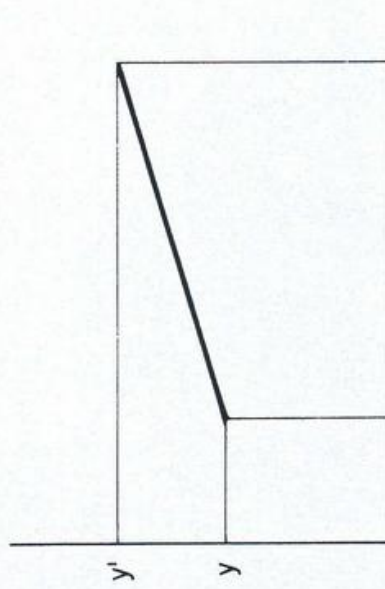
El parámetro «Z» es opcio-
nal y sirve para dibujar arcos.
Ejemplos:

- DRAW 40, 30
- DRAW INK 8; -20, 10
- DRAW 10, 50, 7
- DRAW 7, 10, -3

El punto de origen de una
línea es el último pixel visuali-
zado, bien sea por una sen-
tencia «PLOT», «DRAW» o



PLOT x, y



PLOT x, y : DRAW x', y'

Plot y Draw.

«CIRCLE» y el punto de desti-
no es el especificado por las
coordenadas (relativas al
punto de origen) del argu-
mento de «DRAW»; por ejem-
plo las sentencias:

```

PLOT 100
DRAW 60, -10

```

visualiza una recta entre los

puntos cuyas coordenadas
son: a (100, 100) y b (160,
90).

Las sentencias «RUN»,
«CLS», «NEW» y «CLEAR» po-
sicionan el origen, por defec-
to, en las coordenadas 0, 0.

El programa número «5» vi-
sualiza una serie de rectas
aleatorias de diversos colo-
res, pulsando la tecla «P» (pa-

PROGRAMA 8

[illegible][illegible]

```

410 IF INKEY$="5" THEN LET XO=XO-1
411 LET YO=YO GO SUB detecta
412 LET tiempo=t tiempo+1
414 LET P$=STR$ tiempo
415 PRINT PAPER 6;AT 2,15-LEN P$,1;
416 PRINT PAPER 6;AT 2,15-LEN P$,1;
418 IF tiempo=250 THEN LET punto=50+contador; GO TO 500
419 IF contador=0 THEN LET punto=0; LET contador=0; THEN LET punto=0; GO TO 500
420 GO TO 500
500 REM INSTRUCCIONES
501 PRINT AT 4,X;""
502 PRINT AT 4,X;""
503 PRINT AT 0,2;"HAS OBTENIDO";
504 PRINT AT 0,1; PUNTOS";
505 PRINT AT 0,1; OVER 1; PAPER 6; INK 7;
506 GO TO 100
507 FOR N=0 TO 100: NEXT N
508 PRINT #0;AT 1,4;"Güíters ju";
509 PAUSE 0
510 IF INKEY$="n" OR INKEY$="5" THEN CLS:STOP
511 IF INKEY$="s" OR INKEY$="5" THEN GO TO 590
512 GO TO 590
590 INPUT Q
591 GO TO 590
600 IF puntos>record THEN LET record=puntos
601 PRINT PAPER 6;AT 10,10;record;
602 PRINT AT 0,1;""
603 LET tiempo=0: LET contador=0
604 GO TO 250
1000 REM SUBROUTINE detecta
1001 IF XO=0 THEN LET YO=0
1002 IF YO=0 THEN LET XO=21
1003 IF XO=21 THEN LET YO=21
1004 IF XO>31 THEN LET XO=31
1005 IF SCREEN(YO,XO)="" THEN GO TO 1080
1006 GO TO 1080
1080 IF SCREEN(YO,XO)="" THEN LET contador=contador+1: PRINT PAPER 7;AT 14,15;contador+1;
1081 PRINT AT 4,X;""
1082 LET Y=YO: LET X=XO
1083 PRINT AT Y,X;""
1084 RETURN
2000 REM INSTRUCCIONES
2001 GO TO 0
2002 GO TO 0
2003 GO TO 0
2004 DRAW 255,0
2005 DRAW 255,0
2006 DRAW -255,0
2007 PRINT INVERSE 1;AT 4,0;""
2070 PRINT MICROBERRY;
2080 PRINT INVERSE 1;AT 17,0;""
2090 PRINT #0;AT 0,1;"ABERINTOS";
2091 PRINT #0;AT 0,1;"Pulsa una tecla para continuar";
2100 PAUSE 0
2110 CLS
2120 PRINT
2130 PRINT AT 0,2;"Tienes que re";
2140 PRINT AT 0,2;"cuatro", que hay en la";
2150 PRINT AT 0,2;"habitaco", que hay en la";
2160 PRINT AT 0,2;"n", en el menor tie";
2170 PRINT AT 0,2;"dispones de un tie";
2180 PRINT AT 0,2;"250 Pelictones";
2190 PRINT AT 11,11;"CONTROLES";
2200 PRINT AT 14,9;""
2210 PRINT AT 14,9;""
2220 PRINT AT 16,9;""
2230 PRINT AT 16,9;""
2240 PRINT AT 18,9;""
2250 PRINT AT 18,9;""
2260 PRINT AT 20,9;""
2270 PRINT AT 20,9;""
2280 PRINT #0;AT 1,1;"Pulsa una tecla para comen";
2290 PRINT #0;AT 1,1;"Pulsa una tecla para comen";
2300 INK 6;
2310 RETURN

```

Ejemplo:

PLOT 50, 50
DRAW 100, 100, -1

El siguiente programa visualiza una esfera con sus husos.

```

10 REM *****
11 *          *
12 *          *
13 *          *
14 *          *
15 *          *
16 *          *
17 *          *
18 *          *
19 *          *
20 BORDER 3: PAPER
21
22 FOR N=0 TO 3.14
23   PLOT 120,0
24   DRAW 0.175,N
25   PLOT 120,0
26   DRAW 0.175,-N
27 NEXT N

```

```

L5 20 BORDER 3: PAPER 3: INK 0: C
30 FOR n=0 TO 3.4 STEP 0.2
40 PLOT 120,0
50 DRAW 0,1.5,n
55 PLOT 120,0
60 DRAW 0,1.5,-n
70 NEXT n

```

Utilizando como parámetro «Z» el valor PI que equivale a 180°, una semicircunferencia, el siguiente programa genera una serie de «ondas»:

10 REH ++++++
 + ONDAS +
 ++++++

PROGRAMA 4

```

10 REM
11
12 BORDER 5: PAPER 5:
13 REM CURSO/BASIC
14
15 CURSO/BASIC
16
17 GRAFICO
18
19 GRAFICO
20
21
22
23
24
25
26
27
28
29
30 PRINT PAPER 2; INK
31
32
33
34
35
36
37
38
39
40 NEXT N
41
42
43
44
45
46
47
48
49
50 NEXT N
51
52 INK 4
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

```

CLS
5:01 0.00

Introduzca, por ejemplo, como datos de «ángulo de gi-

```

10 REM *****
11 * CURVAS??? *
12 * ***** *
13
14 BORDER 5: PAPER 5: INK 0: C
15 INPUT "Angulo de giro >>>"
16 CLS
17 PLOT 0,0,127,0
18 DRAW 50,50,Angulo

```

L5 20 BORDER 4: PAPER 4, INK 0: C
20 FOR x=100 TO 20 STEP -20
40 PLOT x TO 3
50 FOR n=100 TO 50
60 DRAW 0, 0, 1
70 DRAW 0, 0, 1
80 NEXT x
90 NEXT n

Gráfico de «alta resolución».

Como efecto curioso de la sentencia «DRAW» ejecute el siguiente programa:

```

340 PRINT " "
350 PRINT " "
360 PRINT " "
370 PRINT " "
380 PRINT " "
390 PRINT " "
400 PRINT " "
410 PRINT " "
420 PRINT " "
430 PRINT " "
440 PRINT " "
450 PRINT " "
460 PRINT " "
470 PRINT " "
480 PRINT " "
490 PRINT " "
500 PRINT " "
510 PRINT " "
520 PRINT " "
530 PRINT " "
540 PRINT " "
550 PRINT " "
560 PRINT " "

```

11; PAPE
8; "L";
6; "1"; AT
8; "1"; INK 0
9; "1"; INK 0
2; "HOBBY"
3; "1"; INK 3
6; 19; "1";
7; 20; "1"; INK
6; 20; INK
7; 20; INK
6; 20; "1";
5; 20; "1";

240 MICROBASIC

MICROBASIC 233

60, 250, 300, 400, 500, 600,
700, 800 u 803

B Integer out of range

iii Sorprendente, verdad!!!

Programa especial

El programa número «6» dibuja el mapa de España punto a punto; para su confección se ha utilizado principalmente la sentencia «PLOT».

Península	718
Baleares	93
Canarias	158
Rótulo	140
Total	1109

Una vez visualizado el dibujo, podrá almacenar el contenido de la pantalla de la forma:

PROGRAMA 5

```

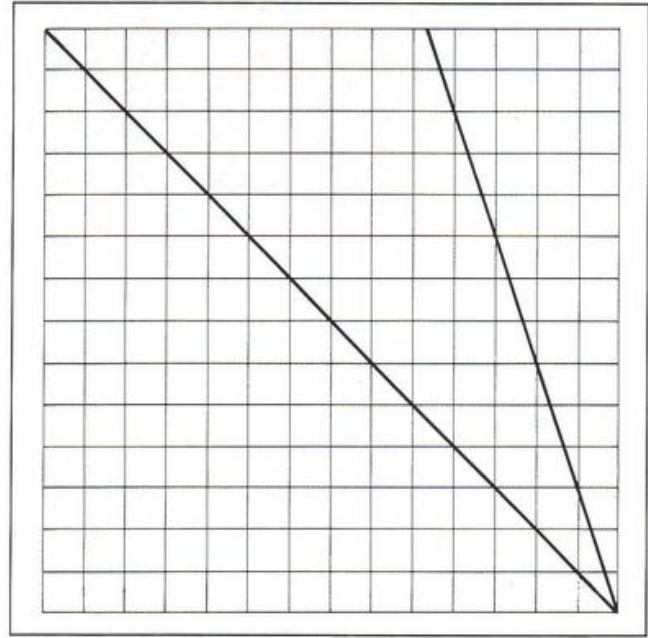
10 REM
*****
* CURSOR/BASIC *
*****
* RECTS *
*****
20 BORDER 1: PAPER 5: CLS
22 LET X = 0
24 REM ONT = 0
26 GO SUB 1000
28 LET X = X + 1
30 LET Y = 0
32 LET Y = Y + 1
34 LET Y = Y + 1
36 LET Y = Y + 1
38 LET Y = Y + 1
40 PLOT X, Y: ALEATORIO
42 PRINT #0: AT 1,3: P = PAR
44 GO SUB 1100
46 GO SUB 1100
48 GO SUB 1100
50 GO SUB 1100
52 GO SUB 1100
54 GO SUB 1100
56 GO SUB 1100
58 GO SUB 1100
60 GO SUB 1100
62 GO SUB 1100
64 GO SUB 1100
66 GO SUB 1100
68 GO SUB 1100
70 LET X = final: X = ALEATORIO - X
72 GO TO 20

```

```

090 LET y final= y aleatorio-y
090 LET x origen=x aleatorio
100 LET y origen=y aleatorio
110 DRAW INK color,x final,y fi
nal
112 IF INKEY$="" OR INKEY$="P"
115 GO SUB 1500
120 STOP TO 50
999
1000 REM SUBROUTINA COORDENADAS
1010 LET x aleatorio=INT (RND#25
6)
1020 LET y aleatorio=INT (RND#17
6)
1030 RETURN
1100 LET color=INT (RND#8)
1110 LET color=INT (RND#8)
1120 RETURN color,color
1200 REM SUBROUTINA ERROR
1300 IF INKEY$="C" OR INKEY$="C"
THEN RETURN
1220 GO TO 1210

```



Resolución Gráfica

Acceso al teclado

SQR

MODE **E**
SYMBOL
SHIFT

Así podrá utilizarlo en cualquier programa «educativo» que se le ocurra.

CIRCLE

PROGRAMA 7

```

1 REM ***** CURSOR/BASIC *****  

***** OVER *****  

2 BORDER 1: PAPER 4:  

FOR Z=1 TO 2  

LET P=0  

DO WHILE P<5  

RANDOMIZE  

LET X=INT (RND*(355)  

LET Y=INT (RND*(175)  

LET X=X-XX  

LET Y=Y-YY  

LET N=X+Y  

FOR M=X+Y STEP  

PLOT X,Y  

NEXT N  

X=X-XX  

Y=Y-YY  

LET N=X+Y  

FOR M=X+Y STEP  

PLOT X,Y  

NEXT N  

P=5  

Z=Z+1  


```

PROGRAMA 7

[illegible]

[illegible][illegible]

* * ESPANA * *

* * * * *

6100 DATA 1,8,2,8,3,8,4,8,5,8,6,

Tipo de sentencia

Comando de dibujo

Definición

La sentencia «CIRCLE» dibuja un círculo. Su estructura es:

SENTENCIA	ARGUMENTO
CIRCLE	coord. x, coord. y, radio

Las coordenadas «x» e «y» corresponden a su centro.

50

- CIRCLE 128, 88, 30
- CIRCLE FLASH 1; 100, 100, 50
- CIRCLE a, b, c
- CIRCLE INK 3; 100, 100, 30

El siguiente programa presenta la diferencia entre un círculo dibujado a base de puntos y el visualizado con «CIRCLE»

```

00 REM *****
01          * CIRCULO *
02          *****
03
04
05 12 BORDER 1: PAPER 4: INK 0: C
06
07 X INITIAL=0
08 LUTCH=0
09 J=0
10 FOR I=0 TO 100
11   FOR J=0 TO 100
12    IF I=J THEN GOTO 100
13    LUTCH=LUTCH+1
14    GOTO 100
15   NEXT J
16 NEXT I
17 PRINT "LUTCH=";LUTCH
18 END

```

Observe las diferencias en precisión y velocidad.

Este otro genera una serie de circunferencias aleatorias; al superponerse cambian de color los bloques de caracteres afectados.

Técnicas avanzadas

Aparte de las sentencias «PLOT», «DRAW» y «CIRCLE»,

Los creadores de pantallas

MICROBASIC 235

existen otros métodos para realizar gráficos; éstos se basan principalmente en la utilización de:

```

35 LET X=INT (RND*216)+20
40 LET Y=INT (RND*136)+20
45 LET C=INT (RND*16)+5
50 LET COLOR=INT (RND*8)
55 IF COLOR=5 THEN GO TO 60
60 PLOT X,Y,COLOR
70 GOTO 35

```

- Programas
- Periféricos

MICROBASIC 235



Técnicas avanzadas en la realización de gráficos.

gráficas utilizan una serie de programas comerciales, que facilitan su labor a la hora de diseñarlas. Entre los más conocidos están:

- MELBOURNE DRAW
- PAINTBOX
- SCREEN MACHINE
- ARTIST

Lógicamente, las características de cada uno de estos programas son distintas de las del resto, pero, generando, se puede:

- Desplazar el cursor en las ocho direcciones, pixel a pixel, sin dibujar.
- Idem dibujando.
- Idem dibujando en inversa.
- Idem borrando.
- Cambiar los atributos de cada bloque de 64 pixel.
- Rellenar áreas de un color determinado.
- Desplazar un gráfico hasta situarlo en la posición adecuada.
- Invertir la pantalla de izquierda a derecha.

- Definir GDU (Gráfico Definido por el Usuario), almacenarlos en cinta y utilizarlos posteriormente.
- Incluir textos.
- Algunos permiten ampliar una zona de la pantalla (zoom), gracias a esta opción se puede dibujar con una mayor precisión.

Existen comercialmente ciertos periféricos, o dispositivos que se conectan al ordenador, con los que también se pueden realizar gráficos; normalmente necesitan de un software para ser utilizados. Principalmente existen cuatro tipos:

- LAPIZ OPTICO
- MESA GRAFICA
- PANTALLA DE VIDEO TACTIL
- RATON

El Lápiz Óptico (Light Pen) es un dispositivo que al des- plazarlo por la pantalla como si estuviera escribiendo, detecta la luz emitida por el televisor y la transforma en señales eléctricas, las cuales son enviadas al ordenador para ser procesadas. Una vez conocidas las coordenadas del punto donde está posicionado el lápiz, el programa se encarga de visualizar el correspondiente pixel.

La Mesa Gráfica es una especie de tablero donde se acopla un brazo articulado, conectado mecánicamente con unos componentes conocidos por el nombre de *potenciómetros*. Este tipo de *tracedor* envía una información al ordenador, que depende de la posición del brazo. Para conocer la posición de un punto se utiliza un sistema de *coordenadas polares*. Existen también otro tipo de mesas más sofisticadas, don-

PROGRAMA 6

```

1 REM *****
2 * CURSOR/BASIC *
3 *****
4 * MAPA *
5 *****
6 *****
7 *****
8 *****
9 *****
10 *****
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****
124 *****
125 *****
126 *****
127 *****
128 *****
129 *****
130 *****
131 *****
132 *****
133 *****
134 *****
135 *****
136 *****
137 *****
138 *****
139 *****
140 *****
141 *****
142 *****
143 *****
144 *****
145 *****
146 *****
147 *****
148 *****
149 *****
150 *****
151 *****
152 *****
153 *****
154 *****
155 *****
156 *****
157 *****
158 *****
159 *****
160 *****
161 *****
162 *****
163 *****
164 *****
165 *****
166 *****
167 *****
168 *****
169 *****
170 *****
171 *****
172 *****
173 *****
174 *****
175 *****
176 *****
177 *****
178 *****
179 *****
180 *****
181 *****
182 *****
183 *****
184 *****
185 *****
186 *****
187 *****
188 *****
189 *****
190 *****
191 *****
192 *****
193 *****
194 *****
195 *****
196 *****
197 *****
198 *****
199 *****
200 *****
201 *****
202 *****
203 *****
204 *****
205 *****
206 *****
207 *****
208 *****
209 *****
210 *****
211 *****
212 *****
213 *****
214 *****
215 *****
216 *****
217 *****
218 *****
219 *****
220 *****
221 *****
222 *****
223 *****
224 *****
225 *****
226 *****
227 *****
228 *****
229 *****
230 *****
231 *****
232 *****
233 *****
234 *****
235 *****
236 *****
237 *****
238 *****
239 *****
240 *****
241 *****
242 *****
243 *****
244 *****
245 *****
246 *****
247 *****
248 *****
249 *****
250 *****
251 *****
252 *****
253 *****
254 *****
255 *****
256 *****
257 *****
258 *****
259 *****
260 *****
261 *****
262 *****
263 *****
264 *****
265 *****
266 *****
267 *****
268 *****
269 *****
270 *****
271 *****
272 *****
273 *****
274 *****
275 *****
276 *****
277 *****
278 *****
279 *****
280 *****
281 *****
282 *****
283 *****
284 *****
285 *****
286 *****
287 *****
288 *****
289 *****
290 *****
291 *****
292 *****
293 *****
294 *****
295 *****
296 *****
297 *****
298 *****
299 *****
300 *****
301 *****
302 *****
303 *****
304 *****
305 *****
306 *****
307 *****
308 *****
309 *****
310 *****
311 *****
312 *****
313 *****
314 *****
315 *****
316 *****
317 *****
318 *****
319 *****
320 *****
321 *****
322 *****
323 *****
324 *****
325 *****
326 *****
327 *****
328 *****
329 *****
330 *****
331 *****
332 *****
333 *****
334 *****
335 *****
336 *****
337 *****
338 *****
339 *****
340 *****
341 *****
342 *****
343 *****
344 *****
345 *****
346 *****
347 *****
348 *****
349 *****
350 *****
351 *****
352 *****
353 *****
354 *****
355 *****
356 *****
357 *****
358 *****
359 *****
360 *****
361 *****
362 *****
363 *****
364 *****
365 *****
366 *****
367 *****
368 *****
369 *****
370 *****
371 *****
372 *****
373 *****
374 *****
375 *****
376 *****
377 *****
378 *****
379 *****
380 *****
381 *****
382 *****
383 *****
384 *****
385 *****
386 *****
387 *****
388 *****
389 *****
390 *****
391 *****
392 *****
393 *****
394 *****
395 *****
396 *****
397 *****
398 *****
399 *****
400 *****
401 *****
402 *****
403 *****
404 *****
405 *****
406 *****
407 *****
408 *****
409 *****
410 *****
411 *****
412 *****
413 *****
414 *****
415 *****
416 *****
417 *****
418 *****
419 *****
420 *****
421 *****
422 *****
423 *****
424 *****
425 *****
426 *****
427 *****
428 *****
429 *****
430 *****
431 *****
432 *****
433 *****
434 *****
435 *****
436 *****
437 *****
438 *****
439 *****
440 *****
441 *****
442 *****
443 *****
444 *****
445 *****
446 *****
447 *****
448 *****
449 *****
450 *****
451 *****
452 *****
453 *****
454 *****
455 *****
456 *****
457 *****
458 *****
459 *****
460 *****
461 *****
462 *****
463 *****
464 *****
465 *****
466 *****
467 *****
468 *****
469 *****
470 *****
471 *****
472 *****
473 *****
474 *****
475 *****
476 *****
477 *****
478 *****
479 *****
480 *****
481 *****
482 *****
483 *****
484 *****
485 *****
486 *****
487 *****
488 *****
489 *****
490 *****
491 *****
492 *****
493 *****
494 *****
495 *****
496 *****
497 *****
498 *****
499 *****
500 *****
501 *****
502 *****
503 *****
504 *****
505 *****
506 *****
507 *****
508 *****
509 *****
510 *****
511 *****
512 *****
513 *****
514 *****
515 *****
516 *****
517 *****
518 *****
519 *****
520 *****
521 *****
522 *****
523 *****
524 *****
525 *****
526 *****
527 *****
528 *****
529 *****
530 *****
531 *****
532 *****
533 *****
534 *****
535 *****
536 *****
537 *****
538 *****
539 *****
540 *****
541 *****
542 *****
543 *****
544 *****
545 *****
546 *****
547 *****
548 *****
549 *****
550 *****
551 *****
552 *****
553 *****
554 *****
555 *****
556 *****
557 *****
558 *****
559 *****
560 *****
561 *****
562 *****
563 *****
564 *****
565 *****
566 *****
567 *****
568 *****
569 *****
570 *****
571 *****
572 *****
573 *****
574 *****
575 *****
576 *****
577 *****
578 *****
579 *****
580 *****
581 *****
582 *****
583 *****
584 *****
585 *****
586 *****
587 *****
588 *****
589 *****
590 *****
591 *****
592 *****
593 *****
594 *****
595 *****
596 *****
597 *****
598 *****
599 *****
600 *****
601 *****
602 *****
603 *****
604 *****
605 *****
606 *****
607 *****
608 *****
609 *****
610 *****
611 *****
612 *****
613 *****
614 *****
615 *****
616 *****
617 *****
618 *****
619 *****
620 *****
621 *****
622 *****
623 *****
624 *****
625 *****
626 *****
627 *****
628 *****
629 *****
630 *****
631 *****
632 *****
633 *****
634 *****
635 *****
636 *****
637 *****
638 *****
639 *****
640 *****
641 *****
642 *****
643 *****
644 *****
645 *****
646 *****
647 *****
648 *****
649 *****
650 *****
651 *****
652 *****
653 *****
654 *****
655 *****
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****
662 *****
663 *****
664 *****
665 *****
666 *****
667 *****
668 *****
669 *****
670 *****
671 *****
672 *****
673 *****
674 *****
675 *****
676 *****
677 *****
678 *****
679 *****
680 *****
681 *****
682 *****
683 *****
684 *****
685 *****
686 *****
687 *****
688 *****
689 *****
690 *****
691 *****
692 *****
693 *****
694 *****
695 *****
696 *****
697 *****
698 *****
699 *****
700 *****
701 *****
702 *****
703 *****
704 *****
705 *****
706 *****
707 *****
708 *****
709 *****
710 *****
711 *****
712 *****
713 *****
714 *****
715 *****
716 *****
717 *****
718 *****
719 *****
720 *****
721 *****
722 *****
723 *****
724 *****
725 *****
726 *****
727 *****
728 *****
729 *****
730 *****
731 *****
732 *****
733 *****
734 *****
735 *****
736 *****
737 *****
738 *****
739 *****
740 *****
741 *****
742 *****
743 *****
744 *****
745 *****
746 *****
747 *****
748 *****
749 *****
750 *****
751 *****
752 *****
753 *****
754 *****
755 *****
756 *****
757 *****
758 *****
759 *****
760 *****
761 *****
762 *****
763 *****
764 *****
765 *****
766 *****
767 *****
768 *****
769 *****
770 *****
771 *****
772 *****
773 *****
774 *****
775 *****
776 *****
777 *****
778 *****
779 *****
780 *****
781 *****
782 *****
783 *****
784 *****
785 *****
786 *****
787 *****
788 *****
789 *****
790 *****
791 *****
792 *****
793 *****
794 *****
795 *****
796 *****
797 *****
798 *****
799 *****
800 *****
801 *****
802 *****
803 *****
804 *****
805 *****
806 *****
807 *****
808 *****
809 *****
810 *****
811 *****
812 *****
813 *****
814 *****
815 *****
816 *****
817 *****
818 *****
819 *****
820 *****
821 *****
822 *****
823 *****
824 *****
825 *****
826 *****
827 *****
828 *****
829 *****
830 *****
831 *****
832 *****
833 *****
834 *****
835 *****
836 *****
837 *****
838 *****
839 *****
840 *****
841 *****
842 *****
843 *****
844 *****
845 *****
846 *****
847 *****
848 *****
849 *****
850 *****
851 *****
852 *****
853 *****
854 *****
855 *****
856 *****
857 *****
858 *****
859 *****
860 *****
861 *****
862 *****
863 *****
864 *****
865 *****
866 *****
867 *****
868 *****
869 *****
870 *****
871 *****
872 *****
873 *****
874 *****
875 *****
876 *****
877 *****
878 *****
879 *****
880 *****
881 *****
882 *****
883 *****
884 *****
885 *****
886 *****
887 *****
888 *****
889 *****
890 *****
891 *****
892 *****
893 *****
894 *****
895 *****
896 *****
897 *****
898 *****
899 *****
900 *****
901 *****
902 *****
903 *****
904 *****
905 *****
906 *****
907 *****
908 *****
909 *****
910 *****
911 *****
912 *****
913 *****
914 *****
915 *****
916 *****
917 *****
918 *****
919 *****
920 *****
921 *****
922 *****
923 *****
924 *****
925 *****
926 *****
927 *****
928 *****
929 *****
930 *****
931 *****
932 *****
933 *****
934 *****
935 *****
936 *****
937 *****
938 *****
939 *****
940 *****
941 *****
942 *****
943 *****
944 *****
945 *****
946 *****
947 *****
948 *****
949 *****
950 *****
951 *****
952 *****
953 *****
954 *****
955 *****
956 *****
957 *****
958 *****
959 *****
960 *****
961 *****
962 *****
963 *****
964 *****
965 *****
966 *****
967 *****
968 *****
969 *****
970 *****
971 *****
972 *****
973 *****
974 *****
975 *****
976 *****
977 *****
978 *****
979 *****
980 *****
981 *****
982 *****
983 *****
984 *****
985 *****
986 *****
987 *****
988 *****
989 *****
990 *****
991 *****
992 *****
993 *****
994 *****
995 *****
996 *****
997 *****
998 *****
999 *****
1000 *****

```


Definición

La sentencia «OVER» controla la *sobreimpresión* de los caracteres. Su estructura general es:

SENTENCIA	ARGUMENTO
OVER	código de control

Ejemplos:

- PRINT OVER 1; «BASIC»
- OVER 0
- OVER c
- PRINT OVER n; 1

Cuando el código de control es «cero», al visualizar un carácter en una posición de la pantalla, se borra el que anteriormente hubiese en esa misma posición. Al conectar el ordenador al código de control por defecto es el «0».

Ejemplo:

```

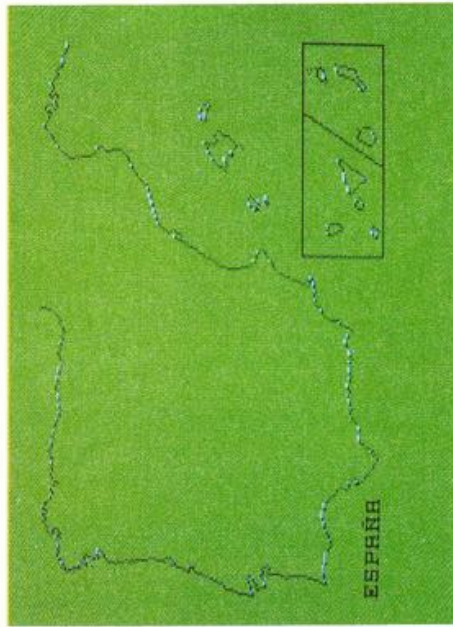
10 REM *****
   : OVER 0 :
   : *****
12 OVER 0
30 PRINT AT 10,10; "MACROBASIC"
50 PRINT "PULSA UNA TECLA PARA CONTINUAR."
60 PRINT AT 10,13; "1."
80 PRINT AT 10,13; "55"

```

Con el código de control «uno», se combinan uno a uno, los píxeles del carácter antiguo y del nuevo, de acuerdo con la función lógica «XOR» (exclusive OR), cuya *tabla de verdad* es la siguiente:

a	b	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Es decir, si los dos píxeles se combinan tienen color de «papel» o de «tinta» el píxel resultante tendrá color de «papel»; por el contrario, si son distintos, es decir, si uno tiene color de papel y otro de tinta,



el píxel resultante tendrá color de «tinta».

Ejemplo:

```

10 REM *****
   : A OVER X :
   : *****
20 PRINT AT 0,0; "A OVER X ="
30 PRINT OVER 1; AT 0,12; "X"

```

Si el código de control es distinto de los mencionados anteriormente nos aparecerá

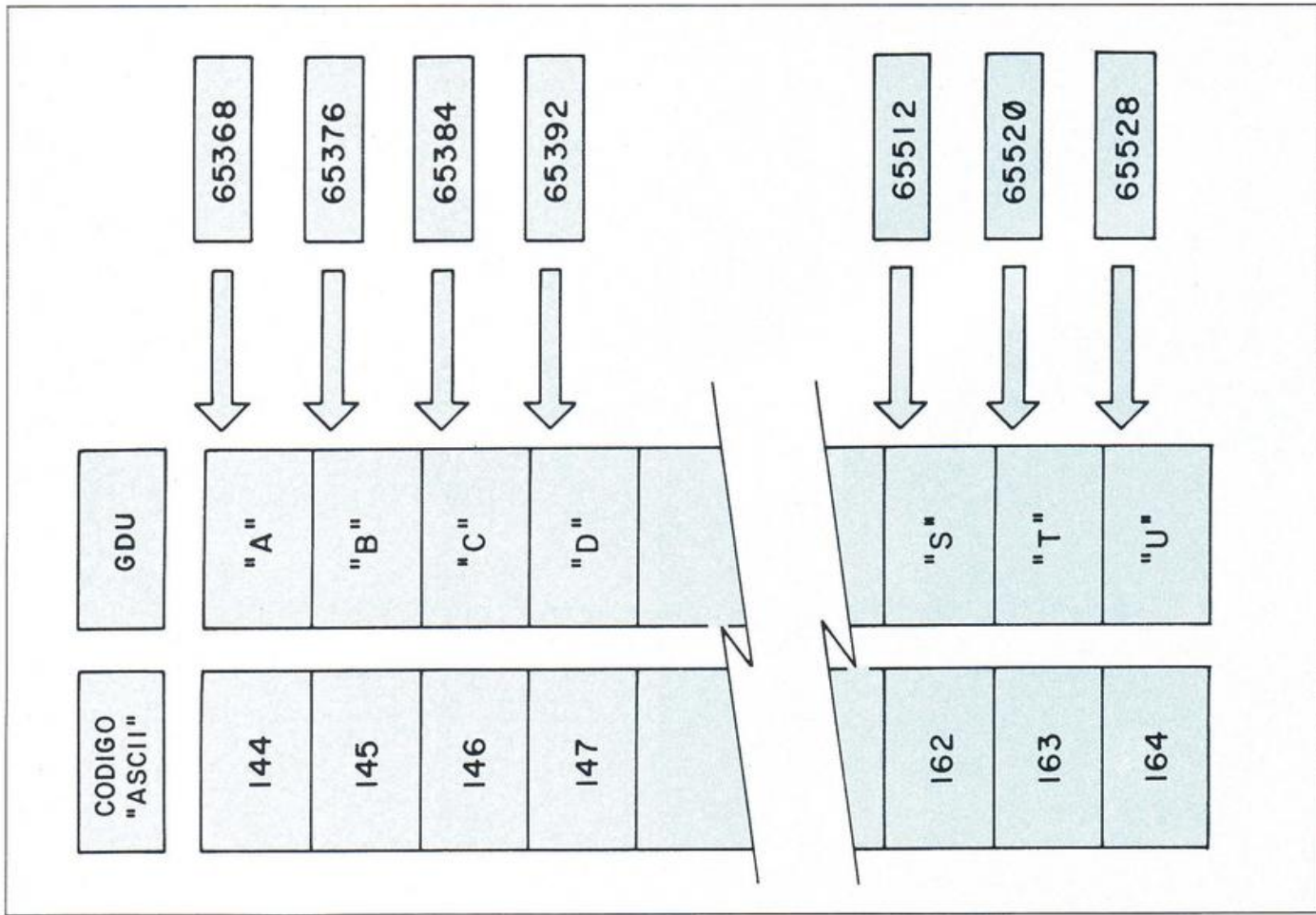
el error de «invalid colour» o «Integer out of range».

El siguiente programa es una aplicación curiosa de la sentencia «OVER».

```

10 REM *****
   : "OVER" :
   : *****
20 REM 0 MICROBASIC
30 LIST OVER 1
40 PRINT AT 0,0; INVERSE 1; "PULSA UNA TECLA PARA CONTINUAR."
50 FOR N=1 TO 3
60 PRINT BRIGHT 1; AT N,13; " "
70 NEXT N
80 PRINT INVERSE 1; AT 6,10; " "
90 OVER 0

```



Direcciones de comienzo «GDU» (48 K).

Al realizar gráficos con la sentencia «OVER» activada, se obtienen unos resultados artísticos.

El siguiente programa realiza unos círculos crecientes, formando como origen las cuatro esquinas; aproximadamente tarda dos minutos en completar el gráfico.

```

10 REM *****
20 OVERCIRC *****
30 BORDER 4: PAPER 4: INK 0: C
40 FOR N=0 TO 80 STEP 2
50 CIRCLE 250-N,N,175-N,N
60 CIRCLE 250-N,N,175-N,N
70 NEXT N
80 OVER 0
90 OVER 0

```

El programa número «7», con la función «OVER» activada, genera un punto aleatorio, a partir del cual se trazan líneas rectas hacia los bordes de la zona de visualización, posteriormente se genera otro y se repite el proceso, de esta manera se obtiene una combinación «espectacular»; el color del fondo va cambiando de color, con una temporización, mientras que la «tinta» tiene color de contraste.

SCREEN\$

Acceso al teclado

MODO E
LEN SYMBOL SHIFT



SCREEN\$

Tipo de sentencia

Función auxiliar.

Definición

Esta función retorna una cadena con el carácter existente en una determinada posición. Su estructura general es la siguiente:

SENTENCIA	ARGUMENTO
SCREEN\$	(fila, columna)

Ejemplo:

```

- PRINT SCREEN$ (5,7)
- IF SCREEN$ (2, 3) = "9" THEN...
- LET a$ = SCREEN$ (10, 11)
- LET b = VAL (SCREEN$ (7, 4))

```

«SCREEN\$» reconoce cualquier carácter comprendido entre los códigos 32 (espacio) y el 127 (©) en decimal; independientemente del atributo temporal que tengan.

Ejemplo:

```

10 PRINT FLASH 1: AT 10, 10:
(HOP)
20 PRINT SCREEN$ (10, 11)

```

Esta función retorna una cadena vacía cuando la posición especificada contiene un carácter fuera del rango.

Ejemplo:

```

10 PRINT CHR$ 135
20 PRINT SCREEN$ (0, 0)

```

El programa número «8» genera un laberinto y utiliza la función «SCREEN\$» para detectar los muros y los objetos.

Almacenamiento de pantallas

«SCREEN\$» puede utilizarse conjuntamente con las sentencias «SAVE» y «LOAD»

para grabar en cinta y posteriormente cargar la imagen (gráficos + texto) que hay en pantalla.

La estructura general es:

- Salvar pantallas.

SAVE nombre SCREEN\$

donde el nombre es una cadena entrecomillada de un máximo de diez caracteres.

Ejemplo:

SAVE (batalla) SCREEN\$

- Cargar pantallas.

LOAD nombre SCREEN\$

Ejemplo:

LOAD (batalla) SCREEN\$

La sentencia «VERIFY» no opera conjuntamente con «SCREEN\$», de manera que no puede verificarse la grabación.

POINT

Acceso al teclado



MODO E

POINT SYMBOL SHIFT

Tipo de sentencia

Función auxiliar

Definición

La función «POINT» indica si un pixel es de color de «lin-

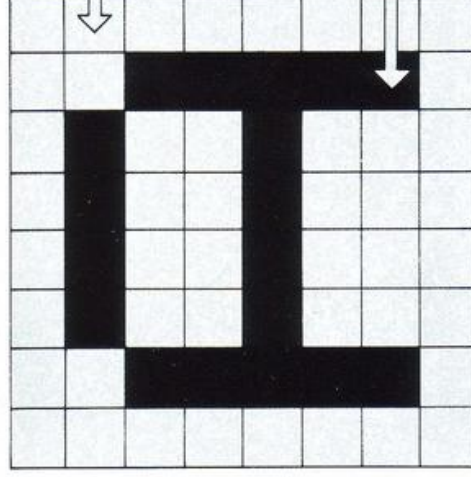
PROGRAMA 2

```

10 REM *****
20 CURSO /BASIC *****
30 PRINT CHR$ (144+n); *****
40 PRINT CHR$ (65+n); *****
50 PRINT CHR$ (154+n); *****
60 PRINT CHR$ (75+n) *****
70 NEXT n AT 14,6:CHR$ 164; *****
80:CHR$ 65 *****
90 PAUSE 0 *****
12 PRINT AT 0,6;"NOTAS GRAFICA *****

```

BYTE 0
BYTE 1
BYTE 2
BYTE 3
BYTE 4
BYTE 5
BYTE 6
BYTE 7



BYT = 0

BYT = 1

Bit y Byte.

El siguiente programa nos indica cuál es la dirección de comienzo de cada «GDU»

```

1 REM *****
2 DIRECCIONES *****
3 GRAFICOS *****
4 *****
5 LET LETC=65: TO USR "un ST
6 FOR N=0 TO 9:CHR$ 164; *****
7 PRINT CHR$ (65+n); *****
8 PRINT CHR$ (154+n); *****
9 NEXT n

```

pero no para uno de 16 K; por lo tanto, para almacenar los gráficos debe utilizarse la expresión:

USR (detrá)

que lo almacena en su posición correspondiente, independientemente de la capacidad de memoria del ordenador. Utilizando esta expresión, el ejemplo anterior que-

Como es lógico, estas direcciones son distintas en los Spectrum de 16 K y 48 K, ya que este último tiene más memoria, por tanto, si para definir el primer «byte» del «GDU» asignado a la letra «A» utilizamos:

POKE 65368, dato

esta fórmula será correcta para un Spectrum de 48 K,

GRAFICOS DEFINIDOS

Posiblemente éste sea uno de los capítulos más esperados por el lector, ya que, tras su lectura, le va a permitir definir, sin dificultad, sus propios gráficos, con el único límite de su imaginación.

Abreviadamente se les conoce por las siglas inglesas «UDG», (User Defined Graphic) o las españolas «GDU» (Gráficos Definidos por Usuario).

Con las técnicas que se van a describir en este capítulo, se pueden definir hasta 21 gráficos, quedando asignados a las letras comprendidas entre la «A» y la «U», ambas inclusive.

¿Cómo se almacenan?

Al igual que los caracteres, cada «GDU» está formado por una matriz de «8» por «8» pixel, en total 64; el contenido de cada uno de éstos (1-activo, 0-desactivo) queda reflejado en un *bit*, que es la unidad de información más pequeña

utilizada en Informática.

Estos «bits» agrupados de «8» en «8» es lo que se denomina *byte*; en el caso de los gráficos, un «byte» queda formado por los «8» pixel de una línea, por tanto para almacenar un «GDU» se necesitan ocho «bytes» de información, y para almacenar los 21 posibles gráficos son necesarios 168 «bytes» de información.

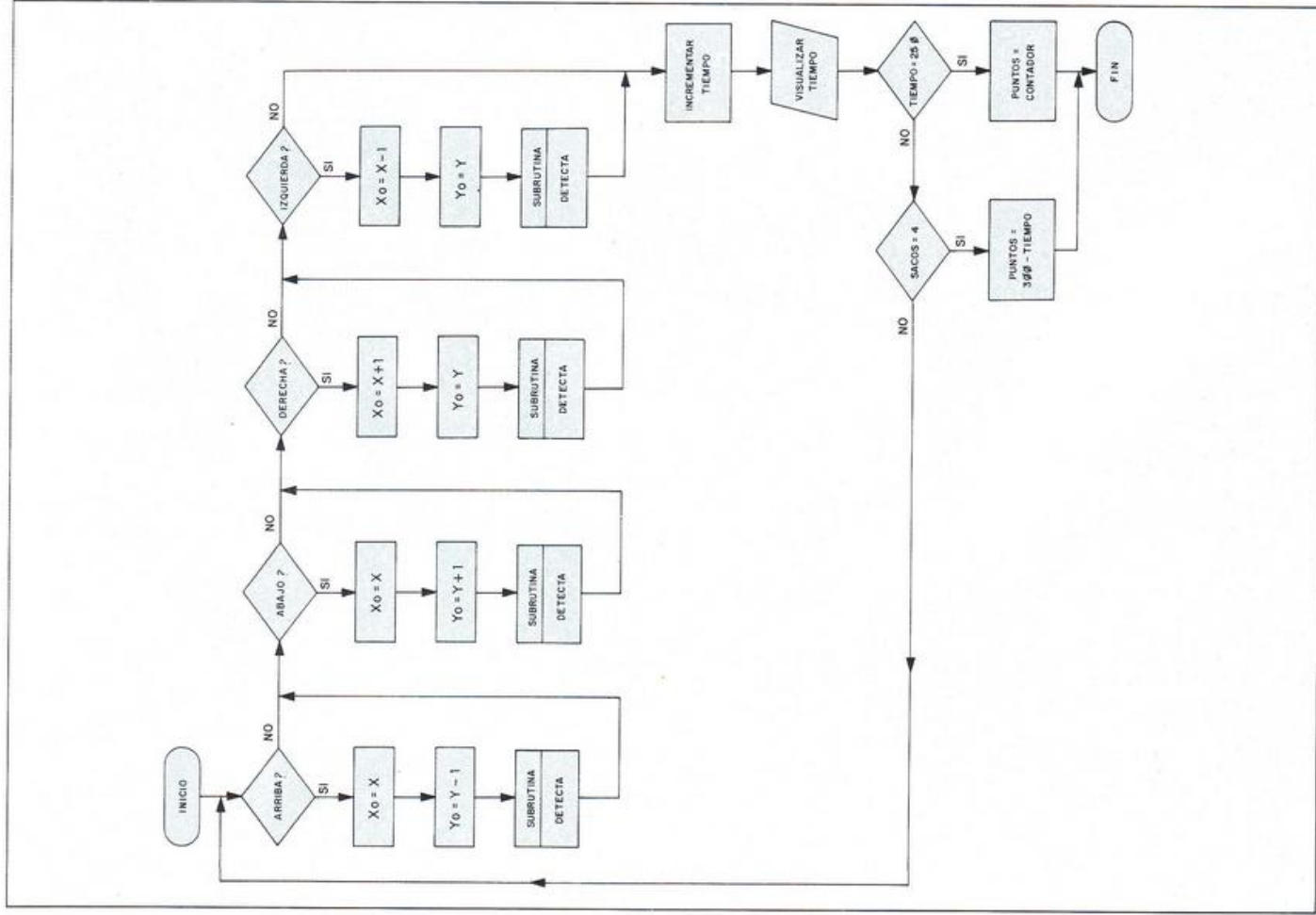
Dentro de la memoria del Spectrum, existe una zona destinada al almacenamiento de los 168 «bytes»; ésta se encuentra localizada al final de la misma.

Para almacenar un «byte» es necesario indicar en qué posición de memoria debe efectuarse. La sentencia encargada de escribir datos en la memoria es «POKE», su estructura general es:

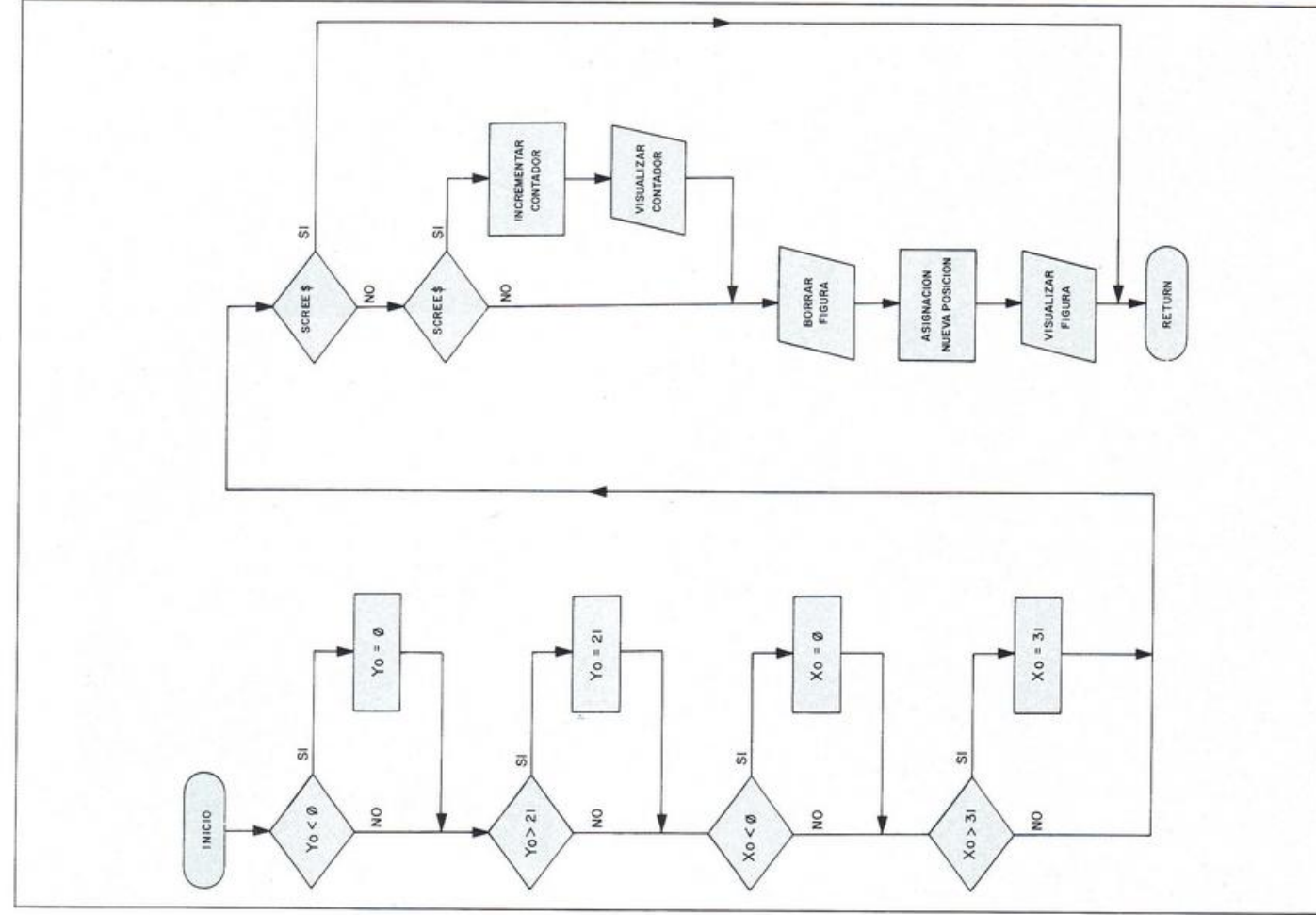
SENTENCIA	ARGUMENTO
POKE	dirección, dato

```

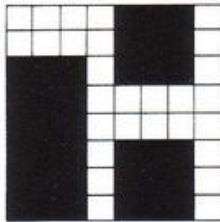
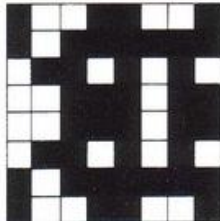
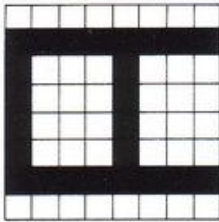
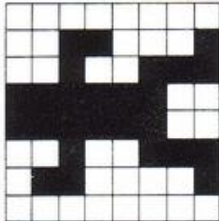
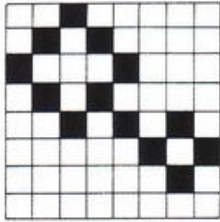
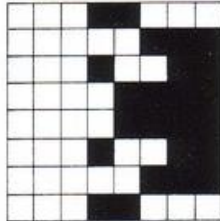
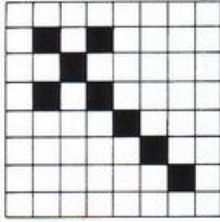
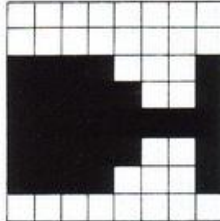
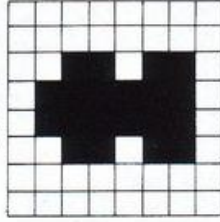
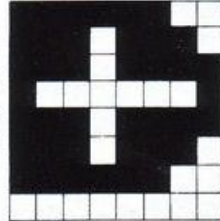
1 REM *****
2 * CURSO/BASIC *
3 *****
4 * ALFABETO *
5 * ESPANOL *
6 *****
7 PRINT #0:AT 1.2,"espera un
momento por favor"
8 FOR P=0 TO 255
9 FOR B=0 TO 7
10 FOR D=0 TO 7
11 FOR S=0 TO 7
12 FOR R=0 TO 7
13 FOR T=0 TO 7
14 FOR U=0 TO 7
15 FOR V=0 TO 7
16 FOR W=0 TO 7
17 FOR X=0 TO 7
18 FOR Y=0 TO 7
19 FOR Z=0 TO 7
20 FOR A=0 TO 7
21 FOR B=0 TO 7
22 FOR C=0 TO 7
23 FOR D=0 TO 7
24 FOR E=0 TO 7
25 FOR F=0 TO 7
26 FOR G=0 TO 7
27 FOR H=0 TO 7
28 FOR I=0 TO 7
29 FOR J=0 TO 7
30 FOR K=0 TO 7
31 FOR L=0 TO 7
32 FOR M=0 TO 7
33 FOR N=0 TO 7
34 FOR O=0 TO 7
35 FOR P=0 TO 7
36 FOR Q=0 TO 7
37 FOR R=0 TO 7
38 FOR S=0 TO 7
39 FOR T=0 TO 7
40 FOR U=0 TO 7
41 FOR V=0 TO 7
42 FOR W=0 TO 7
43 FOR X=0 TO 7
44 FOR Y=0 TO 7
45 FOR Z=0 TO 7
46 FOR A=0 TO 7
47 FOR B=0 TO 7
48 FOR C=0 TO 7
49 FOR D=0 TO 7
50 FOR E=0 TO 7
51 FOR F=0 TO 7
52 FOR G=0 TO 7
53 FOR H=0 TO 7
54 FOR I=0 TO 7
55 FOR J=0 TO 7
56 FOR K=0 TO 7
57 FOR L=0 TO 7
58 FOR M=0 TO 7
59 FOR N=0 TO 7
60 FOR O=0 TO 7
61 FOR P=0 TO 7
62 FOR Q=0 TO 7
63 FOR R=0 TO 7
64 FOR S=0 TO 7
65 FOR T=0 TO 7
66 FOR U=0 TO 7
67 FOR V=0 TO 7
68 FOR W=0 TO 7
69 FOR X=0 TO 7
70 FOR Y=0 TO 7
71 FOR Z=0 TO 7
72 FOR A=0 TO 7
73 FOR B=0 TO 7
74 FOR C=0 TO 7
75 FOR D=0 TO 7
76 FOR E=0 TO 7
77 FOR F=0 TO 7
78 FOR G=0 TO 7
79 FOR H=0 TO 7
80 FOR I=0 TO 7
81 FOR J=0 TO 7
82 FOR K=0 TO 7
83 FOR L=0 TO 7
84 FOR M=0 TO 7
85 FOR N=0 TO 7
86 FOR O=0 TO 7
87 FOR P=0 TO 7
88 FOR Q=0 TO 7
89 FOR R=0 TO 7
90 FOR S=0 TO 7
91 FOR T=0 TO 7
92 FOR U=0 TO 7
93 FOR V=0 TO 7
94 FOR W=0 TO 7
95 FOR X=0 TO 7
96 FOR Y=0 TO 7
97 FOR Z=0 TO 7
98 FOR A=0 TO 7
99 FOR B=0 TO 7
100 FOR C=0 TO 7
101 FOR D=0 TO 7
102 FOR E=0 TO 7
103 FOR F=0 TO 7
104 FOR G=0 TO 7
105 FOR H=0 TO 7
106 FOR I=0 TO 7
107 FOR J=0 TO 7
108 FOR K=0 TO 7
109 FOR L=0 TO 7
110 FOR M=0 TO 7
111 FOR N=0 TO 7
112 FOR O=0 TO 7
113 FOR P=0 TO 7
114 FOR Q=0 TO 7
115 FOR R=0 TO 7
116 FOR S=0 TO 7
117 FOR T=0 TO 7
118 FOR U=0 TO 7
119 FOR V=0 TO 7
120 FOR W=0 TO 7
121 FOR X=0 TO 7
122 FOR Y=0 TO 7
123 FOR Z=0 TO 7
124 FOR A=0 TO 7
125 FOR B=0 TO 7
126 FOR C=0 TO 7
127 FOR D=0 TO 7
128 FOR E=0 TO 7
129 FOR F=0 TO 7
130 FOR G=0 TO 7
131 FOR H=0 TO 7
132 FOR I=0 TO 7
133 FOR J=0 TO 7
134 FOR K=0 TO 7
135 FOR L=0 TO 7
136 FOR M=0 TO 7
137 FOR N=0 TO 7
138 FOR O=0 TO 7
139 FOR P=0 TO 7
140 FOR Q=0 TO 7
141 FOR R=0 TO 7
142 FOR S=0 TO 7
143 FOR T=0 TO 7
144 FOR U=0 TO 7
145 FOR V=0 TO 7
146 FOR W=0 TO 7
147 FOR X=0 TO 7
148 FOR Y=0 TO 7
149 FOR Z=0 TO 7
150 FOR A=0 TO 7
151 FOR B=0 TO 7
152 FOR C=0 TO 7
153 FOR D=0 TO 7
154 FOR E=0 TO 7
155 FOR F=0 TO 7
156 FOR G=0 TO 7
157 FOR H=0 TO 7
158 FOR I=0 TO 7
159 FOR J=0 TO 7
160 FOR K=0 TO 7
161 FOR L=0 TO 7
162 FOR M=0 TO 7
163 FOR N=0 TO 7
164 FOR O=0 TO 7
165 FOR P=0 TO 7
166 FOR Q=0 TO 7
167 FOR R=0 TO 7
168 FOR S=0 TO 7
169 FOR T=0 TO 7
170 FOR U=0 TO 7
171 FOR V=0 TO 7
172 FOR W=0 TO 7
173 FOR X=0 TO 7
174 FOR Y=0 TO 7
175 FOR Z=0 TO 7
176 FOR A=0 TO 7
177 FOR B=0 TO 7
178 FOR C=0 TO 7
179 FOR D=0 TO 7
180 FOR E=0 TO 7
181 FOR F=0 TO 7
182 FOR G=0 TO 7
183 FOR H=0 TO 7
184 FOR I=0 TO 7
185 FOR J=0 TO 7
186 FOR K=0 TO 7
187 FOR L=0 TO 7
188 FOR M=0 TO 7
189 FOR N=0 TO 7
190 FOR O=0 TO 7
191 FOR P=0 TO 7
192 FOR Q=0 TO 7
193 FOR R=0 TO 7
194 FOR S=0 TO 7
195 FOR T=0 TO 7
196 FOR U=0 TO 7
197 FOR V=0 TO 7
198 FOR W=0 TO 7
199 FOR X=0 TO 7
200 FOR Y=0 TO 7
201 FOR Z=0 TO 7
202 FOR A=0 TO 7
203 FOR B=0 TO 7
204 FOR C=0 TO 7
205 FOR D=0 TO 7
206 FOR E=0 TO 7
207 FOR F=0 TO 7
208 FOR G=0 TO 7
209 FOR H=0 TO 7
210 FOR I=0 TO 7
211 FOR J=0 TO 7
212 FOR K=0 TO 7
213 FOR L=0 TO 7
214 FOR M=0 TO 7
215 FOR N=0 TO 7
216 FOR O=0 TO 7
217 FOR P=0 TO 7
218 FOR Q=0 TO 7
219 FOR R=0 TO 7
220 FOR S=0 TO 7
221 FOR T=0 TO 7
222 FOR U=0 TO 7
223 FOR V=0 TO 7
224 FOR W=0 TO 7
225 FOR X=0 TO 7
226 FOR Y=0 TO 7
227 FOR Z=0 TO 7
228 FOR A=0 TO 7
229 FOR B=0 TO 7
230 FOR C=0 TO 7
231 FOR D=0 TO 7
232 FOR E=0 TO 7
233 FOR F=0 TO 7
234 FOR G=0 TO 7
235 FOR H=0 TO 7
236 FOR I=0 TO 7
237 FOR J=0 TO 7
238 FOR K=0 TO 7
239 FOR L=0 TO 7
240 FOR M=0 TO 7
241 FOR N=0 TO 7
242 FOR O=0 TO 7
243 FOR P=0 TO 7
244 FOR Q=0 TO 7
245 FOR R=0 TO 7
246 FOR S=0 TO 7
247 FOR T=0 TO 7
248 FOR U=0 TO 7
249 FOR V=0 TO 7
250 FOR W=0 TO 7
251 FOR X=0 TO 7
252 FOR Y=0 TO 7
253 FOR Z=0 TO 7
254 FOR A=0 TO 7
255 FOR B=0 TO 7
256 FOR C=0 TO 7
257 FOR D=0 TO 7
258 FOR E=0 TO 7
259 FOR F=0 TO 7
260 FOR G=0 TO 7
261 FOR H=0 TO 7
262 FOR I=0 TO 7
263 FOR J=0 TO 7
264 FOR K=0 TO 7
265 FOR L=0 TO 7
266 FOR M=0 TO 7
267 FOR N=0 TO 7
268 FOR O=0 TO 7
269 FOR P=0 TO 7
270 FOR Q=0 TO 7
271 FOR R=0 TO 7
272 FOR S=0 TO 7
273 FOR T=0 TO 7
274 FOR U=0 TO 7
275 FOR V=0 TO 7
276 FOR W=0 TO 7
277 FOR X=0 TO 7
278 FOR Y=0 TO 7
279 FOR Z=0 TO 7
280 FOR A=0 TO 7
281 FOR B=0 TO 7
282 FOR C=0 TO 7
283 FOR D=0 TO 7
284 FOR E=0 TO 7
285 FOR F=0 TO 7
286 FOR G=0 TO 7
287 FOR H=0 TO 7
288 FOR I=0 TO 7
289 FOR J=0 TO 7
290 FOR K=0 TO 7
291 FOR L=0 TO 7
292 FOR M=0 TO 7
293 FOR N=0 TO 7
294 FOR O=0 TO 7
295 FOR P=0 TO 7
296 FOR Q=0 TO 7
297 FOR R=0 TO 7
298 FOR S=0 TO 7
299 FOR T=0 TO 7
300 FOR U=0 TO 7
301 FOR V=0 TO 7
302 FOR W=0 TO 7
303 FOR X=0 TO 7
304 FOR Y=0 TO 7
305 FOR Z=0 TO 7
306 FOR A=0 TO 7
307 FOR B=0 TO 7
308 FOR C=0 TO 7
309 FOR D=0 TO 7
310 FOR E=0 TO 7
311 FOR F=0 TO 7
312 FOR G=0 TO 7
313 FOR H=0 TO 7
314 FOR I=0 TO 7
315 FOR J=0 TO 7
316 FOR K=0 TO 7
317 FOR L=0 TO 7
318 FOR M=0 TO 7
319 FOR N=0 TO 7
320 FOR O=0 TO 7
321 FOR P=0 TO 7
322 FOR Q=0 TO 7
323 FOR R=0 TO 7
324 FOR S=0 TO 7
325 FOR T=0 TO 7
326 FOR U=0 TO 7
327 FOR V=0 TO 7
328 FOR W=0 TO 7
329 FOR X=0 TO 7
330 FOR Y=0 TO 7
331 FOR Z=0 TO 7
332 FOR A=0 TO 7
333 FOR B=0 TO 7
334 FOR C=0 TO 7
335 FOR D=0 TO 7
336 FOR E=0 TO 7
337 FOR F=0 TO 7
338 FOR G=0 TO 7
339 FOR H=0 TO 7
340 FOR I=0 TO 7
341 FOR J=0 TO 7
342 FOR K=0 TO 7
343 FOR L=0 TO 7
344 FOR M=0 TO 7
345 FOR N=0 TO 7
346 FOR O=0 TO 7
347 FOR P=0 TO 7
348 FOR Q=0 TO 7
349 FOR R=0 TO 7
350 FOR S=0 TO 7
351 FOR T=0 TO 7
352 FOR U=0 TO 7
353 FOR V=0 TO 7
354 FOR W=0 TO 7
355 FOR X=0 TO 7
356 FOR Y=0 TO 7
357 FOR Z=0 TO 7
358 FOR A=0 TO 7
359 FOR B=0 TO 7
360 FOR C=0 TO 7
361 FOR D=0 TO 7
362 FOR E=0 TO 7
363 FOR F=0 TO 7
364 FOR G=0 TO 7
365 FOR H=0 TO 7
366 FOR I=0 TO 7
367 FOR J=0 TO 7
368 FOR K=0 TO 7
369 FOR L=0 TO 7
370 FOR M=0 TO 7
371 FOR N=0 TO 7
372 FOR O=0 TO 7
373 FOR P=0 TO 7
374 FOR Q=0 TO 7
375 FOR R=0 TO 7
376 FOR S=0 TO 7
377 FOR T=0 TO 7
378 FOR U=0 TO 7
379 FOR V=0 TO 7
380 FOR W=0 TO 7
381 FOR X=0 TO 7
382 FOR Y=0 TO 7
383 FOR Z=0 TO 7
384 FOR A=0 TO 7
385 FOR B=0 TO 7
386 FOR C=0 TO 7
387 FOR D=0 TO 7
388 FOR E=0 TO 7
389 FOR F=0 TO 7
390 FOR G=0 TO 7
391 FOR H=0 TO 7
392 FOR I=0 TO 7
393 FOR J=0 TO 7
394 FOR K=0 TO 7
395 FOR L=0 TO 7
396 FOR M=0 TO 7
397 FOR N=0 TO 7
398 FOR O=0 TO 7
399 FOR P=0 TO 7
400 FOR Q=0 TO 7
401 FOR R=0 TO 7
402 FOR S=0 TO 7
403 FOR T=0 TO 7
404 FOR U=0 TO 7
405 FOR V=0 TO 7
406 FOR W=0 TO 7
407 FOR X=0 TO 7
408 FOR Y=0 TO 7
409 FOR Z=0 TO 7
410 FOR A=0 TO 7
411 FOR B=0 TO 7
412 FOR C=0 TO 7
413 FOR D=0 TO 7
414 FOR E=0 TO 7
415 FOR F=0 TO 7
416 FOR G=0 TO 7
417 FOR H=0 TO 7
418 FOR I=0 TO 7
419 FOR J=0 TO 7
420 FOR K=0 TO 7
421 FOR L=0 TO 7
422 FOR M=0 TO 7
423 FOR N=0 TO 7
424 FOR O=0 TO 7
425 FOR P=0 TO 7
426 FOR Q=0 TO 7
427 FOR R=0 TO 7
428 FOR S=0 TO 7
429 FOR T=0 TO 7
430 FOR U=0 TO 7
431 FOR V=0 TO 7
432 FOR W=0 TO 7
433 FOR X=0 TO 7
434 FOR Y=0 TO 7
435 FOR Z=0 TO 7
436 FOR A=0 TO 7
437 FOR B=0 TO 7
438 FOR C=0 TO 7
439 FOR D=0 TO 7
440 FOR E=0 TO 7
441 FOR F=0 TO 7
442 FOR G=0 TO 7
443 FOR H=0 TO 7
444 FOR I=0 TO 7
445 FOR J=0 TO 7
446 FOR K=0 TO 7
447 FOR L=0 TO 7
448 FOR M=0 TO 7
449 FOR N=0 TO 7
450 FOR O=0 TO 7
451 FOR P=0 TO 7
452 FOR Q=0 TO 7
453 FOR R=0 TO 7
454 FOR S=0 TO 7
455 FOR T=0 TO 7
456 FOR U=0 TO 7
457 FOR V=0 TO 7
458 FOR W=0 TO 7
459 FOR X=0 TO 7
460 FOR Y=0 TO 7
461 FOR Z=0 TO 7
462 FOR A=0 TO 7
463 FOR B=0 TO 7
464 FOR C=0 TO 7
465 FOR D=0 TO 7
466 FOR E=0 TO 7
467 FOR F=0 TO 7
468 FOR G=0 TO 7
469 FOR H=0 TO 7
470 FOR I=0 TO 7
471 FOR J=0 TO 7
472 FOR K=0 TO 7
473 FOR L=0 TO 7
474 FOR M=0 TO 7
475 FOR N=0 TO 7
476 FOR O=0 TO 7
477 FOR P=0 TO 7
478 FOR Q=0 TO 7
479 FOR R=0 TO 7
480 FOR S=0 TO 7
481 FOR T=0 TO 7
482 FOR U=0 TO 7
483 FOR V=0 TO 7
484 FOR W=0 TO 7
485 FOR X=0 TO 7
486 FOR Y=0 TO 7
487 FOR Z=0 TO 7
488 FOR A=0 TO 7
489 FOR B=0 TO 7
490 FOR C=0 TO 7
491 FOR D=0 TO 7
492 FOR E=0 TO 7
493 FOR F=0 TO 7
494 FOR G=0 TO 7
495 FOR H=0 TO 7
496 FOR I=0 TO 7
497 FOR J=0 TO 7
498 FOR K=0 TO 7
499 FOR L=0 TO 7
500 FOR M=0 TO 7
501 FOR N=0 TO 7
502 FOR O=0 TO 7
503 FOR P=0 TO 7
504 FOR Q=0 TO 7
505 FOR R=0 TO 7
506 FOR S=0 TO 7
507 FOR T=0 TO 7
508 FOR U=0 TO 7
509 FOR V=0 TO 7
510 FOR W=0 TO 7
511 FOR X=0 TO 7
512 FOR Y=0 TO 7
513 FOR Z=0 TO 7
514 FOR A=0 TO 7
515 FOR B=0 TO 7
516 FOR C=0 TO 7
517 FOR D=0 TO 7
518 FOR E=0 TO 7
519 FOR F=0 TO 7
520 FOR G=0 TO 7
521 FOR H=0 TO 7
522 FOR I=0 TO 7
523 FOR J=0 TO 7
524 FOR K=0 TO 7
525 FOR L=0 TO 7
526 FOR M=0 TO 7
527 FOR N=0 TO 7
528 FOR O=0 TO 7
529 FOR P=0 TO 7
530 FOR Q=0 TO 7
531 FOR R=0 TO 7
532 FOR S=0 TO 7
533 FOR T=0 TO 7
534 FOR U=0 TO 7
535 FOR V=0 TO 7
536 FOR W=0 TO 7
537 FOR X=0 TO 7
538 FOR Y=0 TO 7
539 FOR Z=0 TO 7
540 FOR A=0 TO 7
541 FOR B=0 TO 7
542 FOR C=0 TO 7
543 FOR D=0 TO 7
544 FOR E=0 TO 7
545 FOR F=0 TO 7
546 FOR G=0 TO 7
547 FOR H=0 TO 7
548 FOR I=0 TO 7
549 FOR J=0 TO 7
550 FOR K=0 TO 7
551 FOR L=0 TO 7
552 FOR M=0 TO 7
553 FOR N=0 TO 7
554 FOR O=0 TO 7
555 FOR P=0 TO 7
556 FOR Q=0 TO 7
557 FOR R=0 TO 7
558 FOR S=0 TO 7
559 FOR T=0 TO 7
560 FOR U=0 TO 7
561 FOR V=0 TO 7
562 FOR W=0 TO 7
563 FOR X=0 TO 7
564 FOR Y=0 TO 7
565 FOR Z=0 TO 7
566 FOR A=0 TO 7
567 FOR B=0 TO 7
568 FOR C=0 TO 7
569 FOR D=0 TO 7
570 FOR E=0 TO 7
571 FOR F=0 TO 7
572 FOR G=0 TO 7
573 FOR H=0 TO 7
574 FOR I=0 TO 7
575 FOR J=0 TO 7
576 FOR K=0 TO 7
577 FOR L=0 TO 7
578 FOR M=0 TO 7
579 FOR N=0 TO 7
580 FOR O=0 TO 7
581 FOR P=0 TO 7
582 FOR Q=0 TO 7
583 FOR R=0 TO 7
584 FOR S=0 TO 7
585 FOR T=0 TO 7
586 FOR U=0 TO 7
587 FOR V=0 TO 7
588 FOR W=0 TO 7
589 FOR X=0 TO 7
590 FOR Y=0 TO 7
591 FOR Z=0 TO 7
592 FOR A=0 TO 7
593 FOR B=0 TO 7
594 FOR C=0 TO 7
595 FOR D=0 TO 7
596 FOR E=0 TO 7
597 FOR F=0 TO 7
598 FOR G=0 TO 7
599 FOR H=0 TO 7
600 FOR I=0 TO 7
601 FOR J=0 TO 7
602 FOR K=0 TO 7
603 FOR L=0 TO 7
604 FOR M=0 TO 7
605 FOR N=0 TO 7
606 FOR O=0 TO 7
607 FOR P=0 TO 7
608 FOR Q=0 TO 7
609 FOR R=0 TO 7
610 FOR S=0 TO 7
611 FOR T=0 TO 7
612 FOR U=0 TO 7
613 FOR V=0 TO 7
614 FOR W=0 TO 7
615 FOR X=0 TO 7
616 FOR Y=0 TO 7
617 FOR Z=0 TO 7
618 FOR A=0 TO 7
619 FOR B=0 TO 7
620 FOR C=0 TO 7
621 FOR D=0 TO 7
622 FOR E=0 TO 7
623 FOR F=0 TO 7
624 FOR G=0 TO 7
625 FOR H=0 TO 7
626 FOR I=0 TO 7
627 FOR J=0 TO 7
628 FOR K=0 TO 7
629 FOR L=0 TO 7
630 FOR M=0 TO 7
631 FOR N=0 TO 7
632 FOR O=0 TO 7
633 FOR P=0 TO 7
634 FOR Q=0 TO 7
635 FOR R=0 TO 7
636 FOR S=0 TO 7
637 FOR T=0 TO 7
638 FOR U=0 TO 7
639 FOR V=0 TO 7
640 FOR W=0 TO 7
641 FOR X=0 TO 7
642 FOR Y=0 TO 7
643 FOR Z=0 TO 7
644 FOR A=0 TO 7
645 FOR B=0 TO 7
646 FOR C=0 TO 7
647 FOR D=0 TO 7
648 FOR E=0 TO 7
649 FOR F=0 TO 7
650 FOR G=0 TO 7
651 FOR H=0 TO 7
652 FOR I=0 TO 7
653 FOR J=0 TO 7
654 FOR K=0 TO 7
655 FOR L=0 TO 7
656 FOR M=0 TO 7
657 FOR N=0 TO 7
658 FOR O=0 TO 7
659 FOR P=0 TO 7
660 FOR Q=0 TO 7
661 FOR R=0 TO 7
662 FOR S=0 TO 7
663 FOR T=0 TO 7
664 FOR U=0 TO 7
665 FOR V=0 TO 7
666 FOR W=0 TO 7
667 FOR X=0 TO 7
668 FOR Y=0 TO 7
669 FOR Z=0 TO 7
670 FOR A=0 TO 7
671 FOR B=0 TO 7
672 FOR C=0 TO 7
673 FOR D=0 TO 7
674 FOR E=0 TO 7
675 FOR F=0 TO 7
676 FOR G=0 TO 7
677 FOR H=0 TO 7
678 FOR I=0 TO 7
679 FOR J=0 TO 7
680 FOR K=0 TO 7
681 FOR L=0 TO 7
682 FOR M=0 TO 7
683 FOR N=0 TO 7
684 FOR O=0 TO 7
685 FOR P=0 TO 7
686 FOR Q=0 TO 7
687 FOR R=0 TO 7
688 FOR S=0 TO 7
689 FOR T=0 TO 7
690 FOR U=0 TO 7
691 FOR V=0 TO 7
692 FOR W=0 TO 7
693 FOR X=0 TO 7
694 FOR Y=0 TO 7
695 FOR Z=0 TO 7
696 FOR A=0 TO 7
697 FOR B=0 TO 7
698 FOR C=0 TO 7
699 FOR D=0 TO 7
700 FOR E=0 TO 7
701 FOR F=0 TO 7
702 FOR G=0 TO 7
703 FOR H=0 TO 7
704 FOR I=0 TO 7
705 FOR J=0 TO 7
706 FOR K=0 TO 7
707 FOR L=0 TO 7
708 FOR M=0 TO 7
709 FOR N=0 TO 7
710 FOR O=0 TO 7
711 FOR P=0 TO 7
712 FOR Q=0 TO 7
713 FOR R=0 TO 7
714 FOR S=0 TO 7
715 FOR T=0 TO 7
716 FOR U=0 TO 7
717 FOR V=0 TO 7
718 FOR W=0 TO 7
719 FOR X=0 TO 7
720 FOR Y=0 TO 7
721 FOR Z=0 TO 7
722 FOR A=0 TO 7
723 FOR B=0 TO 7
724 FOR C=0 TO 7
725 FOR D=0 TO 7
726 FOR E=0 TO 7
727 FOR F=0 TO 7
728 FOR G=0 TO 7
729 FOR H=0 TO 7
730 FOR I=0 TO 7
731 FOR J=0 TO 7
732 FOR K=0 TO 7
733 FOR L=0 TO 7
734 FOR M=0 TO 7
735 FOR N=0 TO 7
736 FOR O=0 TO 7
737 FOR P=0 TO 7
738 FOR Q=0 TO 7
739 FOR R=0 TO 7
740 FOR S=0 TO 7
741 FOR T=0 TO 7
742 FOR U=0 TO 7
743 FOR V=0 TO 7
744 FOR W=0 TO 7
745 FOR X=0 TO 7
746 FOR Y=0 TO 7
747 FOR Z=0 TO 7
748 FOR A=0 TO 7
749 FOR B=0 TO 7
750 FOR C=0 TO 7
751 FOR D=0 TO 7
752 FOR E=0 TO 7
753 FOR F=0 TO 7
754 FOR G=0 TO 7
755 FOR H=0 TO 7
756 FOR I=0 TO 7
757 FOR J=0 TO 7
758 FOR K=0 TO 7
759 FOR L=0 TO 7
760 FOR M=0 TO 7
761 FOR N=0 TO 7
762 FOR O=0 TO 7
763 FOR P=0 TO 7
764 FOR Q=0 TO 7
765 FOR R=0 TO 7
766 FOR S=0 TO 7
767 FOR T=0 TO 7
768 FOR U=0 TO 7
769 FOR V=0 TO 7
770 FOR W=0 TO 7
771 FOR X=0 TO 7
772 FOR Y=0 TO 7
773 FOR Z=0 TO 7
774 FOR A=0 TO 7
775 FOR B=0 TO 7
776 FOR C=0 TO 7
777 FOR D=0 TO 7
778 FOR E=0 TO 7
779 FOR F=0 TO 7
780 FOR G=0 TO 7
781 FOR H=0 TO 7
782 FOR I=0 TO 7
783 FOR J=0 TO 7
784 FOR K=0 TO 7
785 FOR L=0 TO 7
786 FOR M=0 TO 7
787 FOR N=0 TO 7
788 FOR O=0 TO 7
789 FOR P=0 TO 7
790 FOR Q=0 TO 7
791 FOR R=0 TO 7
792 FOR S=0 TO 7
793 FOR T=0 TO 7
794 FOR U=0 TO 7
795 FOR V=0 TO 7
796 FOR W=0 TO 7
797 FOR X=0 TO 7
798 FOR Y=0 TO 7
799 FOR Z=0 TO 7
800 FOR A=0 TO 7
801 FOR B=0 TO 7
802 FOR C=0 TO 7
803 FOR D=0 TO 7
804 FOR E=0 TO 7
805 FOR F=0 TO 7
806 FOR G=0 TO 7
807 FOR H=0 TO 7
808 FOR I=0 TO 7
809 FOR J=0 TO 7
810 FOR K=0 TO 7
811 FOR L=0 TO 7
812 FOR M=0 TO 7
813 FOR N=0 TO 7
814 FOR O=0 TO 7
815 FOR P=0 TO 7
816 FOR Q=0 TO 7
817 FOR R=0 TO 7
818 FOR S=0 TO 7
819 FOR T=0 TO 7
820 FOR U=0 TO 7
821 FOR V=0 TO 7
822 FOR W=0 TO 7
823 FOR X=0 TO 7
824 FOR Y=0 TO 7
825 FOR Z=0 TO 7
826 FOR A=0 TO 7
827 FOR B=0 TO 7
828 FOR C=0 TO 7
829 FOR D=0 TO 7
830 FOR E=0 TO 7
831 FOR F=0 TO 7
832 FOR G=0 TO 7
833 FOR H=0 TO 7
834 FOR I=0 TO 7
835 FOR J=0 TO 7
836 FOR K=0 TO 7
837 FOR L=0 TO 7
838 FOR M=0 TO 7
839 FOR N=0 TO 7
840 FOR O=0 TO 7
841 FOR P=0 TO 7
842 FOR Q=0 TO 7
843 FOR R=0 TO 7
844 FOR S=0 TO 7
845 FOR T=0 TO 7
846 FOR U=0 TO 7
847 FOR V=0 TO 7
848 FOR W=0 TO 7
849 FOR X=0 TO 7
850 FOR Y=0 TO 7
851 FOR Z=0 TO 7
852 FOR A=0 TO 7
853 FOR B=0 TO 7
854 FOR C=0 TO 7
855 FOR D=0 TO 7
856 FOR E=0 TO 7
857 FOR F=0 TO 7
858 FOR G=0 TO 7
859 FOR H=0 TO 7
860 FOR I=0 TO 7
861 FOR J=0 TO 7
862 FOR K=0 TO 7
863 FOR L=0 TO 7
864 FOR M=0 TO 7
865 FOR N=0 TO 7
866 FOR O=0 TO 7
867 FOR P=0 TO 7
868 FOR Q=0 TO 7
869 FOR R=0 TO 7
870 FOR S=0 TO 7
871 FOR T=0 TO 7
872 FOR U=0 TO 7
873 FOR V=0 TO 7
874 FOR W=0 TO 7
875 FOR X=0 TO 7
876 FOR Y=0 TO 7
877 FOR Z=0 TO 7
878 FOR A=0 TO 7
879 FOR B=0 TO 7
880 FOR C=0 TO 7
881 FOR D=0 TO 7
882 FOR E=0 TO 7
883 FOR F=0 TO 7
884 FOR G=0 TO 7
885 FOR H=0 TO 7
886 FOR I=0 TO 7
887 FOR J=0 TO 7
888 FOR K=0 TO 7
889 FOR L=0 TO 7
890 FOR M=0 TO 7
891 FOR N=0 TO 7
892 FOR O=0 TO 7
893 FOR P=0 TO 7
894 FOR Q=0 TO 7
895 FOR R=0 TO 7
896 FOR S=0 TO 7
897 FOR T=0 TO 7
898 FOR U=0 TO 7
899 FOR V=0 TO 7
900 FOR W=0 TO 7
901 FOR X=0 TO 7
902 FOR Y=0 TO 7
903 FOR Z=0 TO 7
904 FOR A=0 TO 7
905 FOR B=0 TO 7
906 FOR C=0 TO 7
907 FOR D=0 TO 7
908 FOR E=0 TO 7
909 FOR F=0 TO 7
910 FOR G=0 TO 7
911 FOR H=0 TO 7
912 FOR I=0 TO 7
913 FOR J=0 TO 7
914 FOR K=0 TO 7
915 FOR L=0 TO 7
916 FOR M=0 TO 7
917 FOR N=0 TO 7
918 FOR O=0 TO 7
919 FOR P=0 TO 7
920 FOR Q=0 TO 7
921 FOR R=0 TO 7
922 FOR S=0 TO 7
923 FOR T=0 TO 7
924 FOR U=0 TO 7
925 FOR V=0 TO 7
926 FOR W=0 TO 7
927 FOR X=0 TO 7
928 FOR Y=0 TO 7
929 FOR Z=0 TO 7
930 FOR A=0 TO 7
931 FOR B=0 TO 7
932 FOR C=0 TO 7
933 FOR D=0 TO 7
934 FOR E=0 TO 7
935 FOR F=0 TO 7
936 FOR G=0 TO 7
937 FOR H=0 TO 7
938 FOR I=0 TO 7
939 FOR J=0 TO 7
940 FOR K=0 TO 7
941 FOR L=0 TO 7
942 FOR M=0 TO 7
943 FOR N=0 TO 7
944 FOR O=0 TO 7
945 FOR P=0 TO 7
946 FOR Q=0 TO 7
947 FOR R=0 TO 7
948 FOR S=0 TO 7
949 FOR T=0 TO 7
950 FOR U=0 TO 7
951 FOR V=0 TO 7
952 FOR W=0 TO 7
953 FOR X=0 TO 7
954 FOR Y=0 TO 7
955 FOR Z=0 TO 7
956 FOR A=0 TO 7
957 FOR B=0 TO 7
958 FOR C=0 TO 7
959 FOR D=0 TO 7
960 FOR E=0 TO 7
961 FOR F=0 TO 7
962 FOR G=0 TO 7
963 FOR H=0 TO 7
964 FOR I=0 TO 7
965 FOR J=0 TO 7
966 FOR K=0 TO 7
967 FOR L=0 TO 7
968 FOR M=0 TO 7
969 FOR N=0 TO 7
970 FOR O=0 TO 7
971 FOR P=0 TO 7
972 FOR Q=0 TO 7
973 FOR R=0 TO 7
974 FOR S=0 TO 7
975 FOR T=0 TO 7
976 FOR U=0 TO 7
977 FOR V=0 TO 7
978 FOR W=0 TO 7
979 FOR X=0 TO 7
980 FOR Y=0 TO 7
981 FOR Z=0 TO 7
982 FOR A=0 TO 7
983 FOR B=0 TO 7
984 FOR C=0 TO 7
985 FOR D=0 TO 7
986 FOR E=0 TO 7
987 FOR F=0 TO 7
988 FOR G=0 TO 7
989 FOR H=0 TO 7
990 FOR I=0 TO 7
991 FOR J=0 TO 7
992 FOR K=0 TO 7
993 FOR L=0 TO 7
994 FOR M=0 TO 7
995 FOR N=0 TO 7
996 FOR O=0 TO 7
997 FOR P=0 TO 7
998 FOR Q=0 TO 7
999 FOR R=0 TO 7
1000 FOR S=0 TO 7
1001 FOR T=0 TO 7
1002 FOR U=0 TO 7
1003 FOR V=0 TO 7
1004 FOR W=0 TO 7
1005 FOR X=0 TO 7
1006 FOR Y=0 TO 7
1007 FOR Z=0 TO 7
1008 FOR A=0 TO 7
1009 FOR B=0 TO 7
1010 FOR C=0 TO 7
1011 FOR D=0 TO 7
1012 FOR E=0 TO 7
1013 FOR F=0 TO 7
1014 FOR G=0 TO 7
1015 FOR H=0 TO 7
1016 FOR I=0 TO 7
1017 FOR J=0 TO 7
1018 FOR K=0 TO 7
1019 FOR L=0 TO 7
1020 FOR M=0 TO 7
1021 FOR N=0 TO 7
1022 FOR O=0 TO 7
1023 FOR P=0 TO 7
1024 FOR Q=0 TO 7
1025 FOR R=0 TO 7
1026 FOR S=0 TO 7
1027 FOR T=0 TO 7
1028 FOR U=0 TO 7
1029 FOR V=0 TO 7
1030 FOR W=0 TO 7
1031 FOR X=0 TO 7
1032 FOR Y=0 TO 7
1033 FOR Z=0 TO 7
1034 FOR A=0 TO 7
1035 FOR B=0 TO 7
1036 FOR C=0 TO 7
1037 FOR D=0 TO 7
1038 FOR E=0 TO 7
1039 FOR F=0 TO 7
1040 FOR G=0 TO 7
1041 FOR H=0 TO 7
1042 FOR I=0 TO 7
1043 FOR J=0 TO 7
1044 FOR K=0 TO 7
1045 FOR L=0 TO 7
1046 FOR M=0 TO 7
1047 FOR N=0 TO 7
1048 FOR O=0 TO 7
1049 FOR P=0 TO 7
1050 FOR Q=0 TO 7
1051 FOR R=0 TO 7
1052 FOR S=0 TO 7
1053 FOR T=0 TO 7
1054 FOR U=0 TO
```

Rutina movimiento. Programa laberinto.



Subrutina detecta. Programa laberinto.

<p>GDU "A"</p>  <table><tr><td>252</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>252</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>252</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>231</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>231</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>231</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	252										252										252										0										231										231										231										0										<p>GDU "F"</p>  <table><tr><td>195</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>36</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>126</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>219</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>255</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>66</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>126</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>219</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	195										36										126										219										255										66										126										219									
252																																																																																																																																																																	
252																																																																																																																																																																	
252																																																																																																																																																																	
0																																																																																																																																																																	
231																																																																																																																																																																	
231																																																																																																																																																																	
231																																																																																																																																																																	
0																																																																																																																																																																	
195																																																																																																																																																																	
36																																																																																																																																																																	
126																																																																																																																																																																	
219																																																																																																																																																																	
255																																																																																																																																																																	
66																																																																																																																																																																	
126																																																																																																																																																																	
219																																																																																																																																																																	
<p>GDU "B"</p>  <table><tr><td>126</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>66</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>66</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>66</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>126</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>66</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>66</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>66</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	126										66										66										66										126										66										66										66										<p>GDU "G"</p>  <table><tr><td>24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>88</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>126</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>26</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>60</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>36</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>102</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	24										88										126										26										24										60										36										102									
126																																																																																																																																																																	
66																																																																																																																																																																	
66																																																																																																																																																																	
66																																																																																																																																																																	
126																																																																																																																																																																	
66																																																																																																																																																																	
66																																																																																																																																																																	
66																																																																																																																																																																	
24																																																																																																																																																																	
88																																																																																																																																																																	
126																																																																																																																																																																	
26																																																																																																																																																																	
24																																																																																																																																																																	
60																																																																																																																																																																	
36																																																																																																																																																																	
102																																																																																																																																																																	
<p>GDU "C"</p>  <table><tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>10</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>17</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>10</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>20</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>32</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>80</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>32</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	4										10										17										10										20										32										80										32										<p>GDU "H"</p>  <table><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>165</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>153</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>90</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>126</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>126</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	0										0										0										165										153										90										126										126									
4																																																																																																																																																																	
10																																																																																																																																																																	
17																																																																																																																																																																	
10																																																																																																																																																																	
20																																																																																																																																																																	
32																																																																																																																																																																	
80																																																																																																																																																																	
32																																																																																																																																																																	
0																																																																																																																																																																	
0																																																																																																																																																																	
0																																																																																																																																																																	
165																																																																																																																																																																	
153																																																																																																																																																																	
90																																																																																																																																																																	
126																																																																																																																																																																	
126																																																																																																																																																																	
<p>GDU "D"</p>  <table><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>10</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>10</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>16</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>32</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>64</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	0										10										4										10										16										32										64										0										<p>GDU "I"</p>  <table><tr><td>124</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>124</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>124</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>56</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>16</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>16</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>124</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	124										124										124										56										16										16										124																			
0																																																																																																																																																																	
10																																																																																																																																																																	
4																																																																																																																																																																	
10																																																																																																																																																																	
16																																																																																																																																																																	
32																																																																																																																																																																	
64																																																																																																																																																																	
0																																																																																																																																																																	
124																																																																																																																																																																	
124																																																																																																																																																																	
124																																																																																																																																																																	
56																																																																																																																																																																	
16																																																																																																																																																																	
16																																																																																																																																																																	
124																																																																																																																																																																	
<p>GDU "E"</p>  <table><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>60</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>60</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>60</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>60</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	0										24										60										60										24										60										60										0										<p>GDU "J"</p>  <table><tr><td>127</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>119</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>119</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>65</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>119</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>119</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>54</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>28</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	127										119										119										65										119										119										54										28									
0																																																																																																																																																																	
24																																																																																																																																																																	
60																																																																																																																																																																	
60																																																																																																																																																																	
24																																																																																																																																																																	
60																																																																																																																																																																	
60																																																																																																																																																																	
0																																																																																																																																																																	
127																																																																																																																																																																	
119																																																																																																																																																																	
119																																																																																																																																																																	
65																																																																																																																																																																	
119																																																																																																																																																																	
119																																																																																																																																																																	
54																																																																																																																																																																	
28																																																																																																																																																																	

«GDU» del Programa «Palitroque y los Trogloditos».

BITS									
7	6	5	4	3	2	1	0		

DIRECCION	128	64	32	16	8	4	2	1	
USR "A" + 0									
USR "A" + 1									
USR "A" + 2									
USR "A" + 3									
USR "A" + 4									
USR "A" + 5									
USR "A" + 6									
USR "A" + 7									

CODIGO DECIMAL	60	126	219	255	195	255	165	165

«GDU» asignado a la «A».

daria de la siguiente manera:

POKE USR (a), dato

Definición de «GDU»

Para definir un gráfico deberemos primeramente dibujar una cuadrícula de ocho por ocho celdas o cuadrantes; sobre ésta sombrearemos aquellos cuadraditos que nos interesen para formar nuestro dibujo, de una forma similar a la representada en la figura del «terrible monstruo».

Una vez que tengamos el dibujo completo, pasaremos a su programación, para ello, utilizaremos el método más simple que consiste en utilizar la codificación binaria, en la que un pixel o cuadrado sombreado es un «1», y por el

decimal; calcula dichos valores utilizando la función «BIN» en combinación con «PRINT».

Ejemplo:

PRINT BIN 00111100

y así con cada uno de los distintos «bytes». El programa quedaría de la siguiente manera:

```

10 REM *****
11 GDU *****
12 METODO 2 *****
13 POKE USR 0,0
14 POKE USR 1,1
15 POKE USR 2,1
16 POKE USR 3,1
17 POKE USR 4,1
18 POKE USR 5,1
19 POKE USR 6,1
20 POKE USR 7,1
21 POKE USR 8,1
22 POKE USR 9,1
23 POKE USR 10,1
24 POKE USR 11,1
25 POKE USR 12,1
26 POKE USR 13,1
27 POKE USR 14,1
28 POKE USR 15,1
29 POKE USR 16,1
30 POKE USR 17,1
31 POKE USR 18,1
32 POKE USR 19,1
33 POKE USR 20,1
34 POKE USR 21,1
35 POKE USR 22,1
36 POKE USR 23,1
37 POKE USR 24,1
38 POKE USR 25,1
39 POKE USR 26,1
40 POKE USR 27,1
41 POKE USR 28,1
42 POKE USR 29,1
43 POKE USR 30,1
44 POKE USR 31,1
45 POKE USR 32,1
46 POKE USR 33,1
47 POKE USR 34,1
48 POKE USR 35,1
49 POKE USR 36,1
50 POKE USR 37,1
51 POKE USR 38,1
52 POKE USR 39,1
53 POKE USR 40,1
54 POKE USR 41,1
55 POKE USR 42,1
56 POKE USR 43,1
57 POKE USR 44,1
58 POKE USR 45,1
59 POKE USR 46,1
60 POKE USR 47,1
61 POKE USR 48,1
62 POKE USR 49,1
63 POKE USR 50,1
64 POKE USR 51,1
65 POKE USR 52,1
66 POKE USR 53,1
67 POKE USR 54,1
68 POKE USR 55,1
69 POKE USR 56,1
70 POKE USR 57,1
71 POKE USR 58,1
72 POKE USR 59,1
73 POKE USR 60,1
74 POKE USR 61,1
75 POKE USR 62,1
76 POKE USR 63,1
77 POKE USR 64,1
78 POKE USR 65,1
79 POKE USR 66,1
80 POKE USR 67,1
81 POKE USR 68,1
82 POKE USR 69,1
83 POKE USR 70,1
84 POKE USR 71,1
85 POKE USR 72,1
86 POKE USR 73,1
87 POKE USR 74,1
88 POKE USR 75,1
89 POKE USR 76,1
90 POKE USR 77,1
91 POKE USR 78,1
92 POKE USR 79,1
93 POKE USR 80,1
94 POKE USR 81,1
95 POKE USR 82,1
96 POKE USR 83,1
97 POKE USR 84,1
98 POKE USR 85,1
99 POKE USR 86,1
100 POKE USR 87,1
101 POKE USR 88,1
102 POKE USR 89,1
103 POKE USR 90,1
104 POKE USR 91,1
105 POKE USR 92,1
106 POKE USR 93,1
107 POKE USR 94,1
108 POKE USR 95,1
109 POKE USR 96,1
110 POKE USR 97,1
111 POKE USR 98,1
112 POKE USR 99,1
113 POKE USR 100,1
114 POKE USR 101,1
115 POKE USR 102,1
116 POKE USR 103,1
117 POKE USR 104,1
118 POKE USR 105,1
119 POKE USR 106,1
120 POKE USR 107,1
121 POKE USR 108,1
122 POKE USR 109,1
123 POKE USR 110,1
124 POKE USR 111,1
125 POKE USR 112,1
126 POKE USR 113,1
127 POKE USR 114,1
128 POKE USR 115,1
129 POKE USR 116,1
130 POKE USR 117,1
131 POKE USR 118,1
132 POKE USR 119,1
133 POKE USR 120,1
134 POKE USR 121,1
135 POKE USR 122,1
136 POKE USR 123,1
137 POKE USR 124,1
138 POKE USR 125,1
139 POKE USR 126,1
140 POKE USR 127,1
141 POKE USR 128,1
142 POKE USR 129,1
143 POKE USR 130,1
144 POKE USR 131,1
145 POKE USR 132,1
146 POKE USR 133,1
147 POKE USR 134,1
148 POKE USR 135,1
149 POKE USR 136,1
150 POKE USR 137,1
151 POKE USR 138,1
152 POKE USR 139,1
153 POKE USR 140,1
154 POKE USR 141,1
155 POKE USR 142,1
156 POKE USR 143,1
157 POKE USR 144,1
158 POKE USR 145,1
159 POKE USR 146,1
160 POKE USR 147,1
161 POKE USR 148,1
162 POKE USR 149,1
163 POKE USR 150,1
164 POKE USR 151,1
165 POKE USR 152,1
166 POKE USR 153,1
167 POKE USR 154,1
168 POKE USR 155,1
169 POKE USR 156,1
170 POKE USR 157,1
171 POKE USR 158,1
172 POKE USR 159,1
173 POKE USR 160,1
174 POKE USR 161,1
175 POKE USR 162,1
176 POKE USR 163,1
177 POKE USR 164,1
178 POKE USR 165,1
179 POKE USR 166,1
180 POKE USR 167,1
181 POKE USR 168,1
182 POKE USR 169,1
183 POKE USR 170,1
184 POKE USR 171,1
185 POKE USR 172,1
186 POKE USR 173,1
187 POKE USR 174,1
188 POKE USR 175,1
189 POKE USR 176,1
190 POKE USR 177,1
191 POKE USR 178,1
192 POKE USR 179,1
193 POKE USR 180,1
194 POKE USR 181,1
195 POKE USR 182,1
196 POKE USR 183,1
197 POKE USR 184,1
198 POKE USR 185,1
199 POKE USR 186,1
200 POKE USR 187,1
201 POKE USR 188,1
202 POKE USR 189,1
203 POKE USR 190,1
204 POKE USR 191,1
205 POKE USR 192,1
206 POKE USR 193,1
207 POKE USR 194,1
208 POKE USR 195,1
209 POKE USR 196,1
210 POKE USR 197,1
211 POKE USR 198,1
212 POKE USR 199,1
213 POKE USR 200,1
214 POKE USR 201,1
215 POKE USR 202,1
216 POKE USR 203,1
217 POKE USR 204,1
218 POKE USR 205,1
219 POKE USR 206,1
220 POKE USR 207,1
221 POKE USR 208,1
222 POKE USR 209,1
223 POKE USR 210,1
224 POKE USR 211,1
225 POKE USR 212,1
226 POKE USR 213,1
227 POKE USR 214,1
228 POKE USR 215,1
229 POKE USR 216,1
230 POKE USR 217,1
231 POKE USR 218,1
232 POKE USR 219,1
233 POKE USR 220,1
234 POKE USR 221,1
235 POKE USR 222,1
236 POKE USR 223,1
237 POKE USR 224,1
238 POKE USR 225,1
239 POKE USR 226,1
240 POKE USR 227,1
241 POKE USR 228,1
242 POKE USR 229,1
243 POKE USR 230,1
244 POKE USR 231,1
245 POKE USR 232,1
246 POKE USR 233,1
247 POKE USR 234,1
248 POKE USR 235,1
249 POKE USR 236,1
250 POKE USR 237,1
251 POKE USR 238,1
252 POKE USR 239,1
253 POKE USR 240,1
254 POKE USR 241,1
255 POKE USR 242,1
256 POKE USR 243,1
257 POKE USR 244,1
258 POKE USR 245,1
259 POKE USR 246,1
260 POKE USR 247,1
261 POKE USR 248,1
262 POKE USR 249,1
263 POKE USR 250,1
264 POKE USR 251,1
265 POKE USR 252,1
266 POKE USR 253,1
267 POKE USR 254,1
268 POKE USR 255,1
269 POKE USR 256,1
270 POKE USR 257,1
271 POKE USR 258,1
272 POKE USR 259,1
273 POKE USR 260,1
274 POKE USR 261,1
275 POKE USR 262,1
276 POKE USR 263,1
277 POKE USR 264,1
278 POKE USR 265,1
279 POKE USR 266,1
280 POKE USR 267,1
281 POKE USR 268,1
282 POKE USR 269,1
283 POKE USR 270,1
284 POKE USR 271,1
285 POKE USR 272,1
286 POKE USR 273,1
287 POKE USR 274,1
288 POKE USR 275,1
289 POKE USR 276,1
290 POKE USR 277,1
291 POKE USR 278,1
292 POKE USR 279,1
293 POKE USR 280,1
294 POKE USR 281,1
295 POKE USR 282,1
296 POKE USR 283,1
297 POKE USR 284,1
298 POKE USR 285,1
299 POKE USR 286,1
300 POKE USR 287,1
301 POKE USR 288,1
302 POKE USR 289,1
303 POKE USR 290,1
304 POKE USR 291,1
305 POKE USR 292,1
306 POKE USR 293,1
307 POKE USR 294,1
308 POKE USR 295,1
309 POKE USR 296,1
310 POKE USR 297,1
311 POKE USR 298,1
312 POKE USR 299,1
313 POKE USR 300,1
314 POKE USR 301,1
315 POKE USR 302,1
316 POKE USR 303,1
317 POKE USR 304,1
318 POKE USR 305,1
319 POKE USR 306,1
320 POKE USR 307,1
321 POKE USR 308,1
322 POKE USR 309,1
323 POKE USR 310,1
324 POKE USR 311,1
325 POKE USR 312,1
326 POKE USR 313,1
327 POKE USR 314,1
328 POKE USR 315,1
329 POKE USR 316,1
330 POKE USR 317,1
331 POKE USR 318,1
332 POKE USR 319,1
333 POKE USR 320,1
334 POKE USR 321,1
335 POKE USR 322,1
336 POKE USR 323,1
337 POKE USR 324,1
338 POKE USR 325,1
339 POKE USR 326,1
340 POKE USR 327,1
341 POKE USR 328,1
342 POKE USR 329,1
343 POKE USR 330,1
344 POKE USR 331,1
345 POKE USR 332,1
346 POKE USR 333,1
347 POKE USR 334,1
348 POKE USR 335,1
349 POKE USR 336,1
350 POKE USR 337,1
351 POKE USR 338,1
352 POKE USR 339,1
353 POKE USR 340,1
354 POKE USR 341,1
355 POKE USR 342,1
356 POKE USR 343,1
357 POKE USR 344,1
358 POKE USR 345,1
359 POKE USR 346,1
360 POKE USR 347,1
361 POKE USR 348,1
362 POKE USR 349,1
363 POKE USR 350,1
364 POKE USR 351,1
365 POKE USR 352,1
366 POKE USR 353,1
367 POKE USR 354,1
368 POKE USR 355,1
369 POKE USR 356,1
370 POKE USR 357,1
371 POKE USR 358,1
372 POKE USR 359,1
373 POKE USR 360,1
374 POKE USR 361,1
375 POKE USR 362,1
376 POKE USR 363,1
377 POKE USR 364,1
378 POKE USR 365,1
379 POKE USR 366,1
380 POKE USR 367,1
381 POKE USR 368,1
382 POKE USR 369,1
383 POKE USR 370,1
384 POKE USR 371,1
385 POKE USR 372,1
386 POKE USR 373,1
387 POKE USR 374,1
388 POKE USR 375,1
389 POKE USR 376,1
390 POKE USR 377,1
391 POKE USR 378,1
392 POKE USR 379,1
393 POKE USR 380,1
394 POKE USR 381,1
395 POKE USR 382,1
396 POKE USR 383,1
397 POKE USR 384,1
398 POKE USR 385,1
399 POKE USR 386,1
400 POKE USR 387,1
401 POKE USR 388,1
402 POKE USR 389,1
403 POKE USR 390,1
404 POKE USR 391,1
405 POKE USR 392,1
406 POKE USR 393,1
407 POKE USR 394,1
408 POKE USR 395,1
409 POKE USR 396,1
410 POKE USR 397,1
411 POKE USR 398,1
412 POKE USR 399,1
413 POKE USR 400,1
414 POKE USR 401,1
415 POKE USR 402,1
416 POKE USR 403,1
417 POKE USR 404,1
418 POKE USR 405,1
419 POKE USR 406,1
420 POKE USR 407,1
421 POKE USR 408,1
422 POKE USR 409,1
423 POKE USR 410,1
424 POKE USR 411,1
425 POKE USR 412,1
426 POKE USR 413,1
427 POKE USR 414,1
428 POKE USR 415,1
429 POKE USR 416,1
430 POKE USR 417,1
431 POKE USR 418,1
432 POKE USR 419,1
433 POKE USR 420,1
434 POKE USR 421,1
435 POKE USR 422,1
436 POKE USR 423,1
437 POKE USR 424,1
438 POKE USR 425,1
439 POKE USR 426,1
440 POKE USR 427,1
441 POKE USR 428,1
442 POKE USR 429,1
443 POKE USR 430,1
444 POKE USR 431,1
445 POKE USR 432,1
446 POKE USR 433,1
447 POKE USR 434,1
448 POKE USR 435,1
449 POKE USR 436,1
450 POKE USR 437,1
451 POKE USR 438,1
452 POKE USR 439,1
453 POKE USR 440,1
454 POKE USR 441,1
455 POKE USR 442,1
456 POKE USR 443,1
457 POKE USR 444,1
458 POKE USR 445,1
459 POKE USR 446,1
460 POKE USR 447,1
461 POKE USR 448,1
462 POKE USR 449,1
463 POKE USR 450,1
46
```


- Para retornar al modo anterior, pulsa la tecla «9».

PRINT CHR\$ 144

Al estar situada la zona de memoria de los «GDU» por encima de una *variable del sistema* conocida por el nombre de «RAMTOP», al ejecutar una sentencia del tipo «NEW» se borra la zona de memoria destinada para almacenar nuestro programa BASIC, pero en cambio, permanecen *inalterables* nuestros gráficos, a no ser que desconectemos el ordenador o hagamos un RESET.

El programa número «1» nos permite conocer cuales son los GDU que tenemos almacenados en ese momento en el ordenador y a qué letras están asignados.

Esta aplicación nos permite

Grabación de GDU

La estructura de la sentencia «SAVE» es algo distinta de la utilizada en la grabación de programas, ya que debe especificarse en este caso la dirección de memoria a partir de la cual se desea grabar, así como la longitud en «bytes»:

SAVE (nombre) CODE comienzo,
longitud

AVF (adult) CODE USR (a) 168

Ave (odu2) CODE USR (C). 56

Para realizar el proceso inverso, es decir, almacenar en memoria los gráficos grabados en cinta, podemos utilizar cualquiera de estas opciones:

a) Si el siguiente programa a leer, grabado con «CODE» es el especificado:

LOAD -- CODE

LOAD (qdu2) CODE

LOAD (adu2) CODE USR (C)

Este método es más correcto ya que es independiente de la cantidad de memoria que posea el ordenador, por lo tanto el calcula la nueva dirección de carga.

LOAD (qdu2) CODE USR (c), 56

Lectura de los GDU

El programa número "3" nos visualiza en la pantalla las direcciones de cada "byte" de un GDU, así como su contenido.

El correspondiente gráfico nos aparece en la parte inferior de la pantalla.

Programa generador
de GDU

En las cintas demostración que acompañan tanto al Spectrum 16 o 48 K como el Plus, vienen grabados unos programas con los que se pueden generar con facilidad los GDU. Estos cuentan con una serie de opciones que permiten generar, modificar,

de la casa de los
señores de la casa
de la casa de los
señores de la casa
de la casa de los
señores de la casa
de la casa de los
señores de la casa

RIMAS 4 LEYENDAS / G. A. BÉCQUER

Aplicación de los «GDU» en la confección de los textos en español.

NOTES GRAPHIC 55

フ	ク
H	D
I	S
G	K
L	X
E	A
O	N
M	T
J	B

Notas gráficas Programa 4.

PROGRAMA 4

```

1 REM
*****
* CURSOR/BASIC *
*****
* PALITROQUE *
*****
*
3 PRINT FLASH 1,AT 10,8;"ESPE
4 REGARDE"
5 RESTORE : RANDOMIZE : GO SU
B 9000
4 BORDER 0 : PAPER 0 : INK 7 : C
L5 REM "MATHS"
5 POS=11: LET vide=4
10 LET salto=1: LET objet=0
15 LET borras= LET position=2
3 LET subic=0: LET pelando=0
14 REM "SELECCIONA MINUSCULAS"
15 POKE 2055,0
60 REM "LOTO"
70 FOR I=0 TO 15 STEP 4
80 PRINT PAPER 5: INK 3: AT 11,
90 NEXT I
100 NEXT I TO 15
110 PRINT PAPER 6: INK 2,AT 1,0
120 NEXT I
130 PRINT PAPER 6: INK 2,AT 1,1

```

Pantalla de información Programa 4.

```

1100 AT 2,4,""; AT 3,4; INVERSE
1110 PRINT PAPER 5; INK 3; AT 5,4
1120 AT 6,4,""; AT 7,4; INVERSE
1130 PRINT PAPER 6; INK 2; AT 9,2
1140 AT 10,27,""; AT 11,27; INVERSE
1150 PRINT PAPER 6; INK 3; AT 13,27
1160 AT 14,27,""; AT 15,27; INVERSE
1170 REM "ESCAPERA"
1180 LET X=15
1190 LET Y=INT (RND*7)+2
1200 GO SUB 8000
1210 LET Y=Y+1

```

64389	195
64390	36
64391	128
64392	219
64393	255
64394	66
64395	126
64396	219

GRAFICO "F":.

Ejemplo Programa 3.

PERSONALITIES

大 英
英 大

DELETED

- ♣ Espada Gloriosa
- ♠ Escudo Protector
- ♥ Corona Real
- ♣ Nectar de los Dioses
- ♠ Llaves


```

1000 LET X=INT (RAND*7)+13
1010 GO TO SUB 6000
1020 LET X=INT (RAND*7)+23
1030 GO TO SUB 6000
1040 REM CLAVES
1050 LET X=INT (RAND*14)+11
1060 PRINT BRIGHT 1; INK 2; AT 13
1070 GOTO 1000
1080 LET X=INT (RAND*7)+3
1090 PRINT BRIGHT 1; INK 3; AT 9,
1100 GOTO 1000
1110 LET X=INT (RAND*14)+7
1120 PRINT BRIGHT 1; INK 4; AT 5,
1130 GOTO 1000
1140 LET X=INT (RAND*22)+7
1150 PRINT BRIGHT 1; INK 5; AT 1,
1160 GOTO 1000
1170 REM OBJETOS
1180 PRINT INK 2; AT 15; 29; " "
1190 PRINT INK 3; AT 11; 3; 23; " "
1200 REM CONTADORES
1210 LET X=0
1220 LET Y=0
1230 LET Z=0
1240 REM CONTADORES
1250 FOR N=0 TO 31
1260 PRINT INK 2; AT 17; 0; " "
1270 GOTO 1000
1280 NEXT N
1290 LET N=8 TO 30
1300 PRINT INK 2; AT 0; 0; " "
1310 GOTO 1000
1320 NEXT N
1330 PRINT INK 1; AT 19; 1; "VIDAS"
1340 PRINT INK 4; "X"
1350 PRINT INK 1; AT 15; 5; "OBJETOS"
1360 GOTO 1000
1370 REM MOVIMIENTO
1380 LET PASO=0
1390 LET incremento=
1400 PRINT INK 5; AT 3; 20; PASO; " "
1410 LET PASO=PASO+1
1420 PRINT INK 5; AT 11; 3; PASO; " "
1430 IF PASO=30 THEN
1440 IF ATTR=0 THEN X=X+11; 6 OR ATTR
1450 X=X+11; 6 THEN GO SUB 8100
1460 IF INKEY$="q" THEN GO SUB 7
1470 IF SUB$="1" THEN GO TO 800
1480 IF INKEY$="p" THEN LET salt
1490 X=X+salt; 1 THEN LET salt
1500 IF OBJETOS=4 THEN LET
1510 GO TO 1000
1520 IF INKEY$="a" THEN GO SUB 8
1530 IF INKEY$="s" THEN GO TO 50
1540 GO TO 600
1550 IF INKEY$="c" THEN GO TO 60
1560 GO TO 500
1570 PRINT AT 3; 20; PASO; " "
1580 PRINT AT 11; 3; PASO; " "
1590 LET PASO=PASO+incremento
1600 IF PASO=30 THEN LET incremento
1610 IF PASO=0 THEN LET incremento
1620 GO TO 510
1630 REM CLAVES
1640 LET Z=0
1650 FOR N=0 TO 10 STEP -1
1660 PRINT INK 2; AT 13; 20; PASO; " "
1670 PRINT INK 3; AT 11; 3; PASO; " "
1680 LET PASO=PASO+incremento
1690 IF PASO=30 THEN LET incremento
1700 IF PASO=0 THEN LET incremento
1710 GO TO 510
1720 REM CLAVES
1730 PRINT AT 3; 20; PASO; " "
1740 PRINT AT 11; 3; PASO; " "
1750 IF INKEY$="a" THEN GO TO 1000
1760 PRINT AT 4; 2; 0; "siento, tu
1770 este motivo e insuflante
1780 para que me des un key"
1790 LET CLAVES=CLAVES+1
1800 IF CLAVES=30 THEN
1810 GO TO 510
1820 GOTO 1000

```

[illegible]

```

000100 LET salto=1
000110 RETURN
000120 REM SUB: SALTO
000130 GO TO 1100
000140 IF (ATR (YF+1,XF)+salto)=50 TH
000150 RETURN
000160 PRINT 8100
000170 BRIGHT 1,AT Y
000180 LET X=XF+salto; LET Y=YF+
000190
000200 PRINT OVER 1, INK 4,AT YF,X
000210
000220 IF ATTR (YF+1,XF)=50 OR ATTR
000230 (YF,XF+salto)=50 THEN GO SUB 81
000240 RETURN
000250 GO SUB 5500
000260 PRINT OVER 1, BRIGHT 1,AT Y
000270
000280 LET X=XF+salto; LET Y=YF+
000290
000300 PRINT OVER 1, INK 4,AT YF,X
000310
000320 IF ATTR (YF+1,XF)+salto)=50 TH
000330 RETURN
000340 GO SUB 8100
000350 PRINT OVER 1, BRIGHT 1,AT Y
000360
000370 LET X=XF+salto; LET Y=YF+
000380
000390 PRINT OVER 1, INK 4,AT YF,X
000400
000410 IF ATTR (YF+1,XF)+salto)=50 TH
000420 RETURN
000430 GO SUB 8100
000440 PRINT OVER 1, BRIGHT 1,AT Y
000450
000460 LET X=XF+salto; LET Y=YF+
000470
000480 PRINT OVER 1, INK 4,AT YF,X
000490
000500 IF ATTR (YF+1,XF)+salto)=50 OR ATTR
000510 (YF,XF+salto)=50 THEN GO SUB 81
000520 RETURN
000530 PRINT OVER 1, BRIGHT 1,AT Y
000540
000550 LET X=XF+salto; LET Y=YF+
000560
000570 PRINT OVER 1, INK 4,AT YF,X
000580
000590 IF ATTR (YF+1,XF)+salto)=50 TH
000600 RETURN
000610 GO SUB 8100
000620 PRINT OVER 1, BRIGHT 1,AT Y
000630
000640 LET X=XF+salto; LET Y=YF+
000650
000660 PRINT OVER 1, INK 4,AT YF,X
000670
000680 IF ATTR (YF+1,XF)+salto)=50 TH
000690 RETURN
000700 GO SUB 8100
000710 PRINT OVER 1, BRIGHT 1,AT Y
000720
000730 LET X=XF+salto; LET Y=YF+
000740
000750 PRINT OVER 1, INK 4,AT YF,X
000760
000770 IF ATTR (YF+1,XF)+salto)=50 TH
000780 RETURN
000790 GO SUB 8100
000800 PRINT OVER 1, BRIGHT 1,AT Y
000810
000820 LET X=XF+salto; LET Y=YF+
000830
000840 PRINT OVER 1, INK 4,AT YF,X
000850
000860 IF ATTR (YF+1,XF)+salto)=50 TH
000870 RETURN
000880 GO SUB 8100
000890 PRINT OVER 1, BRIGHT 1,AT Y
000900
000910 LET X=XF+salto; LET Y=YF+
000920
000930 PRINT OVER 1, INK 4,AT YF,X
000940
000950 IF ATTR (YF+1,XF)+salto)=50 TH
000960 RETURN
000970 GO SUB 8100
000980 PRINT OVER 1, BRIGHT 1,AT Y
000990
001000 LET X=XF+salto; LET Y=YF+
001010
001020 PRINT OVER 1, INK 4,AT YF,X
001030
001040 IF ATTR (YF+1,XF)+salto)=50 TH
001050 RETURN
001060 GO SUB 8100
001070 PRINT OVER 1, BRIGHT 1,AT Y
001080
001090 LET X=XF+salto; LET Y=YF+
001100
001110 PRINT OVER 1, INK 4,AT YF,X
001120
001130 IF ATTR (YF+1,XF)+salto)=50 TH
001140 RETURN
001150 GO SUB 8100
001160 PRINT OVER 1, BRIGHT 1,AT Y
001170
001180 LET X=XF+salto; LET Y=YF+
001190
001200 PRINT OVER 1, INK 4,AT YF,X
001210
001220 IF ATTR (YF+1,XF)+salto)=50 TH
001230 RETURN
001240 GO SUB 8100
001250 PRINT OVER 1, BRIGHT 1,AT Y
001260
001270 LET X=XF+salto; LET Y=YF+
001280
001290 PRINT OVER 1, INK 4,AT YF,X
001300
001310 IF ATTR (YF+1,XF)+salto)=50 TH
001320 RETURN
001330 GO SUB 8100
001340 PRINT OVER 1, BRIGHT 1,AT Y
001350
001360 LET X=XF+salto; LET Y=YF+
001370
001380 PRINT OVER 1, INK 4,AT YF,X
001390
001400 IF ATTR (YF+1,XF)+salto)=50 TH
001410 RETURN
001420 GO SUB 8100
001430 PRINT OVER 1, BRIGHT 1,AT Y
001440
001450 LET X=XF+salto; LET Y=YF+
001460
001470 PRINT OVER 1, INK 4,AT YF,X
001480
001490 IF ATTR (YF+1,XF)+salto)=50 TH
001500 RETURN
001510 GO SUB 8100
001520 PRINT OVER 1, BRIGHT 1,AT Y
001530
001540 LET X=XF+salto; LET Y=YF+
001550
001560 PRINT OVER 1, INK 4,AT YF,X
001570
001580 IF ATTR (YF+1,XF)+salto)=50 TH
001590 RETURN
001600 GO SUB 8100
001610 PRINT OVER 1, BRIGHT 1,AT Y
001620
001630 LET X=XF+salto; LET Y=YF+
001640
001650 PRINT OVER 1, INK 4,AT YF,X
001660
001670 IF ATTR (YF+1,XF)+salto)=50 TH
001680 RETURN
001690 GO SUB 8100
001700 PRINT OVER 1, BRIGHT 1,AT Y
001710
001720 LET X=XF+salto; LET Y=YF+
001730
001740 PRINT OVER 1, INK 4,AT YF,X
001750
001760 IF ATTR (YF+1,XF)+salto)=50 TH
001770 RETURN
001780 GO SUB 8100
001790 PRINT OVER 1, BRIGHT 1,AT Y
001800
001810 LET X=XF+salto; LET Y=YF+
001820
001830 PRINT OVER 1, INK 4,AT YF,X
001840
001850 IF ATTR (YF+1,XF)+salto)=50 TH
001860 RETURN
001870 GO SUB 8100
001880 PRINT OVER 1, BRIGHT 1,AT Y
001890
001900 LET X=XF+salto; LET Y=YF+
001910
001920 PRINT OVER 1, INK 4,AT YF,X
001930
001940 IF ATTR (YF+1,XF)+salto)=50 TH
001950 RETURN
001960 GO SUB 8100
001970 PRINT OVER 1, BRIGHT 1,AT Y
001980
001990 LET X=XF+salto; LET Y=YF+
002000
002010 PRINT OVER 1, INK 4,AT YF,X
002020
002030 IF ATTR (YF+1,XF)+salto)=50 TH
002040 RETURN
002050 GO SUB 8100
002060 PRINT OVER 1, BRIGHT 1,AT Y
002070
002080 LET X=XF+salto; LET Y=YF+
002090
002100 PRINT OVER 1, INK 4,AT YF,X
002110
002120 IF ATTR (YF+1,XF)+salto)=50 TH
002130 RETURN
002140 GO SUB 8100
002150 PRINT OVER 1, BRIGHT 1,AT Y
002160
002170 LET X=XF+salto; LET Y=YF+
002180
002190 PRINT OVER 1, INK 4,AT YF,X
002200
002210 IF ATTR (YF+1,XF)+salto)=50 TH
002220 RETURN
002230 GO SUB 8100
002240 PRINT OVER 1, BRIGHT 1,AT Y
002250
002260 LET X=XF+salto; LET Y=YF+
002270
002280 PRINT OVER 1, INK 4,AT YF,X
002290
002300 IF ATTR (YF+1,XF)+salto)=50 TH
002310 RETURN
002320 GO SUB 8100
002330 PRINT OVER 1, BRIGHT 1,AT Y
002340
002350 LET X=XF+salto; LET Y=YF+
002360
002370 PRINT OVER 1, INK 4,AT YF,X
002380
002390 IF ATTR (YF+1,XF)+salto)=50 TH
002400 RETURN
002410 GO SUB 8100
002420 PRINT OVER 1, BRIGHT 1,AT Y
002430
002440 LET X=XF+salto; LET Y=YF+
002450
002460 PRINT OVER 1, INK 4,AT YF,X
002470
002480 IF ATTR (YF+1,XF)+salto)=50 TH
002490 RETURN
002500 GO SUB 8100
002510 PRINT OVER 1, BRIGHT 1,AT Y
002520
002530 LET X=XF+salto; LET Y=YF+
002540
002550 PRINT OVER 1, INK 4,AT YF,X
002560
002570 IF ATTR (YF+1,XF)+salto)=50 TH
002580 RETURN
002590 GO SUB 8100
002600 PRINT OVER 1, BRIGHT 1,AT Y
002610
002620 LET X=XF+salto; LET Y=YF+
002630
002640 PRINT OVER 1, INK 4,AT YF,X
002650
002660 IF ATTR (YF+1,XF)+salto)=50 TH
002670 RETURN
002680 GO SUB 8100
002690 PRINT OVER 1, BRIGHT 1,AT Y
002700
002710 LET X=XF+salto; LET Y=YF+
002720
002730 PRINT OVER 1, INK 4,AT YF,X
002740
002750 IF ATTR (YF+1,XF)+salto)=50 TH
002760 RETURN
002770 GO SUB 8100
002780 PRINT OVER 1, BRIGHT 1,AT Y
002790
002800 LET X=XF+salto; LET Y=YF+
002810
002820 PRINT OVER 1, INK 4,AT YF,X
002830
002840 IF ATTR (YF+1,XF)+salto)=50 TH
002850 RETURN
002860 GO SUB 8100
002870 PRINT OVER 1, BRIGHT 1,AT Y
002880
002890 LET X=XF+salto; LET Y=YF+
002900
002910 PRINT OVER 1, INK 4,AT YF,X
002920
002930 IF ATTR (YF+1,XF)+salto)=50 TH
002940 RETURN
002950 GO SUB 8100
002960 PRINT OVER 1, BRIGHT 1,AT Y
002970
002980 LET X=XF+salto; LET Y=YF+
002990
003000 PRINT OVER 1, INK 4,AT YF,X

```

```

90600 DATA 4,10,17,10,20,32,30,33
90700 DATA 0,10,4,10,15,32,60,0
90800 DATA 0,10,4,60,20,32,60,0
90900 DATA 195,36,126,23,255,66
91000 DATA 24,36,126,25,24,60,36
91100 DATA 0,0,0,155,153,90,126,1
91200 DATA 124,124,124,124,56,16
91300 DATA 127,119,119,65,119,119
91400 RETURN
91500 REM CARATULA
91600 DRAW 0,1,5: DRAW 255,0
91700 DRAW 0,1,75: DRAW 255,0
91800 PRINT AT 3,0:
91900 REM HOBBY
92000 PRINT AT 18,0:
92100 PRINT AT 7,11: "PALATROQUE"
92200 PRINT "LOS,AT 13,10,"PROG
92300 PRINT "AT 1,0,"QUIERES LA
92400 PRINT "INSTRUCCIONES (S/N)"; PAUSE 0
92500 IF INKEY$="N" THEN CLS: RE
TURN
92600 IF INKEY$="S" THEN GO TO 93
2000
92700 GO TO 9280
92800 CLS TO 9280
92900 PRINT AT 0,2: "Palatroque, e
ste juego se trata de jugar a los 3
3000 PRINT "Para recoger las arm
as y objetos los personajes de la s
3100 "y Gumbesindoo, porque se va a
3200 "las Cruzadas."
93400 PRINT "La mision de Palit
roque no es tan sencilla, ya que en
3300 "los niveles 1, 2 y 3, a veces se encuent
ran monstruos."
93500 PRINT "Palatroque debera cu
rarse de las "Llaves y tener cu
93600 "idado con los "Terribles Trogl
oditos."
93700 PRINT "0:AT 1,1:"Pulsa una
tecla para continuar"; PAUSE 0
93800 PRINT AT 0,4: "PERSONAJES."
93900 PRINT BRIGHT 1, INK 1, AT 3,
94000 "Palatroque"
94100 PRINT BRIGHT 1, INK 6, AT 5,
94200 "Trogloditos 1, INK 6, AT 5,
94300 PRINT AT 9,4: "OBJETOS."
94400 PRINT BRIGHT 1, INK 2, AT 11
94500 "Espada color 1."
94600 PRINT BRIGHT 1, INK 3, AT 13
94700 "Palo de hierro color 1."
94800 PRINT BRIGHT 1, INK 4, AT 15
94900 "Cruzada 1."
95000 PRINT BRIGHT 1, INK 5, AT 17
95100 "Nectar de los Dioses."
95200 PRINT BRIGHT 1, INK 6, AT 19
95300 "Llaves."
95400 PRINT "0:AT 1,1:"Pulsa una
tecla para continuar"; PAUSE 0
95500 PRINT AT 0,4: "CONTROLES"
95600 PRINT AT 1,4: "IZQUIERDA"
95700 PRINT AT 5,4: "P - DERECHA"
95800 PRINT AT 7,4: "Q - SUBIR"
95900 PRINT AT 9,4: "A - SALTAR"
96000 PRINT AT 12,4: "S - Para el
nuevo
96100 PRINT AT 14,4: "C - Continua
96200 PRINT "0:AT 1,1:"Pulsa una
tecla para comenzar"; PAUSE 0
96300 CLS: RETURN

```


PROGRAMA 9

```

10 REM
*****
* CURSO/BASIC *
*****
* NAVIDAD *
* (TREMOLADO) *
*****

```

```

30 FOR C=-12 TO 0 STEP 12
40 READ nota,duracion
50 IF nota THEN RESTORE
60 PAUSE 50: GO TO 70
70 LET J=duracion/0.12
80 FOR X=1 TO J
90 BEEP 0.01,nota+c
100 BORDER 2
110 BORDER 7
120 NEXT X
130 GO TO 30
140 NEXT C

```

ción relativa se refiere, cuatro negras equivalen a una redonda, a dos blancas, a ocho corcheas o a dieciséis semicorcheas.

Compás

El compás es la separación entre unidades rítmicas, y viene determinado por dos números; el primero, situado en la parte superior del pentagrama, indica el número de partes de un compás y el segundo, situado en la parte inferior, la duración de una de las partes de acuerdo con la siguiente tabla:

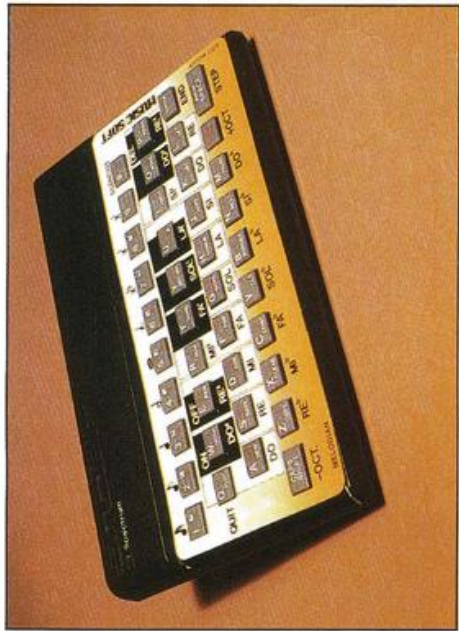
NÚMERO	DURACIÓN
2	Blanca
4	Negra
8	Corchea

Ejemplo:

En un compás «dos por cuatro», es decir, compuesto por dos partes y cada una con una duración equivalente a una negra, podrían componer el compás cualquiera de las siguientes combinaciones:

- Una blanca
- Dos negras
- Cuatro corcheas
- Una negra y dos corcheas
- Etc.

Utilizando la técnica del programa número «4», para



Teclado musical simulado.

definir las notas y su duración relativa, puede programar cualquier melodía a través de un pentagrama.

Si desea modificar la duración de las notas o la escala, puede hacerlo variando el contenido de las variables «duración» y «escala», teniendo en cuenta que esta última debe tener un valor múltiplo de doce, bien positivo o negativo.

Variables relacionadas

Dentro de las variables del sistema existen dos relacionadas con el sonido, la primera, denominada «RASP», se encuentra localizada en la dirección 23608 y controla la duración del zumbador de

bucle esperando que se pulse la tecla «C» (continuar).

600 : Borrado de los «trogloditos».

610 : Incremento de la posición de los «trogloditos».

620-630 : Calcula si el próximo movimiento de los «trogloditos» debe ser a la derecha o a la izquierda.

640 : Salto a la visualización de la nueva posición de los «trogloditos».

1000 : Fin del juego.

1002-1040 : Efectos de color en el borde de la pantalla.

1050 : Borrado de la pantalla.

1060 : Comprueba por que motivo se ha terminado el juego.

1070 : Mensaje en caso de haberse quedado «Paltiroque» sin vidas.

1090 : Mensaje en caso de haberse completado la misión.

1100-1130 : Comprueba si se desea jugar otra vez o no.

7000 : Comienzo de la subrutina «detecta».

7010 : Comprueba si la nueva posición está libre o es una escalera.

7022 : Comprueba si la puerta de la habitación está abierta.

7024 : Llamada a la subrutina de coger objetos.

7030-7050 : Borra la antigua posición de «Paltiroque» y visualiza la nueva.

7100 : Comienzo de la subrutina de coger objetos.

7110 : Comprueba si el objeto cogido es la espada global.

8190 : Inicializa la posición de origen de «Paltiroque».

8200 : Visualiza a «Paltiroque».

8207-8209 : Inicialización de las variables.

8220 : Comienzo de la subrutina salto.

8222-8430 : Comprueba si durante el salto «Paltiroque» choca con el muro, es atrapado por algún «troglodito» o recoge alguna llave.

8500 : Comienzo de la subrutina de recogida de llaves.

8510-8520 : Comprueban que se recoge la llave del piso inferior.

8530-8540 : Idem. del piso primero.

8550-8560 : Idem. del piso segundo.

8570-8580 : Idem. del piso superior.

8600 : Papeado de la cerradura del piso inferior.

8610 : Idem. del piso primero.

8620 : Idem. del piso segundo.

8630 : Idem. del piso superior.

9000 : Comienzo de la subrutina que genera los gráficos.

9002-9030 : Bucle para la lectura y generación de los GDU.

9040-9130 : Tabla con los datos de los GDU.

9200 : Subrutina «Carátula e instrucciones».

9210-9230 : Recuerdo de la pantalla.

9240-9260 : Mensajes de pantalla.

9270-9300 : Comprueba si se desea visualizar las instrucciones.

9320-9360 : Primera pantalla de información.

9380-9490 : Segunda pantalla de información.

9510-9590 : Última pantalla de información.

alarmas, esta se activa, por ejemplo, cuando intentamos editar una sentencia de más de 22 líneas.

La otra variable, denominada «PIP», y localizada en la dirección 23609, controla la duración del chasquido del teclado al pulsar una tecla.

Introduzca el siguiente comando directo y observe la diferencia al pulsar cualquier tecla:

POKE 23609, 30

Estas dos variables pueden tener cualquier valor comprendido entre 0 y 255.

Grabación de sonidos

Si conecta un grabador de

SONIDO

El Spectrum tiene capacidad para producir una amplia variedad de notas musicales, con el manejo de una sola instrucción.

Los sonidos son escuchados a través de un altavoz interno, por lo tanto, el volumen es relativamente bajo, aunque puede mejorarse con el empleo de ciertos periféricos.

En un ordenador personal esta característica es muy apreciada, ya que da vida a ciertos programas que en un principio podrían parecerse «sosos».

Introduzca en el programa del capítulo anterior, «Palitro» que y los trogloditos», las siguientes instrucciones y observe las diferencias:

a) Movimiento de los Trogloditos.

```
511 BEEP 0.01, -12
602 BEEP 0.01, 0
```

b) Pérdida de una vida.

```
8122 BEEP 0.01, 12
8132 BEEP 0.01, 24
```

c) Recogida de una llave. Antes de la sentencia «RE-TURN» de las líneas de programa.

```
8600
8610
8620
8630
```

incluir

258 MICROBASIC



Amplificador de sonido.

VARIABLE	DIRECCION	VALOR INICIAL
RASP	23608	64
PIP	23609	0
RASP: Zumbador de alarma PIP: Chasquido del teclado		

Tabla I.

Acceso al teclado

BEEP 0.01, 48

d) Fin de juego.

```
1012 BEEP 0.01, 2
1014 BEEP 0.01, n
```

BEEP

LN

MODO E



BEEP SYMBOL SHIFT

vas hay dos semitonos, excepto entre «Mi» - «Fa» y «Si» - «Do» que en ambos casos hay un solo semitono.

En un pentagrama cada nota viene identificada por la posición que ocupa dentro del mismo.

El programa número «2» convierte el teclado del Spectrum en un piano con escala diatónica y el número «3», en un cromático. Las teclas que deben pulsarse en cada caso están contenidas en las instrucciones del propio programa.

PROGRAMA 5

```
10 REM
**
** CURSO/BASIC
**
**
** EFECTO 1
**
**
20 FOR i=30 TO 0 STEP -1
30 BEEP :005,i
40 NEXT i
```

PROGRAMA 7

```
10 REM
**
** CURSO/BASIC
**
**
** EFECTO 3
**
**
20 FOR a=1 TO 25
30 FOR b=1 TO 10
40 BEEP 0.05,INT (RND*b) -12
50 BEEP 0.05,INT (RND*b) +12
60 NEXT b
70 NEXT a
```

Duración

La duración de la nota viene identificada por unos símbolos característicos. Los más utilizados son:

REDONDA
BLANCA
NEGRA
CORCHEA
SEMICORCHEA

PROGRAMA 6

```
10 REM
**
** CURSO/BASIC
**
**
** EFECTO 2
**
**
20 FOR a=30 TO 0 STEP -1
30 BEEP :05,20-a
40 NEXT a
50 BEEP :5,-10
```

PROGRAMA 8

```
10 REM
**
** CURSO/BASIC
**
**
** EFECTO 4
**
**
20 FOR a=1 TO 10
30 FOR b=1 TO 1 STEP -1
40 BEEP 0.05,a
50 BEEP 0.05,b
60 NEXT b
70 NEXT a
```

La que más dura es la redonda, la blanca es la mitad que la redonda, la negra es la mitad que la blanca y así sucesivamente.

Portanto, en cuanto a dura-

En un piano las teclas negras son los correspondientes a los sostenidos.

La unidad máxima de altura tonal es el *semitono*. Un semitono es el resultado de dividir una *octrava* en doce partes iguales.

Entre dos notas consecuti-

El símbolo # es el *identificador del sostenido*.

En cambio en la escala *cromática* son doce:

Entre dos notas consecuti-

The diagram illustrates the C major scale. On the left, a musical staff shows the notes C, D, E, F, G, A, B, and C, each with a corresponding solfège syllable (DO, RE, MI, FA, SOL, LA, SI) written below it. To the right of the staff, a piano keyboard diagram shows the keys for each note: DO (white), RE (white), MI (black), FA (white), SOL (white), LA (black), SI (white), and DO (white). Arrows indicate the sequence of notes in the scale.

262 MICROBASIC

MICROBASIC 259

Sintetizador de voz.

[illegible]

```


INKEY$="u");(11 AND INKEY$="j")
80 IF TONO=0 AND INKEY$="q") T
    ENO GO TO 50
    70 BEEP
    80 GO TO 50
    9000 REM *****
    9001 REM PRINT AT 0,8;"ESCALA CROMAT
    9002 REM PRINT AT 0,8;"
    9003 REM PRINT AT 12,0;"do
    9004 REM PRINT AT 14,1;"a
    9005 REM PRINT AT 16,2;"b
    9006 REM PRINT AT 18,3;"c
    9007 REM PRINT AT 20,4;"d
    9008 REM PRINT AT 22,5;"e
    9009 REM PRINT AT 24,6;"f
    9010 REM PRINT AT 26,7;"g
    9011 REM PRINT AT 28,8;"a
    9012 REM PRINT AT 30,9;"b
    9013 REM PRINT AT 32,10;"c
    9014 REM PRINT AT 34,11;"d
    9015 REM PRINT AT 36,12;"e
    9016 REM PRINT AT 38,13;"f
    9017 REM PRINT AT 40,14;"g
    9018 REM PRINT AT 42,15;"a
    9019 REM PRINT AT 44,16;"b
    9020 REM PRINT AT 46,17;"c
    9021 REM PRINT AT 48,18;"d
    9022 REM PRINT AT 50,19;"e
    9023 REM PRINT AT 52,20;"f
    9024 REM PRINT AT 54,21;"g
    9025 REM PRINT AT 56,22;"a
    9026 REM PRINT AT 58,23;"b
    9027 REM PRINT AT 60,24;"c
    9028 REM PRINT AT 62,25;"d
    9029 REM PRINT AT 64,26;"e
    9030 REM PRINT AT 66,27;"f
    9031 REM PRINT AT 68,28;"g
    9032 REM PRINT AT 70,29;"a
    9033 REM PRINT AT 72,30;"b
    9034 REM PRINT AT 74,31;"c
    9035 REM PRINT AT 76,32;"d
    9036 REM PRINT AT 78,33;"e
    9037 REM PRINT AT 80,34;"f
    9038 REM PRINT AT 82,35;"g
    9039 REM PRINT AT 84,36;"a
    9040 REM PRINT AT 86,37;"b
    9041 REM PRINT AT 88,38;"c
    9042 REM PRINT AT 90,39;"d
    9043 REM PRINT AT 92,40;"e
    9044 REM PRINT AT 94,41;"f
    9045 REM PRINT AT 96,42;"g
    9046 REM PRINT AT 98,43;"a
    9047 REM PRINT AT 100,44;"b
    9048 REM PRINT AT 102,45;"c
    9049 REM PRINT AT 104,46;"d
    9050 REM PRINT AT 106,47;"e
    9051 REM PRINT AT 108,48;"f
    9052 REM PRINT AT 110,49;"g
    9053 REM PRINT AT 112,50;"a
    9054 REM PRINT AT 114,51;"b
    9055 REM PRINT AT 116,52;"c
    9056 REM PRINT AT 118,53;"d
    9057 REM PRINT AT 120,54;"e
    9058 REM PRINT AT 122,55;"f
    9059 REM PRINT AT 124,56;"g
    9060 REM PRINT AT 126,57;"a
    9061 REM PRINT AT 128,58;"b
    9062 REM PRINT AT 130,59;"c
    9063 REM PRINT AT 132,60;"d
    9064 REM PRINT AT 134,61;"e
    9065 REM PRINT AT 136,62;"f
    9066 REM PRINT AT 138,63;"g
    9067 REM PRINT AT 140,64;"a
    9068 REM PRINT AT 142,65;"b
    9069 REM PRINT AT 144,66;"c
    9070 REM PRINT AT 146,67;"d
    9071 REM PRINT AT 148,68;"e
    9072 REM PRINT AT 150,69;"f
    9073 REM PRINT AT 152,70;"g
    9074 REM PRINT AT 154,71;"a
    9075 REM PRINT AT 156,72;"b
    9076 REM PRINT AT 158,73;"c
    9077 REM PRINT AT 160,74;"d
    9078 REM PRINT AT 162,75;"e
    9079 REM PRINT AT 164,76;"f
    9080 REM PRINT AT 166,77;"g
    9081 REM PRINT AT 168,78;"a
    9082 REM PRINT AT 170,79;"b
    9083 REM PRINT AT 172,80;"c
    9084 REM PRINT AT 174,81;"d
    9085 REM PRINT AT 176,82;"e
    9086 REM PRINT AT 178,83;"f
    9087 REM PRINT AT 180,84;"g
    9088 REM PRINT AT 182,85;"a
    9089 REM PRINT AT 184,86;"b
    9090 REM PRINT AT 186,87;"c
    9091 REM PRINT AT 188,88;"d
    9092 REM PRINT AT 190,89;"e
    9093 REM PRINT AT 192,90;"f
    9094 REM PRINT AT 194,91;"g
    9095 REM PRINT AT 196,92;"a
    9096 REM PRINT AT 198,93;"b
    9097 REM PRINT AT 200,94;"c
    9098 REM PRINT AT 202,95;"d
    9099 REM PRINT AT 204,96;"e
    9100 REM PRINT AT 206,97;"f
    9101 REM PRINT AT 208,98;"g
    9102 REM PRINT AT 210,99;"a
    9103 REM PRINT AT 212,100;"b
    9104 REM PRINT AT 214,101;"c
    9105 REM PRINT AT 216,102;"d
    9106 REM PRINT AT 218,103;"e
    9107 REM PRINT AT 220,104;"f
    9108 REM PRINT AT 222,105;"g
    9109 REM PRINT AT 224,106;"a
    9110 REM PRINT AT 226,107;"b
    9111 REM PRINT AT 228,108;"c
    9112 REM PRINT AT 230,109;"d
    9113 REM PRINT AT 232,110;"e
    9114 REM PRINT AT 234,111;"f
    9115 REM PRINT AT 236,112;"g
    9116 REM PRINT AT 238,113;"a
    9117 REM PRINT AT 240,114;"b
    9118 REM PRINT AT 242,115;"c
    9119 REM PRINT AT 244,116;"d
    9120 REM PRINT AT 246,117;"e
    9121 REM PRINT AT 248,118;"f
    9122 REM PRINT AT 250,119;"g
    9123 REM PRINT AT 252,120;"a
    9124 REM PRINT AT 254,121;"b
    9125 REM PRINT AT 256,122;"c
    9126 REM PRINT AT 258,123;"d
    9127 REM PRINT AT 260,124;"e
    9128 REM PRINT AT 262,125;"f
    9129 REM PRINT AT 264,126;"g
    9130 REM PRINT AT 266,127;"a
    9131 REM PRINT AT 268,128;"b
    9132 REM PRINT AT 270,129;"c
    9133 REM PRINT AT 272,130;"d
    9134 REM PRINT AT 274,131;"e
    9135 REM PRINT AT 276,132;"f
    9136 REM PRINT AT 278,133;"g
    9137 REM PRINT AT 280,134;"a
    9138 REM PRINT AT 282,135;"b
    9139 REM PRINT AT 284,136;"c
    9140 REM PRINT AT 286,137;"d
    9141 REM PRINT AT 288,138;"e
    9142 REM PRINT AT 290,139;"f
    9143 REM PRINT AT 292,140;"g
    9144 REM PRINT AT 294,141;"a
    9145 REM PRINT AT 296,142;"b
    9146 REM PRINT AT 298,143;"c
    9147 REM PRINT AT 300,144;"d
    9148 REM PRINT AT 302,145;"e
    9149 REM PRINT AT 304,146;"f
    9150 REM PRINT AT 306,147;"g
    9151 REM PRINT AT 308,148;"a
    9152 REM PRINT AT 310,149;"b
    9153 REM PRINT AT 312,150;"c
    9154 REM PRINT AT 314,151;"d
    9155 REM PRINT AT 316,152;"e
    9156 REM PRINT AT 318,153;"f
    9157 REM PRINT AT 320,154;"g
    9158 REM PRINT AT 322,155;"a
    9159 REM PRINT AT 324,156;"b
    9160 REM PRINT AT 326,157;"c
    9161 REM PRINT AT 328,158;"d
    9162 REM PRINT AT 330,159;"e
    9163 REM PRINT AT 332,160;"f
    9164 REM PRINT AT 334,161;"g
    9165 REM PRINT AT 336,162;"a
    9166 REM PRINT AT 338,163;"b
    9167 REM PRINT AT 340,164;"c
    9168 REM PRINT AT 342,165;"d
    9169 REM PRINT AT 344,166;"e
    9170 REM PRINT AT 346,167;"f
    9171 REM PRINT AT 348,168;"g
    9172 REM PRINT AT 350,169;"a
    9173 REM PRINT AT 352,170;"b
    9174 REM PRINT AT 354,171;"c
    9175 REM PRINT AT 356,172;"d
    9176 REM PRINT AT 358,173;"e
    9177 REM PRINT AT 360,174;"f
    9178 REM PRINT AT 362,175;"g
    9179 REM PRINT AT 364,176;"a
    9180 REM PRINT AT 366,177;"b
    9181 REM PRINT AT 368,178;"c
    9182 REM PRINT AT 370,179;"d
    9183 REM PRINT AT 372,180;"e
    9184 REM PRINT AT 374,181;"f
    9185 REM PRINT AT 376,182;"g
    9186 REM PRINT AT 378,183;"a
    9187 REM PRINT AT 380,184;"b
    9188 REM PRINT AT 382,185;"c
    9189 REM PRINT AT 384,186;"d
    9190 REM PRINT AT 386,187;"e
    9191 REM PRINT AT 388,188;"f
    9192 REM PRINT AT 390,189;"g
    9193 REM PRINT AT 392,190;"a
    9194 REM PRINT AT 394,191;"b
    9195 REM PRINT AT 396,192;"c
    9196 REM PRINT AT 398,193;"d
    9197 REM PRINT AT 400,194;"e
    9198 REM PRINT AT 402,195;"f
    9199 REM PRINT AT 404,196;"g
    9200 REM PRINT AT 406,197;"a
    9201 REM PRINT AT 408,198;"b
    9202 REM PRINT AT 410,199;"c
    9203 REM PRINT AT 412,200;"d
    9204 REM PRINT AT 414,201;"e
    9205 REM PRINT AT 416,202;"f
    9206 REM PRINT AT 418,203;"g
    9207 REM PRINT AT 420,204;"a
    9208 REM PRINT AT 422,205;"b
    9209 REM PRINT AT 424,206;"c
    9
```

```

1  REM
2  *****
3  *      CURSO/BASIC      *
4  *      *                *
5  *      *****        *
6  *      *                *
7  *      DOS CRUCES      *
8  *      *                *
9  *      *****        *
10 *****
11
12 BORUS 2 : PAPER 2 : CLS
13 CUR 1,0,10,10
14 CUR 1,1,12
15 CUR 1,1,12
16 *****
17 *      DEFINITION NOTES *
18 *      *                *
19 *      00=esala        *
20 *      05=escala+1     *
21 *      10=escala+2     *
22 *      15=escala+3     *
23 *      20=escala+4     *
24 *      25=escala+5     *
25 *      30=escala+6     *
26 *      35=escala+7     *
27 *      40=escala+8     *
28 *      45=escala+9     *
29 *      50=escala+10    *
30 *****
31
32 100 LET T=0
33 110 IF T=0 THEN GOTO 140
34 120 LET T=T+1
35 130 LET T=T+1
36 140 LET T=T+1
37 150 LET T=T+1
38 160 LET T=T+1
39 170 LET T=T+1
40 180 LET T=T+1
41 190 LET T=T+1
42 200 LET T=T+1
43 210 LET T=T+1
44 220 LET T=T+1
45 230 LET T=T+1
46 240 LET T=T+1
47 250 LET T=T+1
48 260 LET T=T+1
49 270 LET T=T+1
50 280 LET T=T+1
51 290 LET T=T+1
52 300 LET T=T+1
53 310 LET T=T+1
54 320 LET T=T+1
55 330 LET T=T+1
56 340 LET T=T+1
57 350 LET T=T+1
58 360 LET T=T+1
59 370 LET T=T+1
60 380 LET T=T+1
61 390 LET T=T+1
62 400 LET T=T+1
63 410 LET T=T+1
64 420 LET T=T+1
65 430 LET T=T+1
66 440 LET T=T+1
67 450 LET T=T+1
68 460 LET T=T+1
69 470 LET T=T+1
70 480 LET T=T+1
71 490 LET T=T+1
72 500 LET T=T+1
73 510 LET T=T+1
74 520 LET T=T+1
75 530 LET T=T+1
76 540 LET T=T+1
77 550 LET T=T+1
78 560 LET T=T+1
79 570 LET T=T+1
80 580 LET T=T+1
81 590 LET T=T+1
82 600 LET T=T+1
83 610 LET T=T+1
84 620 LET T=T+1
85 630 LET T=T+1
86 640 LET T=T+1
87 650 LET T=T+1
88 660 LET T=T+1
89 670 LET T=T+1
90 680 LET T=T+1
91 690 LET T=T+1
92 700 LET T=T+1
93 710 LET T=T+1
94 720 LET T=T+1
95 730 LET T=T+1
96 740 LET T=T+1
97 750 LET T=T+1
98 760 LET T=T+1
99 770 LET T=T+1
100 780 LET T=T+1
101 790 LET T=T+1
102 800 LET T=T+1
103 810 LET T=T+1
104 820 LET T=T+1
105 830 LET T=T+1
106 840 LET T=T+1
107 850 LET T=T+1
108 860 LET T=T+1
109 870 LET T=T+1
110 880 LET T=T+1
111 890 LET T=T+1
112 900 LET T=T+1
113 910 LET T=T+1
114 920 LET T=T+1
115 930 LET T=T+1
116 940 LET T=T+1
117 950 LET T=T+1
118 960 LET T=T+1
119 970 LET T=T+1
120 980 LET T=T+1
121 990 LET T=T+1
122 1000 LET T=T+1
123 1010 LET T=T+1
124 1020 LET T=T+1
125 1030 LET T=T+1
126 1040 LET T=T+1
127 1050 LET T=T+1
128 1060 LET T=T+1
129 1070 LET T=T+1
130 1080 LET T=T+1
131 1090 LET T=T+1
132 1100 LET T=T+1
133 1110 LET T=T+1
134 1120 LET T=T+1
135 1130 LET T=T+1
136 1140 LET T=T+1
137 1150 LET T=T+1
138 1160 LET T=T+1
139 1170 LET T=T+1
140 1180 LET T=T+1
141 1190 LET T=T+1
142 1200 LET T=T+1
143 1210 LET T=T+1
144 1220 LET T=T+1
145 1230 LET T=T+1
146 1240 LET T=T+1
147 1250 LET T=T+1
148 1260 LET T=T+1
149 1270 LET T=T+1
150 1280 LET T=T+1
151 1290 LET T=T+1
152 1300 LET T=T+1
153 1310 LET T=T+1
154 1320 LET T=T+1
155 1330 LET T=T+1
156 1340 LET T=T+1
157 1350 LET T=T+1
158 1360 LET T=T+1
159 1370 LET T=T+1
160 1380 LET T=T+1
161 1390 LET T=T+1
162 1400 LET T=T+1
163 1410 LET T=T+1
164 1420 LET T=T+1
165 1430 LET T=T+1
166 1440 LET T=T+1
167 1450 LET T=T+1
168 1460 LET T=T+1
169 1470 LET T=T+1
170 1480 LET T=T+1
171 1490 LET T=T+1
172 1500 LET T=T+1
173 1510 LET T=T+1
174 1520 LET T=T+1
175 1530 LET T=T+1
176 1540 LET T=T+1
177 1550 LET T=T+1
178 1560 LET T=T+1
179 1570 LET T=T+1
180 1580 LET T=T+1
181 1590 LET T=T+1
182 1600 LET T=T+1
183 1610 LET T=T+1
184 1620 LET T=T+1
185 1630 LET T=T+1
186 1640 LET T=T+1
187 1650 LET T=T+1
188 1660 LET T=T+1
189 1670 LET T=T+1
190 1680 LET T=T+1
191 1690 LET T=T+1
192 1700 LET T=T+1
193 1710 LET T=T+1
194 1720 LET T=T+1
195 1730 LET T=T+1
196 1740 LET T=T+1
197 1750 LET T=T+1
198 1760 LET T=T+1
199 1770 LET T=T+1
200 1780 LET T=T+1
201 1790 LET T=T+1
202 1800 LET T=T+1
203 1810 LET T=T+1
204 1820 LET T=T+1
205 1830 LET T=T+1
206 1840 LET T=T+1
207 1850 LET T=T+1
208 1860 LET T=T+1
209 1870 LET T=T+1
210 1880 LET T=T+1
211 1890 LET T=T+1
212 1900 LET T=T+1
213 1910 LET T=T+1
214 1920 LET T=T+1
215 1930 LET T=T+1
216 1940 LET T=T+1
217 1950 LET T=T+1
218 1960 LET T=T+1
219 1970 LET T=T+1
220 1980 LET T=T+1
221 1990 LET T=T+1
222 2000 LET T=T+1
223 2010 LET T=T+1
224 2020 LET T=T+1
225 2030 LET T=T+1
226 2040 LET T=T+1
227 2050 LET T=T+1
228 2060 LET T=T+1
229 2070 LET T=T+1
230 2080 LET T=T+1
231 2090 LET T=T+1
232 2100 LET T=T+1
233 2110 LET T=T+1
234 2120 LET T=T+1
235 2130 LET T=T+1
236 2140 LET T=T+1
237 2150 LET T=T+1
238 2160 LET T=T+1
239 2170 LET T=T+1
240 2180 LET T=T+1
241 2190 LET T=T+1
242 2200 LET T=T+1
243 2210 LET T=T+1
244 2220 LET T=T+1
245 2230 LET T=T+1
246 2240 LET T=T+1
247 2250 LET T=T+1
248 2260 LET T=T+1
249 2270 LET T=T+1
250 2280 LET T=T+1
251 2290 LET T=T+1
252 2300 LET T=T+1
253 2310 LET T=T+1
254 2320 LET T=T+1
255 2330 LET T=T+1
256 2340 LET T=T+1
257 2350 LET T=T+1
258 2360 LET T=T+1
259 2370 LET T=T+1
260 2380 LET T=T+1
261 2390 LET T=T+1
262 2400 LET T=T+1
263 2410 LET T=T+1
264 2420 LET T=T+1
265 2430 LET T=T+1
266 2440 LET T=T+1
267 2450 LET T=T+1
268 2460 LET T=T+1
269 2470 LET T=T+1
270 2480 LET T=T+1
271 2490 LET T=T+1
272 2500 LET T=T+1
273 2510 LET T=T+1
274 2520 LET T=T+1
275 2530 LET T=T+1
276 2540 LET T=T+1
277 2550 LET T=T+1
278 2560 LET T=T+1
279 2570 LET T=T+1
280 2580 LET T=T+1
281 2590 LET T=T+1
282 2600 LET T=T+1
283 2610 LET T=T+1
284 2620 LET T=T+1
285 2630 LET T=T+1
286 2640 LET T=T+1
287 2650 LET T=T+1
288 2660 LET T=T+1
289 2670 LET T=T+1
290 2680 LET T=T+1
291 2690 LET T=T+1
292 2700 LET T=T+1
293 2710 LET T=T+1
294 2720 LET T=T+1
295 2730 LET T=T+1
296 2740 LET T=T+1
297 2750 LET T=T+1
298 2760 LET T=T+1
299 2770 LET T=T+1
300 2780 LET T=T+1
301 2790 LET T=T+1
302 2800 LET T=T+1
303 2810 LET T=T+1
304 2820 LET T=T+1
305 2830 LET T=T+1
306 2840 LET T=T+1
307 2850 LET T=T+1
308 2860 LET T=T+1
309 2870 LET T=T+1
310 2880 LET T=T+1
311 2890 LET T=T+1
312 290
```

[illegible][illegible]

NOTAS		VALOR DE LOS SEMITONOS												
C	DO	-60	-48	-36	-24	-12	0	12	24	36	48	60		
C#	DO#	-59	-47	-35	-23	-11	1	13	25	37	49	61		
D	RE	-58	-46	-34	-22	-10	2	14	26	38	50	62		
D#	RE#	-57	-45	-33	-21	-9	3	15	27	39	51	63		
E	MI	-56	-44	-32	-20	-8	4	16	28	40	52	64		
F	FA	-55	-43	-31	-19	-7	5	17	29	41	53	65		
F#	FA#	-54	-42	-30	-18	-6	6	18	30	42	54	66		
G	SOL	-53	-41	-29	-17	-5	7	19	31	43	55	67		
G#	SOL#	-52	-40	-28	-16	-4	8	20	32	44	56	68		
A	LA	-51	-39	-27	-15	-3	9	21	33	45	57	69		
A#	LA#	-50	-38	-26	-14	-2	10	22	34	46	58			
B	SI	-49	-37	-25	-13	-1	11	23	35	47	59			



OCTAVA CENTRAL

II SOSTENIDO

Correspondencia entre «notas» y «semitonos» de las diferentes octavas.

Nociones musicales

Nociones musicales

Tono
(tono agudo) y más bajo

En este párrafo se van a explicar una serie de conceptos útiles para aquellas personas que deseen proporcionar una melodía en el

tono
El tono es el grado de elevación del sonido, conocido también como <i>altura tonal</i> . Siendo el tono más alto <i>total</i> .

En una escala *diatónica*, las notas musicales identificadas del tono son:

- VERIFY "T" DATA M\$ ()
Los campos "nombre" y "ti-
po" siguen las mismas reglas
que en el caso de "SAVE".
a) Verificación de un pro-
grama BASIC.

VERIFY "nombre"

b) Verificación de un pro-
grama en C/M o de bytes.
Pueden utilizarse cualquiera
de estas opciones:

VERIFY "nombre" CODE comienzo,
longitud
VERIFY "nombre" CODE comienzo
VERIFY "nombre" CODE

Existen pequeñas diferen-
cias en cuanto a la forma de
verificar de cada una de ellas,
siendo la más exacta la pri-
mera.

c) Verificación de una ma-
triz numérica.

VERIFY "nombre" DATA letra ()

d) Verificación de una ma-
triz de cadena.

VERIFY "nombre" DATA letra\$ ()

Los programas almacena-
dos como:

SAVE "nombre" SCREEN\$
0
SAVE "nombre" CODE 16384, 6912

no pueden verificarse ya que
al detectar el ordenador la ca-
becera de el programa, apa-
rece en pantalla:

Bytes: nombre

modificando por tanto la me-
moria de pantalla y dando lu-
gar al correspondiente error.

Aceso al teclado

VAL



VAL \$ Definición

La sentencia "LOAD" per-
mite cargar en la memoria del
ordenador un programa gra-
bado en cinta de cassette.

"LOAD" borra el programa
existente en ese momento en
el ordenador y sus variables.
La estructura general es:

SENTENCIA	ARGUMENTO
LOAD	"nombre" tipo

Ejemplo:

- LOAD "OH"
- LOAD "T" CODE
- LOAD "PP" DATA Z ()
- LOAD "NT" DATA P\$ ()

Los campos "nombre" y "ti-
po" siguen las mismas reglas
que en el caso de "SAVE".

a) Carga de programas
BASIC.

LOAD "nombre"

Si el programa fue almace-
nado con el tipo "LINE" auto-
máticamente se autoejecu-
tará.

b) Carga de programas en
C/M o de bytes.

LOAD "nombre" CODE comienzo,
longitud
LOAD "nombre" CODE comienzo
LOAD "nombre" CODE

como casos específicos pue-
de utilizarse para la carga de
pantallas:

LOAD "nombre" SCREEN\$

La forma de cargar en pan-
talla una información es un
tanto peculiar ya que su me-
moria está distribuida en tres
zonas. Primero, se cargan los
bytes cero de cada carácter
(matriz de 8 por 8 pixel), de la
primera zona; a continuación,
los bytes uno y así sucesiva-
mente hasta completar la pri-
mera zona. Las zonas dos y
tres se cargan de la misma
manera y al final se incluyen
los atributos.

El siguiente programa ayu-
dará a comprender la se-
cuencia de carga de la me-
moria de pantalla.
y para los GDU

```
10 REM *****
20 LET direccion=6304
40 FOR direccion TO direccion+255
50 NEXT direccion
```

LOAD "nombre" CODE USR "a"

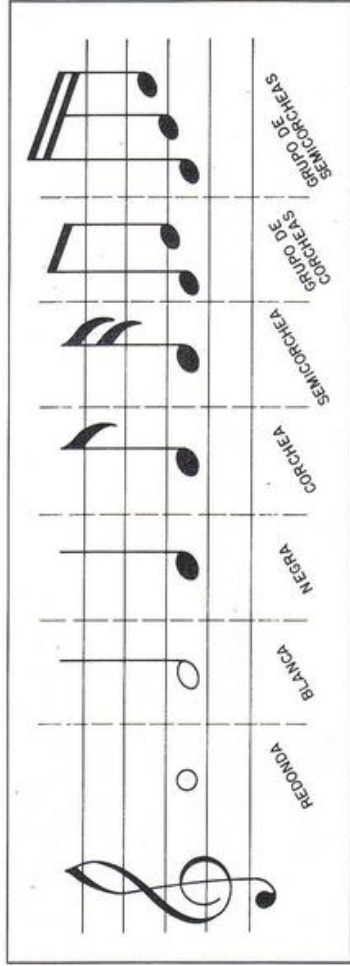
c) Carga de matrices nu-
méricas.

LOAD "nombre" DATA letra ()

d) Carga de matrices de
cadena

LOAD "nombre" DATA letra \$ ()

Al cargar una matriz, bien
sea de cadena o numérica, no
se borra el programa BASIC
existente, pero sí el contenido
de cualquier matriz que ante-
riormente estuviera definida
con el mismo nombre.



Simbología utilizada en la duración relativa de las notas.

INDICATIVO
DEL COMPAS

COMPAS



DOS POR CUATRO



TRES POR CUATRO



CUATRO POR CUATRO

Ejemplos típicos de compases.

cassette en la salida «MIC» del
Spectrum podrá grabar cual-
quier melodía que genere el
ordenador, ya que la señal
que activa el altavoz también

se encuentra presente en di-
cho conector.

También es posible conec-
tar un amplificador a dicha
salida, con lo que aumentará

el volumen de la melodía. Si
conecta unos auriculares po-
drá escuchar simultánea-
mente los sonidos que el or-
denador genere, aunque hay

que tener en cuenta que el altavoz interno no se desconecta.

Periféricos

Hay ciertos periféricos o dispositivos que podemos conectar al ordenador y que aumentan las posibilidades sonoras de éste.

- Amplificador de sonido que permite escuchar el sonido generado por el ordenador, a través del altavoz de nuestro receptor de T.V.; de esta manera se dispone de un control de volumen a gusto del usuario.

- Sintetizador de voz que permite al ordenador «hablar» de la misma manera que lo haría un robot. Los sonidos a

emitir deben ser asignados a cierta variable de cadena.

Software musical

Aunque no abundantes, existen en el mercado algunos programas musicales: unos son del tipo didáctico y permiten seguir un curso de «teoría de la música» a través del ordenador. Otro, conocido por el nombre comercial de «Melodian», permite asistir a todas las fases de una composición musical, permitiendo:

- Escribir partituras.
- Ejecutarlas.
- Imprimirlas.
- Almacenarlas en cinta.

El programa viene acompañado de una plantilla especial de cartulina que convierte al Spectrum en un teclado musical simulado.

Efectos sonoros

Los programas «5» a «8» proporcionan diversos efectos sonoros que pueden ser utilizados en nuestros programas.

El programa número «9» proporciona un efecto de «TREMOLADO»; debe ser utilizado con la tabla de datos del programa número «1», (Navidad):

```

4240 NEXT I
4250 RETURN
4300 REM SUB PROGRAMA
4310 PRINT AT 21,1,TECLLEE EL NU
4320 DE PROGRAMA.; FLASH 1.););
4330 M. INT
4340 PAUSE 0
4350 LET a$=INKEY$
4360 IF a$="" GO TO 4320
4370 PRINT AT 21,30,a$; BEEP 0.
4380
4390 LET programa=VAL (a$)
4400 PAUSE 0
4410 LET a$=INKEY$
4420 IF a$=CHR$ 13 THEN GO TO 44
4430
4440 IF a$="" OR a$="9" THEN BE
4450
4460 IF a$="0" THEN BE
4470
4480 IF a$="1" THEN BE
4490
4500 IF a$="2" THEN BE
4510
4520 IF a$="3" THEN BE
4530
4540 IF a$="4" THEN BE
4550
4560 IF a$="5" THEN BE
4570
4580 IF a$="6" THEN BE
4590
4600 IF a$="7" THEN BE
4610
4620 IF a$="8" THEN BE
4630
4640 IF a$="9" THEN BE
4650
4660 IF a$="A" THEN BE
4670
4680 IF a$="B" THEN BE
4690
4700 IF a$="C" THEN BE
4710
4720 IF a$="D" THEN BE
4730
4740 IF a$="E" THEN BE
4750
4760 IF a$="F" THEN BE
4770
4780 IF a$="G" THEN BE
4790
4800 IF a$="H" THEN BE
4810
4820 IF a$="I" THEN BE
4830
4840 IF a$="J" THEN BE
4850
4860 IF a$="K" THEN BE
4870
4880 IF a$="L" THEN BE
4890
4900 IF a$="M" THEN BE
4910
4920 IF a$="N" THEN BE
4930
4940 IF a$="O" THEN BE
4950
4960 IF a$="P" THEN BE
4970
4980 IF a$="Q" THEN BE
4990
5000 IF a$="R" THEN BE
5010
5020 IF a$="S" THEN BE
5030
5040 IF a$="T" THEN BE
5050
5060 IF a$="U" THEN BE
5070
5080 IF a$="V" THEN BE
5090
5100 IF a$="W" THEN BE
5110
5120 IF a$="X" THEN BE
5130
5140 IF a$="Y" THEN BE
5150
5160 IF a$="Z" THEN BE
5170
5180 IF a$=" " THEN BE
5190
5200 IF a$="." THEN BE
5210
5220 IF a$="," THEN BE
5230
5240 IF a$=";" THEN BE
5250
5260 IF a$=":" THEN BE
5270
5280 IF a$="@" THEN BE
5290
5300 IF a$="!" THEN BE
5310
5320 IF a$="#" THEN BE
5330
5340 IF a$="$" THEN BE
5350
5360 IF a$="%" THEN BE
5370
5380 IF a$="&" THEN BE
5390
5400 IF a$="'" THEN BE
5410
5420 IF a$="(" THEN BE
5430
5440 IF a$=")" THEN BE
5450
5460 IF a$="{" THEN BE
5470
5480 IF a$="}" THEN BE
5490
5500 IF a$="[" THEN BE
5510
5520 IF a$="]" THEN BE
5530
5540 IF a$="`" THEN BE
5550
5560 IF a$="~" THEN BE
5570
5580 IF a$="_" THEN BE
5590
5600 IF a$="+" THEN BE
5610
5620 IF a$="-" THEN BE
5630
5640 IF a$="=" THEN BE
5650
5660 IF a$="*" THEN BE
5670
5680 IF a$="/" THEN BE
5690
5700 IF a$="\" THEN BE
5710
5720 IF a$="|" THEN BE
5730
5740 IF a$="|" THEN BE
5750
5760 IF a$="|" THEN BE
5770
5780 IF a$="|" THEN BE
5790
5800 IF a$="|" THEN BE
5810
5820 IF a$="|" THEN BE
5830
5840 IF a$="|" THEN BE
5850
5860 IF a$="|" THEN BE
5870
5880 IF a$="|" THEN BE
5890
5900 IF a$="|" THEN BE
5910
5920 IF a$="|" THEN BE
5930
5940 IF a$="|" THEN BE
5950
5960 IF a$="|" THEN BE
5970
5980 IF a$="|" THEN BE
5990
6000 IF a$="|" THEN BE
6010
6020 IF a$="|" THEN BE
6030
6040 IF a$="|" THEN BE
6050
6060 IF a$="|" THEN BE
6070
6080 IF a$="|" THEN BE
6090
6100 IF a$="|" THEN BE
6110
6120 IF a$="|" THEN BE
6130
6140 IF a$="|" THEN BE
6150
6160 IF a$="|" THEN BE
6170
6180 IF a$="|" THEN BE
6190
6200 IF a$="|" THEN BE
6210
6220 IF a$="|" THEN BE
6230
6240 IF a$="|" THEN BE
6250
6260 IF a$="|" THEN BE
6270
6280 IF a$="|" THEN BE
6290
6300 IF a$="|" THEN BE
6310
6320 IF a$="|" THEN BE
6330
6340 IF a$="|" THEN BE
6350
6360 IF a$="|" THEN BE
6370
6380 IF a$="|" THEN BE
6390
6400 IF a$="|" THEN BE
6410
6420 IF a$="|" THEN BE
6430
6440 IF a$="|" THEN BE
6450
6460 IF a$="|" THEN BE
6470
6480 IF a$="|" THEN BE
6490
6500 IF a$="|" THEN BE
6510
6520 IF a$="|" THEN BE
6530
6540 IF a$="|" THEN BE
6550
6560 IF a$="|" THEN BE
6570
6580 IF a$="|" THEN BE
6590
6600 IF a$="|" THEN BE
6610
6620 IF a$="|" THEN BE
6630
6640 IF a$="|" THEN BE
6650
6660 IF a$="|" THEN BE
6670
6680 IF a$="|" THEN BE
6690
6700 IF a$="|" THEN BE
6710
6720 IF a$="|" THEN BE
6730
6740 IF a$="|" THEN BE
6750
6760 IF a$="|" THEN BE
6770
6780 IF a$="|" THEN BE
6790
6800 IF a$="|" THEN BE
6810
6820 IF a$="|" THEN BE
6830
6840 IF a$="|" THEN BE
6850
6860 IF a$="|" THEN BE
6870
6880 IF a$="|" THEN BE
6890
6900 IF a$="|" THEN BE
6910
6920 IF a$="|" THEN BE
6930
6940 IF a$="|" THEN BE
6950
6960 IF a$="|" THEN BE
6970
6980 IF a$="|" THEN BE
6990
7000 IF a$="|" THEN BE
7010
7020 IF a$="|" THEN BE
7030
7040 IF a$="|" THEN BE
7050
7060 IF a$="|" THEN BE
7070
7080 IF a$="|" THEN BE
7090
7100 IF a$="|" THEN BE
7110
7120 IF a$="|" THEN BE
7130
7140 IF a$="|" THEN BE
7150
7160 IF a$="|" THEN BE
7170
7180 IF a$="|" THEN BE
7190
7200 IF a$="|" THEN BE
7210
7220 IF a$="|" THEN BE
7230
7240 IF a$="|" THEN BE
7250
7260 IF a$="|" THEN BE
7270
7280 IF a$="|" THEN BE
7290
7300 IF a$="|" THEN BE
7310
7320 IF a$="|" THEN BE
7330
7340 IF a$="|" THEN BE
7350
7360 IF a$="|" THEN BE
7370
7380 IF a$="|" THEN BE
7390
7400 IF a$="|" THEN BE
7410
7420 IF a$="|" THEN BE
7430
7440 IF a$="|" THEN BE
7450
7460 IF a$="|" THEN BE
7470
7480 IF a$="|" THEN BE
7490
7500 IF a$="|" THEN BE
7510
7520 IF a$="|" THEN BE
7530
7540 IF a$="|" THEN BE
7550
7560 IF a$="|" THEN BE
7570
7580 IF a$="|" THEN BE
7590
7600 IF a$="|" THEN BE
7610
7620 IF a$="|" THEN BE
7630
7640 IF a$="|" THEN BE
7650
7660 IF a$="|" THEN BE
7670
7680 IF a$="|" THEN BE
7690
7700 IF a$="|" THEN BE
7710
7720 IF a$="|" THEN BE
7730
7740 IF a$="|" THEN BE
7750
7760 IF a$="|" THEN BE
7770
7780 IF a$="|" THEN BE
7790
7800 IF a$="|" THEN BE
7810
7820 IF a$="|" THEN BE
7830
7840 IF a$="|" THEN BE
7850
7860 IF a$="|" THEN BE
7870
7880 IF a$="|" THEN BE
7890
7900 IF a$="|" THEN BE
7910
7920 IF a$="|" THEN BE
7930
7940 IF a$="|" THEN BE
7950
7960 IF a$="|" THEN BE
7970
7980 IF a$="|" THEN BE
7990
8000 IF a$="|" THEN BE
8010
8020 IF a$="|" THEN BE
8030
8040 IF a$="|" THEN BE
8050
8060 IF a$="|" THEN BE
8070
8080 IF a$="|" THEN BE
8090
8100 IF a$="|" THEN BE
8110
8120 IF a$="|" THEN BE
8130
8140 IF a$="|" THEN BE
8150
8160 IF a$="|" THEN BE
8170
8180 IF a$="|" THEN BE
8190
8200 IF a$="|" THEN BE
8210
8220 IF a$="|" THEN BE
8230
8240 IF a$="|" THEN BE
8250
8260 IF a$="|" THEN BE
8270
8280 IF a$="|" THEN BE
8290
8300 IF a$="|" THEN BE
8310
8320 IF a$="|" THEN BE
8330
8340 IF a$="|" THEN BE
8350
8360 IF a$="|" THEN BE
8370
8380 IF a$="|" THEN BE
8390
8400 IF a$="|" THEN BE
8410
8420 IF a$="|" THEN BE
8430
8440 IF a$="|" THEN BE
8450
8460 IF a$="|" THEN BE
8470
8480 IF a$="|" THEN BE
8490
8500 IF a$="|" THEN BE
8510
8520 IF a$="|" THEN BE
8530
8540 IF a$="|" THEN BE
8550
8560 IF a$="|" THEN BE
8570
8580 IF a$="|" THEN BE
8590
8600 IF a$="|" THEN BE
8610
8620 IF a$="|" THEN BE
8630
8640 IF a$="|" THEN BE
8650
8660 IF a$="|" THEN BE
8670
8680 IF a$="|" THEN BE
8690
8700 IF a$="|" THEN BE
8710
8720 IF a$="|" THEN BE
8730
8740 IF a$="|" THEN BE
8750
8760 IF a$="|" THEN BE
8770
8780 IF a$="|" THEN BE
8790
8800 IF a$="|" THEN BE
8810
8820 IF a$="|" THEN BE
8830
8840 IF a$="|" THEN BE
8850
8860 IF a$="|" THEN BE
8870
8880 IF a$="|" THEN BE
8890
8900 IF a$="|" THEN BE
8910
8920 IF a$="|" THEN BE
8930
8940 IF a$="|" THEN BE
8950
8960 IF a$="|" THEN BE
8970
8980 IF a$="|"
```

Acceso al teclado

Definición

Su estructura general es:

SENTENCIA	ARGUMENTO
VERIFY	"nombre" tipo

Ejemplos:

- VERIFY "CAO CAO"
- VERIFY "HOP" CODE
- VERIFY "S" DATA K ()

La sentencia "VERIFY" comprueba que el programa o datos se han grabado correctamente, para ello compara la información de la cinta con la resistencia en memoria.



Organización de la tabla de datos utilizada en el programa DIRECTORIO.

PROGRAMA 2

```

10 REM *****
** CURSOR/BASIC **
** *****
** DIRECTORIO **
** *****
80 GO SUB 4200
90 BORDER 1: PAPER 1: INK 7: C
100 REM BORROR
110 GO SUB 5000
120 PRINT INVERSE 1: AT 3,12: "CA
RA": PRINT "": PRINT D$(21,1
1 TO 13)
120 PRINT INVERSE 1: AT 8,1: "NOM
BRE": PRINT "": PRINT D$(21,
1 TO 13)
130 PRINT INVERSE 1: AT 11,1: "CO
MENTARIOS": PRINT
140 PRINT "": D$(21,34 TO 53)
150 PRINT "": D$(21,64 TO 93)
160 PRINT "": D$(21,94 TO 123)
170 PRINT INVERSE 1: AT 18,1: "PR
OPRIETARIO": PRINT "": PRINT
D$(21,14 TO 33)
180 PRINT INVERSE 1: AT 21,1: "PUL
SE ENTER": PRINT
190 PRINT INVERSE 1: AT 21,1: "PUL
SE 0"
200 LET AS$="INKEY$
210 IF CODE AS$=13 THEN BEEP 0.
215 GO TO 190
220 REM OPCIONES
230 GO SUB 5000
240 PRINT AT 3,8: "MENU DE OPCIO
NES"
250 PRINT AT 4,8: "
260 PRINT INVERSE 1: AT 8,8: "1",
PRESENTACION": PRINT
270 PRINT INVERSE 1: AT 11,8: "2",
PRINT": PRINT
280 PRINT INVERSE 1: AT 14,8: "3",
DIRECTORIO": PRINT
290 PRINT INVERSE 1: AT 17,8: "4",
EJECUCION": PRINT
300 PRINT INVERSE 1: AT 20,8: "5",
INFORMACION": PRINT
310 PRINT AT 21,1: "PULSE LA OPC
ION DESIADA"
320 REM SELECCION
330 PRINT "
340 PRINT 0
350 PRINT 0
360 PRINT AT 21,0: "
370 LET AS$="INKEY$
380 IF AS$=1, THEN PRINT FLASH
390 IF AS$=2, THEN PRINT "PRESENTACION": PRINT
400 GO SUB 5000: GO TO 110
410 IF AS$=3, THEN PRINT FLASH
420 IF AS$=4, THEN PRINT "DIRECTORIO": PRINT
430 GO TO 1000: THEN LET SELECCION
440 AS$="0"
450 PRINT AT 21,0: "PULSE LA OPCION
DESEADA": PRINT
460 PRINT 0
470 PRINT 0
480 PRINT AT 21,0: "
490 LET AS$="INKEY$
500 IF AS$=1, THEN PRINT FLASH
510 PRINT 0
520 PRINT AT 21,0: "PULSE ENTER"
530 PRINT 0
540 IF CODE AS$=13 THEN BEEP 0.
550 GO TO 215
560 REM OPCIONES 3 y 4
570 PRINT AT 21,0: "
580 PRINT 0
590 PRINT 0
600 PRINT AT 21,0: "PULSE ENTER"
610 PRINT 0
620 IF CODE AS$=13 THEN BEEP 0.
630 GO TO 215
640 REM OPCIONES 5 y 6
650 PRINT AT 21,0: "PULSE ENTER"
660 PRINT 0
670 IF AS$="INKEY$
680 LET AS$="CHR$ 12 THEN BEEP 0.0
690 GO TO FLASH=0: GO SUB 4100:
700 IF AS$="CHR$ 13 THEN GO TO 21
710 BEEP 0.2: 15: GO TO 2065

```

```

2110 CLS : GO SUB 5000
2120 IF selection=3 THEN GO TO 2
2130 GO TO 3000
2140 REM OPCION/LOGIC
22510 PRINT AT 3,7,"CARGA DE PROG
22520 PRINT AT 4,7,"
2300 PRINT INVERSE,1,AT 6,1,"PRO
2310 PRINT ":",D$(programa,
1 TO 10)
2340 PRINT INVERSE 1,AT 12,1,"CO
2350 PRINT ":",D$(programa
11 TO 13)
2360 PRINT "AT 16,1,"PULSE LA TEC
LA "PLAY," DEL CA,"
2370 PRINT AT 18,1,"SSETTE Y A C
ON INICIA PULSA 1,"
2380 PRINT AT 20,1,"LA TECLA ""E
NTER""
2390 PAUSE 0
2500 LET $=INKEY$
2510 IF $=CHR$(13) THEN BEEP 0,20
2520 IF $=CHR$(13) THEN BEEP 0,20
2530 GO TO 2580
2540 BEEP 0.05,20
2550 STOP
2560 D$(programa,1 TO 10)
2600 REM GEOTON/INFOR
30010 PRINT AT 2,10,"INFORMACION"
30020 PRINT AT 3,10,"
30030 PRINT INVERSE 1,AT 7,1,"PRO
GRAMA 10,"
30040 PRINT ":",D$(programa,
1 TO 10)
30050 PRINT INVERSE 1,AT 10,1,"CO
MENTARIOS"
30060 PRINT AT 12,1,D$(programa,3
4 TO 63)
30070 PRINT AT 13,1,D$(programa,6
4 TO 93)
30080 PRINT AT 14,1,D$(programa,9
4 TO 120)
30090 PRINT INVERSE 1,AT 17,1,"PR
OGRAMA RETURN,"
30100 PRINT ":",D$(prog
rama,11 TO 25)
30110 PRINT INVERSE 1,AT 20,1,"FE
CHA "
30120 PRINT ":",D$(programa,26
4 TO 33)
30130 PRINT INVERSE 1,AT 20,25,"C
OND 13"
30140 PRINT ":",D$(programa,11
TO 13)
3100 RETURN," PULSE ""ENTER""
3110 PAUSE 0
3120 LET $=INKEY$
3130 IF $=CHR$(13) THEN BEEP 0.
2
3140 GO TO 3110
3150 GO TO 3110
3160 STOP
3200 REM SUB/DIE
4000
4005 CLS : BEEP 0.05,20 : LET J=-
1
4010 FOR I=1 TO 10
4020 IF I=10 THEN PRINT INVERSE
1,AT 1,10,"
4030 IF I=1,2,11 PRINT ":",D$(1,1
TO 10)
4040 PRINT INVERSE 1,AT 1,10,"
7,1,10," PRINT ":",D$(1,10,1 TO
10)
4050 IF I=10 THEN PRINT INVERSE
1,AT 1,10,"
4060 IF I=1,11 PRINT ":",D$(1,1
TO 10)
4070 PRINT ":",D$(1,10,1 TO
10)
4085 LET J=J+1
4090 NEXT I
4100 REM SUB/ELEPH
4110
4120 IF (programa#11) THEN LET x=4
4130 LET y=(programa#2)-2: GO TO 41
40
4140 LET x=20: LET y=(programa#1
0) #2-2
4150 PRINT FLASH (flash,AT y,x,D$
(programa,1
TO 10)
4160 RETURN
4170 REM RETORNE ULTIMO
4180
4190 FOR I=1 TO 20
4200 IF D$(1,1 TO 10)="
"
4210 THEN LET ultimo=I-1: GO TO 42
50
4220 LET ultimo=I
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490
4500

```

SENTENCIAS DE GRABACION Y CARGA

El capítulo número tres, "Almacenamiento de programas", fue una breve introducción a la grabación y recuperación de programas BASIC; este va a ser más completo, ya que analizará todos los posibles argumentos de las sentencias:

OF" quedará almacenado en cinta.

Una forma particular de grabar un programa BASIC, es indicar el número de línea donde se desea que comience la *autoejecución*, así, cuando se realice una carga posterior, no será necesario pulsar "RUN" n, donde "n" es el número de línea.

Ejemplo:

"Tragon"
 "EMULADOR"
 "1234.5"
 "\$.71"

SAVE "ON/OFF" LINE 1000

El programa "ON/OFF" se autoejecutará a partir de la línea 1000. Si se desea que un programa se autoejecute a partir de la primera sentencia, puede grabarse con LINE 1.

Grabación de código máquina

Para grabar C/M (Código Máquina), es decir, los programas escritos en el lenguaje interno del ordenador, se utiliza el tipo "CODE", donde es necesario indicar la dirección de memoria a partir de la cual se desea grabar y la longitud del programa expresada en bytes.

SAVE "S.O." CODE 0 16384

de esta forma se almacena en cinta una copia del Sistema operativo del Spectrum, que comienza en la dirección "0"

270 MICROBASIC

MICROBASIC 267

Acceso al teclado

RESTORE



MODOK

Definición

La sentencia "SAVE" permite almacenar en cinta cualquier programa escrito en lenguaje BASIC o en "Código Máquina", así como las matrices.

Su estructura general es:

SENTENCIA	ARGUMENTO
SAVE	"nombre" tipo

pantalla el gráfico o texto que
 desee a base de "PLOT",
 "DRAW", "CIRCLE" o "PRINT".
 Cuando esté confeccionado,
 se podrá grabar en cinta de la
 forma:

SAVE "nombre" SCREEN\$

La palabra clave "SCREEN\$" es equivalente en este caso a:

CODE 16384 6912

ya que la dirección de comienzo de la zona de pantalla es 16384 y tiene una longitud de 6912 bytes. Puede utilizar cualquiera de las dos fórmulas posibles.

Para almacenar la zona de memoria correspondiente a los GDU, puede utilizar, como

1 Kbyte = 1024 bytes

1 KByte = 1024 bytes

Un caso particular, es la grabación de los datos de la zona de memoria correspon-

PROGRAMA 1

```

10 REM *****
   CURSOR/BASIC *****
   *****
   LISTADOR *****
   *****

20 REM COMIGO Gomez 0 1981
30 CLEAR 29999
40 BORDER 4: PAPER 4: INK 0:
   LS
50 LET contador=1
60 RESTORE 60 SUB 500
70 REM BARRA-CHORR-CHORR
   "/
80 INPUT "LINE # Nombre de la cinta >"
90 IF LEN J<15 THEN LET J$=1
   ( TO 15)
100 FOR n=1 TO (15-LEN J$)
110 LET J$=J$+CHR$ 32
120 NEXT n
130 INPUT "Cará >" THEN LET J$=J$+
   1)
140 IF J$="" THEN LET J$=CHR$ 3
   2
150 REM BARRA-CHORR-CHORR
160 LET J$=CHR$ 138:"CINTA:"+CHR$
   R$ 32+J$+CHR$ 32:"CARA:"+CHR$ 32
170 LET J$=""
180 LET J$=J$+d$
190 LET J$=CHR$ 138:"N TIPO No
   mbe Comigo Long."+CHR$ 133
200 LET J$=J$+J$+J$+J$
210 FOR n=2 TO 3
220 PRINT n;d$
230 NEXT n
240 REM BARRA-CHORR-CHORR
250 LET J$="STR# contador
   260 LET J$=J$+LEN J$+2 THEN LET J$=CHR$
   3)

```

[illegible]

268 MICROBASIC

Diversas pantallas del programa «Directorio».

MICROBASIC 269

Elección de una impresora

A la hora de elegir una impresora deben tenerse en cuenta ciertos factores que intervendrán directamente en su precio.

- La velocidad de impresión en cps.
- Si imprime en dos sentidos (bidireccional).
- Número de caracteres por línea.
- Tipo de papel:
 - Rollo
 - Perforado
 - Térmico
 - Hojas
- Tipo de tinta.
- Juegos de caracteres que incorpora.

Fricción y de Arrastre combinado que permite introducir o bien hojas sueltas o bien papel perforado.

- Si imprime en varios colores.
- La elección deberá ir en concordancia con el tipo de utilidad a la que se vaya a destinar y con el volumen de trabajo que deberá soportar.

Caracteres de control

La mayoría de las impresoras utilizan algunos caracteres ASCII para cambiar el tamaño de las letras, la distancia entre líneas, el tipo de letra, si deben ir subrayadas o en negrita, etc.

Estos códigos dependen del tipo de impresora, por tanto deberá leer las instrucciones específicas de la que esté manejando; pero en general la forma de introducir estos comandos es:

LPRINT CHR\$ n

donde "n" es el código de control a enviar.

FIRMWARE ID: 21066 REV A
FIRMWARE ID: 21065 REV A
GENERATOR INTERNAL
1K RAM

FORM LENGTH 12 INCH
BOTTOM OF FORM SKIP YES
LINES PER INCH 6
CHARACTER PER INCH 10
CR = CR LF YES
LF AT LINE END NO
CHARACTER SET US ASCII
SLASHED ZERO YES
TEST MODUS ROLLING ASCII
END OF MENU.

CHANGE MENU YES
CHANGE FORM LENGTH YES
4 INCH NO
5 INCH NO
6 INCH NO
8 INCH NO
8.5 INCH NO
11 INCH NO
12 INCH NO
14 INCH NO
NO CHANGE YES
BOTTOM OF FORMAT SKIP YES
CHANGE PRINT FORMAT YES
CHANGE LPI YES
6 NO
8 NO
NO CHANGE YES
10 NO
12.5 NO
16 2/3 NO
20 NO
NO CHANGE YES
CR = CR LF YES
LF AT LINE END NO
CHANGE CHARACTER SET YES
US ASCII NO
UK ASCII NO
FRENCH/BELGIAN NO
GERMAN NO
ITALIAN NO
SWEDISH/FINNISH NO
DANISH/NORWEGIAN NO
SPANISH YES
SLASHED ZERO YES
CHANGE TEST MODUS YES
ROLLING ASCII YES
GENERATOR TEST YES
END OF MENU.

Menú de opciones de impresión programables por el usuario.

PROGRAMA 3

```
10 REM
11 *****
12 * CURSO/BASIC *
13 *****
14 * EDIT/DIR *
15 *****
16 BORDER 4. PAPER 4. INK 0. C
17 DIM D$(11,13)
18 DIM D$(11,13)
19 REM CARGA CARATULA
20 INPUT "NOMBRE DE LA CINTA: "; LINE AS$
21 IF LEN AS$ > 10 THEN LET AS$ = AS$(1 TO 10)
22 PRINT "NOMBRE"; AS$
23 LET D$(1,1) = AS$
24 INPUT "CARA. "; LINE AS$
25 IF LEN AS$ > 1 THEN LET AS$ = AS$(1)
26 PRINT "CARA"; AS$
27 LET D$(2,1) = AS$
28 INPUT "PROPIETARIO "; LINE AS$
29 IF LEN AS$ > 12 THEN LET AS$ = AS$(1 TO 12)
30 PRINT "PROPIETARIO"; AS$
31 LET D$(3,1) = AS$
32 INPUT "COMENTARIOS "; LINE AS$
33 IF LEN AS$ > 90 THEN LET AS$ = AS$(1 TO 90)
34 PRINT "COMENTARIOS"; AS$
35 LET D$(4,1) = AS$
36 LET D$(5,1) = AS$
37 IF AS$ = "S" THEN GO TO 300
38 IF AS$ = "N" THEN GO TO 1000
39 CLS
40 PRINT "X";
41 PRINT "X";
42 PRINT "X";
43 PRINT "X";
44 PRINT "X";
45 PRINT "X";
46 PRINT "X";
47 PRINT "X";
48 PRINT "X";
49 PRINT "X";
50 PRINT "X";
51 PRINT "X";
52 PRINT "X";
53 PRINT "X";
54 PRINT "X";
55 PRINT "X";
56 PRINT "X";
57 PRINT "X";
58 PRINT "X";
59 PRINT "X";
60 PRINT "X";
61 PRINT "X";
62 PRINT "X";
63 PRINT "X";
64 PRINT "X";
65 PRINT "X";
66 PRINT "X";
67 PRINT "X";
68 PRINT "X";
69 PRINT "X";
70 PRINT "X";
71 PRINT "X";
72 PRINT "X";
73 PRINT "X";
74 PRINT "X";
75 PRINT "X";
76 PRINT "X";
77 PRINT "X";
78 PRINT "X";
79 PRINT "X";
80 CLS
81 REM CARGA PROGRAMAS
82 FOR X=1 TO 20
83 INPUT "PROGRAMA "; LINE AS$
84 IF LEN AS$ > 10 THEN LET AS$ = AS$(1 TO 10)
85 PRINT "PROGRAMA N. "; AS$
86 IF X > 9 THEN GO TO 126
87 PRINT "X";
88 PRINT "X";
89 PRINT "X";
90 LET D$(X,1) = AS$
91 INPUT "CONTADOR "; LINE AS$
92 IF LEN AS$ > 3 THEN LET AS$ = AS$(1 TO 3)
93 CLS
94 PRINT "CONTADOR"; AS$
95 LET D$(X,2) = AS$
96 INPUT "PROGRAMADOR "; LINE AS$
97 IF LEN AS$ > 12 THEN LET AS$ = AS$(1 TO 12)
98 PRINT "PROGRAMADOR"; AS$
99 LET D$(X,3) = AS$
100 PRINT "PROGRAMA"; AS$
101 LET D$(X,4) = AS$
102 LET D$(X,5) = AS$
103 LET D$(X,6) = AS$
104 LET D$(X,7) = AS$
105 LET D$(X,8) = AS$
106 LET D$(X,9) = AS$
107 LET D$(X,10) = AS$
108 LET D$(X,11) = AS$
109 LET D$(X,12) = AS$
110 LET D$(X,13) = AS$
111 LET D$(X,14) = AS$
112 LET D$(X,15) = AS$
113 LET D$(X,16) = AS$
114 LET D$(X,17) = AS$
115 LET D$(X,18) = AS$
116 LET D$(X,19) = AS$
117 LET D$(X,20) = AS$
118 LET D$(X,21) = AS$
119 LET D$(X,22) = AS$
120 LET D$(X,23) = AS$
121 LET D$(X,24) = AS$
122 LET D$(X,25) = AS$
123 LET D$(X,26) = AS$
124 LET D$(X,27) = AS$
125 LET D$(X,28) = AS$
126 LET D$(X,29) = AS$
127 LET D$(X,30) = AS$
128 LET D$(X,31) = AS$
129 LET D$(X,32) = AS$
130 LET D$(X,33) = AS$
131 LET D$(X,34) = AS$
132 LET D$(X,35) = AS$
133 LET D$(X,36) = AS$
134 LET D$(X,37) = AS$
135 LET D$(X,38) = AS$
136 LET D$(X,39) = AS$
137 LET D$(X,40) = AS$
138 LET D$(X,41) = AS$
139 LET D$(X,42) = AS$
140 LET D$(X,43) = AS$
141 LET D$(X,44) = AS$
142 LET D$(X,45) = AS$
143 LET D$(X,46) = AS$
144 LET D$(X,47) = AS$
145 LET D$(X,48) = AS$
146 LET D$(X,49) = AS$
147 LET D$(X,50) = AS$
148 LET D$(X,51) = AS$
149 LET D$(X,52) = AS$
150 LET D$(X,53) = AS$
151 LET D$(X,54) = AS$
152 LET D$(X,55) = AS$
153 LET D$(X,56) = AS$
154 LET D$(X,57) = AS$
155 LET D$(X,58) = AS$
156 LET D$(X,59) = AS$
157 LET D$(X,60) = AS$
158 LET D$(X,61) = AS$
159 LET D$(X,62) = AS$
160 LET D$(X,63) = AS$
161 LET D$(X,64) = AS$
162 LET D$(X,65) = AS$
163 LET D$(X,66) = AS$
164 LET D$(X,67) = AS$
165 LET D$(X,68) = AS$
166 LET D$(X,69) = AS$
167 LET D$(X,70) = AS$
168 LET D$(X,71) = AS$
169 LET D$(X,72) = AS$
170 LET D$(X,73) = AS$
171 LET D$(X,74) = AS$
172 LET D$(X,75) = AS$
173 LET D$(X,76) = AS$
174 LET D$(X,77) = AS$
175 LET D$(X,78) = AS$
176 LET D$(X,79) = AS$
177 LET D$(X,80) = AS$
178 LET D$(X,81) = AS$
179 LET D$(X,82) = AS$
180 LET D$(X,83) = AS$
181 LET D$(X,84) = AS$
182 LET D$(X,85) = AS$
183 LET D$(X,86) = AS$
184 LET D$(X,87) = AS$
185 LET D$(X,88) = AS$
186 LET D$(X,89) = AS$
187 LET D$(X,90) = AS$
188 LET D$(X,91) = AS$
189 LET D$(X,92) = AS$
190 LET D$(X,93) = AS$
191 LET D$(X,94) = AS$
192 LET D$(X,95) = AS$
193 LET D$(X,96) = AS$
194 LET D$(X,97) = AS$
195 LET D$(X,98) = AS$
196 LET D$(X,99) = AS$
197 LET D$(X,100) = AS$
198 LET D$(X,101) = AS$
199 LET D$(X,102) = AS$
200 LET D$(X,103) = AS$
201 LET D$(X,104) = AS$
202 LET D$(X,105) = AS$
203 LET D$(X,106) = AS$
204 LET D$(X,107) = AS$
205 LET D$(X,108) = AS$
206 LET D$(X,109) = AS$
207 LET D$(X,110) = AS$
208 LET D$(X,111) = AS$
209 LET D$(X,112) = AS$
210 LET D$(X,113) = AS$
211 LET D$(X,114) = AS$
212 LET D$(X,115) = AS$
213 LET D$(X,116) = AS$
214 LET D$(X,117) = AS$
215 LET D$(X,118) = AS$
216 LET D$(X,119) = AS$
217 LET D$(X,120) = AS$
218 LET D$(X,121) = AS$
219 LET D$(X,122) = AS$
220 LET D$(X,123) = AS$
221 LET D$(X,124) = AS$
222 LET D$(X,125) = AS$
223 LET D$(X,126) = AS$
224 LET D$(X,127) = AS$
225 LET D$(X,128) = AS$
226 LET D$(X,129) = AS$
227 LET D$(X,130) = AS$
228 LET D$(X,131) = AS$
229 LET D$(X,132) = AS$
230 LET D$(X,133) = AS$
231 LET D$(X,134) = AS$
232 LET D$(X,135) = AS$
233 LET D$(X,136) = AS$
234 LET D$(X,137) = AS$
235 LET D$(X,138) = AS$
236 LET D$(X,139) = AS$
237 LET D$(X,140) = AS$
238 LET D$(X,141) = AS$
239 LET D$(X,142) = AS$
240 LET D$(X,143) = AS$
241 LET D$(X,144) = AS$
242 LET D$(X,145) = AS$
243 LET D$(X,146) = AS$
244 LET D$(X,147) = AS$
245 LET D$(X,148) = AS$
246 LET D$(X,149) = AS$
247 LET D$(X,150) = AS$
248 LET D$(X,151) = AS$
249 LET D$(X,152) = AS$
250 LET D$(X,153) = AS$
251 LET D$(X,154) = AS$
252 LET D$(X,155) = AS$
253 LET D$(X,156) = AS$
254 LET D$(X,157) = AS$
255 LET D$(X,158) = AS$
256 LET D$(X,159) = AS$
257 LET D$(X,160) = AS$
258 LET D$(X,161) = AS$
259 LET D$(X,162) = AS$
260 LET D$(X,163) = AS$
261 LET D$(X,164) = AS$
262 LET D$(X,165) = AS$
263 LET D$(X,166) = AS$
264 LET D$(X,167) = AS$
265 LET D$(X,168) = AS$
266 LET D$(X,169) = AS$
267 LET D$(X,170) = AS$
268 LET D$(X,171) = AS$
269 LET D$(X,172) = AS$
270 LET D$(X,173) = AS$
271 LET D$(X,174) = AS$
272 LET D$(X,175) = AS$
273 LET D$(X,176) = AS$
274 LET D$(X,177) = AS$
275 LET D$(X,178) = AS$
276 LET D$(X,179) = AS$
277 LET D$(X,180) = AS$
278 LET D$(X,181) = AS$
279 LET D$(X,182) = AS$
280 LET D$(X,183) = AS$
281 LET D$(X,184) = AS$
282 LET D$(X,185) = AS$
283 LET D$(X,186) = AS$
284 LET D$(X,187) = AS$
285 LET D$(X,188) = AS$
286 LET D$(X,189) = AS$
287 LET D$(X,190) = AS$
288 LET D$(X,191) = AS$
289 LET D$(X,192) = AS$
290 LET D$(X,193) = AS$
291 LET D$(X,194) = AS$
292 LET D$(X,195) = AS$
293 LET D$(X,196) = AS$
294 LET D$(X,197) = AS$
295 LET D$(X,198) = AS$
296 LET D$(X,199) = AS$
297 LET D$(X,200) = AS$
298 LET D$(X,201) = AS$
299 LET D$(X,202) = AS$
300 LET D$(X,203) = AS$
301 LET D$(X,204) = AS$
302 LET D$(X,205) = AS$
303 LET D$(X,206) = AS$
304 LET D$(X,207) = AS$
305 LET D$(X,208) = AS$
306 LET D$(X,209) = AS$
307 LET D$(X,210) = AS$
308 LET D$(X,211) = AS$
309 LET D$(X,212) = AS$
310 LET D$(X,213) = AS$
311 LET D$(X,214) = AS$
312 LET D$(X,215) = AS$
313 LET D$(X,216) = AS$
314 LET D$(X,217) = AS$
315 LET D$(X,218) = AS$
316 LET D$(X,219) = AS$
317 LET D$(X,220) = AS$
318 LET D$(X,221) = AS$
319 LET D$(X,222) = AS$
320 LET D$(X,223) = AS$
321 LET D$(X,224) = AS$
322 LET D$(X,225) = AS$
323 LET D$(X,226) = AS$
324 LET D$(X,227) = AS$
325 LET D$(X,228) = AS$
326 LET D$(X,229) = AS$
327 LET D$(X,230) = AS$
328 LET D$(X,231) = AS$
329 LET D$(X,232) = AS$
330 LET D$(X,233) = AS$
331 LET D$(X,234) = AS$
332 LET D$(X,235) = AS$
333 LET D$(X,236) = AS$
334 LET D$(X,237) = AS$
335 LET D$(X,238) = AS$
336 LET D$(X,239) = AS$
337 LET D$(X,240) = AS$
338 LET D$(X,241) = AS$
339 LET D$(X,242) = AS$
340 LET D$(X,243) = AS$
341 LET D$(X,244) = AS$
342 LET D$(X,245) = AS$
343 LET D$(X,246) = AS$
344 LET D$(X,247) = AS$
345 LET D$(X,248) = AS$
346 LET D$(X,249) = AS$
347 LET D$(X,250) = AS$
348 LET D$(X,251) = AS$
349 LET D$(X,252) = AS$
350 LET D$(X,253) = AS$
351 LET D$(X,254) = AS$
352 LET D$(X,255) = AS$
353 LET D$(X,256) = AS$
354 LET D$(X,257) = AS$
355 LET D$(X,258) = AS$
356 LET D$(X,259) = AS$
357 LET D$(X,260) = AS$
358 LET D$(X,261) = AS$
359 LET D$(X,262) = AS$
360 LET D$(X,263) = AS$
361 LET D$(X,264) = AS$
362 LET D$(X,265) = AS$
363 LET D$(X,266) = AS$
364 LET D$(X,267) = AS$
365 LET D$(X,268) = AS$
366 LET D$(X,269) = AS$
367 LET D$(X,270) = AS$
368 LET D$(X,271) = AS$
369 LET D$(X,272) = AS$
370 LET D$(X,273) = AS$
371 LET D$(X,274) = AS$
372 LET D$(X,275) = AS$
373 LET D$(X,276) = AS$
374 LET D$(X,277) = AS$
375 LET D$(X,278) = AS$
376 LET D$(X,279) = AS$
377 LET D$(X,280) = AS$
378 LET D$(X,281) = AS$
379 LET D$(X,282) = AS$
380 LET D$(X,283) = AS$
381 LET D$(X,284) = AS$
382 LET D$(X,285) = AS$
383 LET D$(X,286) = AS$
384 LET D$(X,287) = AS$
385 LET D$(X,288) = AS$
386 LET D$(X,289) = AS$
387 LET D$(X,290) = AS$
388 LET D$(X,291) = AS$
389 LET D$(X,292) = AS$
390 LET D$(X,293) = AS$
391 LET D$(X,294) = AS$
392 LET D$(X,295) = AS$
393 LET D$(X,296) = AS$
394 LET D$(X,297) = AS$
395 LET D$(X,298) = AS$
396 LET D$(X,299) = AS$
397 LET D$(X,300) = AS$
398 LET D$(X,301) = AS$
399 LET D$(X,302) = AS$
400 LET D$(X,303) = AS$
401 LET D$(X,304) = AS$
402 LET D$(X,305) = AS$
403 LET D$(X,306) = AS$
404 LET D$(X,307) = AS$
405 LET D$(X,308) = AS$
406 LET D$(X,309) = AS$
407 LET D$(X,310) = AS$
408 LET D$(X,311) = AS$
409 LET D$(X,312) = AS$
410 LET D$(X,313) = AS$
411 LET D$(X,314) = AS$
412 LET D$(X,315) = AS$
413 LET D$(X,316) = AS$
414 LET D$(X,317) = AS$
415 LET D$(X,318) = AS$
416 LET D$(X,319) = AS$
417 LET D$(X,320) = AS$
418 LET D$(X,321) = AS$
419 LET D$(X,322) = AS$
420 LET D$(X,323) = AS$
421 LET D$(X,324) = AS$
422 LET D$(X,325) = AS$
423 LET D$(X,326) = AS$
424 LET D$(X,327) = AS$
425 LET D$(X,328) = AS$
426 LET D$(X,329) = AS$
427 LET D$(X,330) = AS$
428 LET D$(X,331) = AS$
429 LET D$(X,332) = AS$
430 LET D$(X,333) = AS$
431 LET D$(X,334) = AS$
432 LET D$(X,335) = AS$
433 LET D$(X,336) = AS$
434 LET D$(X,337) = AS$
435 LET D$(X,338) = AS$
436 LET D$(X,339) = AS$
437 LET D$(X,340) = AS$
438 LET D$(X,341) = AS$
439 LET D$(X,342) = AS$
440 LET D$(X,343) = AS$
441 LET D$(X,344) = AS$
442 LET D$(X,345) = AS$
443 LET D$(X,346) = AS$
444 LET D$(X,347) = AS$
445 LET D$(X,348) = AS$
446 LET D$(X,349) = AS$
447 LET D$(X,350) = AS$
448 LET D$(X,351) = AS$
449 LET D$(X,352) = AS$
450 LET D$(X,353) = AS$
451 LET D$(X,354) = AS$
452 LET D$(X,355) = AS$
453 LET D$(X,356) = AS$
454 LET D$(X,357) = AS$
455 LET D$(X,358) = AS$
456 LET D$(X,359) = AS$
457 LET D$(X,360) = AS$
458 LET D$(X,361) = AS$
459 LET D$(X,362) = AS$
460 LET D$(X,363) = AS$
461 LET D$(X,364) = AS$
462 LET D$(X,365) = AS$
463 LET D$(X,366) = AS$
464 LET D$(X,367) = AS$
465 LET D$(X,368) = AS$
466 LET D$(X,369) = AS$
467 LET D$(X,370) = AS$
468 LET D$(X,371) = AS$
469 LET D$(X,372) = AS$
470 LET D$(X,373) = AS$
471 LET D$(X,374) = AS$
472 LET D$(X,375) = AS$
473 LET D$(X,376) = AS$
474 LET D$(X,377) = AS$
475 LET D$(X,378) = AS$
476 LET D$(X,379) = AS$
477 LET D$(X,380) = AS$
478 LET D$(X,381) = AS$
479 LET D$(X,382) = AS$
480 LET D$(X,383) = AS$
481 LET D$(X,384) = AS$
482 LET D$(X,385) = AS$
483 LET D$(X,386) = AS$
484 LET D$(X,387) = AS$
485 LET D$(X,388) = AS$
486 LET D$(X,389) = AS$
487 LET D$(X,390) = AS$
488 LET D$(X,391) = AS$
489 LET D$(X,392) = AS$
490 LET D$(X,393) = AS$
491 LET D$(X,394) = AS$
492 LET D$(X,395) = AS$
493 LET D$(X,396) = AS$
494 LET D$(X,397) = AS$
495 LET D$(X,398) = AS$
496 LET D$(X,399) = AS$
497 LET D$(X,400) = AS$
498 LET D$(X,401) = AS$
499 LET D$(X,402) = AS$
500 LET D$(X,403) = AS$
501 LET D$(X,404) = AS$
502 LET D$(X,405) = AS$
503 LET D$(X,406) = AS$
504 LET D$(X,407) = AS$
505 LET D$(X,408) = AS$
506 LET D$(X,409) = AS$
507 LET D$(X,410) = AS$
508 LET D$(X,411) = AS$
509 LET D$(X,412) = AS$
510 LET D$(X,413) = AS$
511 LET D$(X,414) = AS$
512 LET D$(X,415) = AS$
513 LET D$(X,416) = AS$
514 LET D$(X,417) = AS$
515 LET D$(X,418) = AS$
516 LET D$(X,419) = AS$
517 LET D$(X,420) = AS$
518 LET D$(X,421) = AS$
519 LET D$(X,422) = AS$
520 LET D$(X,423) = AS$
521 LET D$(X,424) = AS$
522 LET D$(X,425) = AS$
523 LET D$(X,426) = AS$
524 LET D$(X,427) = AS$
525 LET D$(X,428) = AS$
526 LET D$(X,429) = AS$
527 LET D$(X,430) = AS$
528 LET D$(X,431) = AS$
529 LET D$(X,432) = AS$
530 LET D$(X,433) = AS$
531 LET D$(X,434) = AS$
532 LET D$(X,435) = AS$
533 LET D$(X,436) = AS$
534 LET D$(X,437) = AS$
535 LET D$(X,438) = AS$
536 LET D$(X,439) = AS$
537 LET D$(X,440) = AS$
538 LET D$(X,441) = AS$
539 LET D$(X,442) = AS$
540 LET D$(X,443) = AS$
541 LET D$(X,444) = AS$
542 LET D$(X,445) = AS$
543 LET D$(X,446) = AS$
544 LET D$(X,447) = AS$
545 LET D$(X,448) = AS$
546 LET D$(X,449) = AS$
547 LET D$(X,450) = AS$
548 LET D$(X,451) = AS$
549 LET D$(X,452) = AS$
550 LET D$(X,453) = AS$
551 LET D$(X,454) = AS$
552 LET D$(X,455) = AS$
553 LET D$(X,456) = AS$
554 LET D$(X,457) = AS$
555 LET D$(X,458) = AS$
556 LET D$(X,459) = AS$
557 LET D$(X,460) = AS$
558 LET D$(X,461) = AS$
559 LET D$(X,462) = AS$
560 LET D$(X,463) = AS$
561 LET D$(X,464) = AS$
562 LET D$(X,465) = AS$
563 LET D$(X,466) = AS$
564 LET D$(X,467) = AS$
565 LET D$(X,468) = AS$
566 LET D$(X,469) = AS$
567 LET D$(X,470) = AS$
568 LET D$(X,471) = AS$
569 LET D$(X,472) = AS$
570 LET D$(X,473) = AS$
571 LET D$(X,474) = AS$
572 LET D$(X,475) = AS$
573 LET D$(X,476) = AS$
574 LET D$(X,477) = AS$
575 LET D$(X,478) = AS$
576 LET D$(X,479) = AS$
577 LET D$(X,480) = AS$
578 LET D$(X,481) = AS$
579 LET D$(X,482) = AS$
580 LET D$(X,483) = AS$
581 LET D$(X,484) = AS$
582 LET D$(X,485) = AS$
583 LET D$(X,486) = AS$
584 LET D$(X,487) = AS$
585 LET D$(X,488) = AS$
586 LET D$(X,489) = AS$
587 LET D$(X,490) = AS$
588 LET D$(X,491) = AS$
589 LET D$(X,492) = AS$
590 LET D$(X,493) = AS$
591 LET D$(X,494) = AS$
592 LET D$(X,495) = AS$
593 LET D$(X,496) = AS$
594 LET D$(X,497) = AS$
595 LET D$(X,498) = AS$
596 LET D$(X,499) = AS$
597 LET D$(X,500) = AS$
598 LET D$(X,501) = AS$
599 LET D$(X,502) = AS$
600 LET D$(X,503) = AS$
601 LET D$(X,504) = AS$
602 LET D$(X,505) = AS$
603 LET D$(X,506) = AS$
604 LET D$(X,507) = AS$
605 LET D$(X,508) = AS$
606 LET D$(X,509) = AS$
607 LET D$(X,510) = AS$
608 LET D$(X,511) = AS$
609 LET D$(X,512) = AS$
610 LET D$(X,513) = AS$
611 LET D$(X,514) = AS$
612 LET D$(X,515) = AS$
613 LET D$(X,516) = AS$
614 LET D$(X,517) = AS$
615 LET D$(X,518) = AS$
616 LET D$(X,519) = AS$
617 LET D$(X,520) = AS$
618 LET D$(X,521) = AS$
619 LET D$(X,522) = AS$
620 LET D$(X,523) = AS$
621 LET D$(X,524) = AS$
622 LET D$(X,525) = AS$
623 LET D$(X,526) = AS$
624 LET D$(X,527) = AS$
625 LET D$(X,528) = AS$
626 LET D$(X,529) = AS$
627 LET D$(X,530) = AS$
628 LET D$(X,531) = AS$
629 LET D$(X,532) = AS$
630 LET D$(X,533) = AS$
631 LET D$(X,534) = AS$
632 LET D$(X,535) = AS$
633 LET D$(X,536) = AS$
634 LET D$(X,537) = AS$
635 LET D$(X,538) = AS$
636 LET D$(X,539) = AS$
637 LET D$(X,540) = AS$
638 LET D$(X,541) = AS$
639 LET D$(X,542) = AS$
640 LET D$(X,543) = AS$
641 LET D$(X,544) = AS$
642 LET D$(X,545) = AS$
643 LET D$(X,546) = AS$
644 LET D$(X,547) = AS$
645 LET D$(X,548) = AS$
646 LET D$(X,549) = AS$
647 LET D$(X,550) = AS$
648 LET D$(X,551) = AS$
649 LET D$(X,552) = AS$
650 LET D$(X,553) = AS$
651 LET D$(X,554) = AS$
652 LET D$(X,555) = AS$
653 LET D$(X,556) = AS$
654 LET D$(X,557) = AS$
655 LET D$(X,558) = AS$
656 LET D$(X,559) = AS$
657 LET D$(X,560) = AS$
658 LET D$(X,561) = AS$
659 LET D$(X,562) = AS$
660 LET D$(X,563) = AS$
661 LET D$(X,564) = AS$
662 LET D$(X,565) = AS$
663 LET D$(X,566) = AS$
664 LET D$(X,567) = AS$
665 LET D$(X,568) = AS$
666 LET D$(X,569) = AS$
667 LET D$(X,570) = AS$
668 LET D$(X,571) = AS$
669 LET D$(X,572) = AS$
670 LET D$(X,573) = AS$
671 LET D$(X,574) = AS$
672 LET D$(X,575) = AS$
673 LET D$(X,576) = AS$
674 LET D$(X,577) = AS$
675 LET D$(X,578) = AS$
676 LET D$(X,579) = AS$
677 LET D$(X,580) = AS$
678 LET D$(X,581) = AS$
679 LET D$(X,582) = AS$
680 LET D$(X,583) = AS$
681 LET D$(X,584) = AS$
682 LET D$(X,585) = AS$
683 LET D$(X,586) = AS$
684 LET D$(X,587) = AS$
685 LET D$(X,588) = AS$
686 LET D$(X,589) = AS$
687 LET D$(X,590) = AS$
688 LET D$(X,591) = AS$
689 LET D$(X,592) = AS$
690 LET D$(X,593) = AS$
691 LET D$(X,594) = AS$
692 LET D$(X,595) = AS$
693 LET D$(X,596) = AS$
694 LET D$(X,597) = AS$
695 LET D$(X,598) = AS$
696 LET D$(X,599) = AS$
697 LET D$(X,600) = AS$
698 LET D$(X,601) = AS$
699 LET D$(X,602) = AS$
700 LET D$(X,603) = AS$
701 LET D$(X,604) = AS$
702 LET D$(X,605) = AS$
703 LET D$(X,606) = AS$
704 LET D$(X,607) = AS$
705 LET D$(X,608) = AS$
706 LET D$(X,609) = AS$
707 LET D$(X,610) = AS$
708 LET D$(X,611) = AS$
709 LET D$(X,612) = AS$
710 LET D$(X,613) = AS$
711 LET D$(X,614) = AS$
712 LET D$(X,615) = AS$
713 LET D$(X,616) = AS$
714 LET D$(X,617) = AS$
715 LET D$(X,618) = AS$
716 LET D$(X,619) = AS$
717 LET D$(X,620) = AS$
718 LET D$(X,621) = AS$
719 LET D$(X,622) = AS$
720 LET D$(X,623) = AS$
721 LET D$(X,624) = AS$
722 LET D$(X,625) = AS$
723 LET D$(X,626) = AS$
724 LET D$(X,627) = AS$
725 LET D$(X,628) = AS$
726 LET D$(X,629) = AS$
727 LET D$(X,630) = AS$
728 LET D$(X,631) = AS$
729 LET D$(X,632) = AS$
730 LET D$(X,633) = AS$
731 LET D$(X,634) = AS$
732 LET D$(X,635) = AS$
733 LET D$(X,636) = AS$
734 LET D$(X,637) = AS$
735 LET D$(X,638) = AS$
736 LET D$(X,639) = AS$
737 LET D$(X,640) = AS$
738 LET D$(X,641) = AS$
739 LET D$(X,642) = AS$
740 LET D$(X,643) = AS
```


NOTA

Hay que tener cuidado en que no haya líneas de programa con el mismo número, ya que el resultado podría ser desastroso.

Comodidad de uso

Las sentencias "LOAD", "VERIFY" y "MERGE" tienen cierta particularidad que las permita cargar, verificar o combinar el primer programa que encuentren, aunque no conozcamos su nombre; para ello debemos sustituir el nombre por una cadena vacía.

Ejemplos:

```
LOAD ""  
LOAD "" CODE  
VERIFY ""  
VERIFY "" CODE  
MERGE ""
```

Búsqueda de programas

Durante la búsqueda de programas para su carga, verificación o fusión con otros, se van visualizando en pantalla todos aquellos programas o datos que el ordenador va encontrando. Esto puede servirnos de referencia para una mejor localización, ya que podemos utilizar el avance rápido hacia adelante o hacia atrás, de nuestro aparato de cassette, para ahorrar tiempo.

Referencias

Cuando el ordenador localiza un programa, se nos visualiza en pantalla su nom-

bre, así como una referencia indicativa del tipo de programa; éstas pueden ser de cuatro tipos:

- a) Programas BASIC.

Program : nombre

- b) Programas en C/M, pantallas o bytes.

Bytes : nombre

- c) Matrices numéricas.

Number array : nombre

- d) Matrices de cadena.

Character array : nombre

Errores

En el manejo de estas sentencias pueden aparecernos cualquiera de estos errores:

- a) Fuera de memoria.

4 out of memory

Este error se produce cuando no hay suficiente memoria para cargar o combinar un programa, bytes o matrices.

- b) Nombre de programa inválido.

F Invalid file name

Ocurre al grabar un programa con un nombre de más de diez caracteres o al asignarle una cadena vacía.

- c) Error de carga en la cinta.

R Tape loading error

Aparece cuando el ordena-

dor no puede cargar un programa o cuando la verificación ha sido incorrecta.

Este error puede ser debido entre otras causas a:

- Cabeza lectora del cassette sucia.
- Pilas gastadas en exceso.
- Cintas apelmazadas.
- Cintas deterioradas.
- Etc.

Programas

El programa número "1" permite conocer el contenido de una cinta de cassette, dando información de los programas almacenados, así como una serie de datos útiles: su comienzo y su longitud. Si tenemos conectada una impresora los datos saldrán simultáneamente.

Una vez ejecutado, nos pregunta el nombre que queremos dar a la cinta (máx. 15 caracteres) y a continuación la cara (o). Posteriormente se pone el cassette, con la cinta, en marcha y a esperar que el ordenador visualice la información según vaya detectando las cabeceras de los programas.

El programa utiliza una pequeña rutina en código máquina, ya que parte de la información que suministra es inaccesible desde el BASIC.

El programa número "2" es un programa de utilidad que permite conocer el directorio de una cinta, es decir los programas que hay grabados, así como una serie de datos anexos: Nombre del programador, marcaje del contador del cassette, etc. También permite elegir un programa y cargarlo automáticamente.

Letra Normal de matriz de puntos
Letra Itálica o cursiva
Letra de calidad (NLQ)
Escritura en negrita
Escritura subrayada

Paso de escritura condensado

Paso de escritura "Elite"

Paso de escritura "Pica"

Itálica condensada

Línea con superíndices y subíndices

La letra sea pequeña posible

Normal expandido

Itálica expandida

Condensado expandido

Itálica condensada expandida

Diversos tipos de letra, proporcionados por una impresora de calidad.

acoplar cualquier impresora, que no sea "ZX", al Spectrum. Los protocolos de comunicación más importantes, por los que se rigen estas impresoras, son:

CENTRONICS
RS-232

El protocolo "CENTRONICS" se base en el envío paralelo de la información, mientras que la "RS-232" lo realiza en serie.

La mayoría de estos Interfaces ocupan una pequeña parte de la memoria del Spectrum y necesitan de un Software que las gestione.

Tipos de Impresora

En la actualidad existe una gran variedad. Atendiendo a

su sistema de funcionamiento se pueden clasificar de la siguiente manera:

MARGARITA
AGUJAS
TÉRMICAS
LINER
INYECCIÓN DE TINTA
ELECTROSTÁTICAS
LASER
ETC...

Las impresoras de Margarita tienen la ventaja de tener un tipo de letra idéntico al proporcionado por una máquina de escribir, pero tienen el inconveniente de que su velocidad expresada en "cps" (caracteres por segundo) es relativamente baja.

Las impresoras de agujas son las más utilizadas en los

ordenadores personales, ya que su relación servicio/prestación es bastante adecuada.

La velocidad de estas impresoras es superior a las de Margarita, pero su tipo de letra es más imperfecto, ya que las realiza a base de puntos.

Las impresoras térmicas también se utilizan en los ordenadores personales; en éstas, el tipo de papel utilizado es especial, ya que cambian de color al calentarse.

El resto de impresoras no son de uso frecuente con ordenadores personales, ya que su precio es bastante elevado y están diseñadas para equipos donde el volumen de impresión es grande; por ejemplo, las de líneas imprimen una línea de una sola vez y las de láser hasta una página.

Los listados por impresora son muy convenientes, ya que ayudan en la depuración de programas al poder localizar con mayor facilidad las sentencias.



Acceso al teclado

LN



Definición

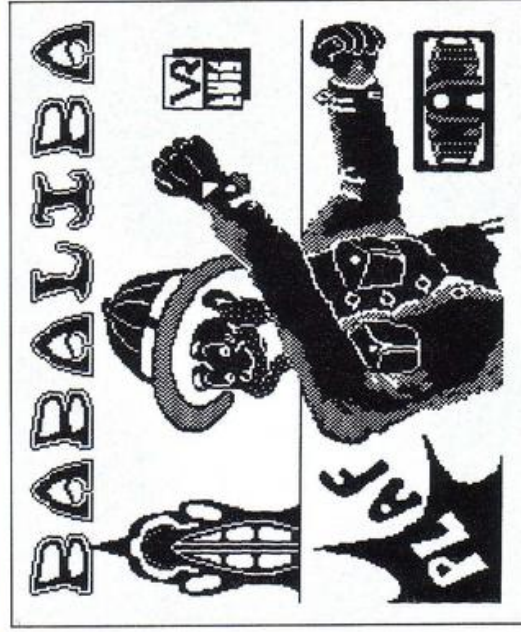
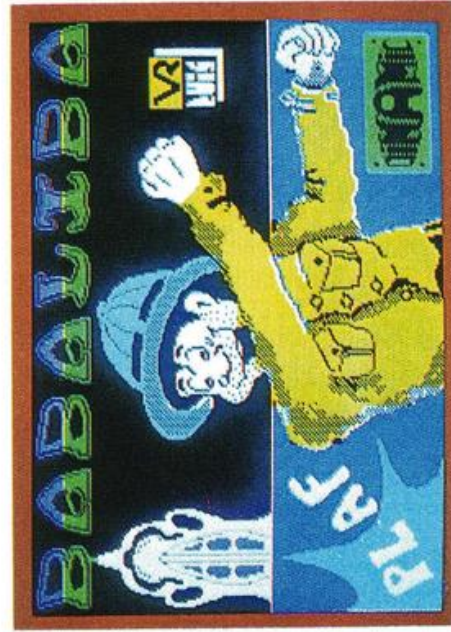
La sentencia "COPY" permite sacar por impresora una copia de la imagen visualizada en pantalla.

"COPY" no necesita de ningún argumento para ejecutarse.

La reproducción que hace "COPY" de la pantalla se basa en la impresión de los píxeles con color de "tinta", por tanto, no se sorprenda si al intentar realizar un "COPY" del dibujo proporcionado por el programa de la página 227, no se imprime absolutamente nada, ya que está realizado a base de bloques de color "papel" con el carácter "espacio"; si desea imprimirlo, modifique la siguiente línea:



Tampoco se imprime nada si intenta realizar un "COPY" de los listados que proporcionó el ordenador al pulsar la



Sólo los píxeles con color de tinta se imprimen al realizar un "copy".

tecla "ENTER", ya que éstos se borran al ejecutarse una sentencia o comando.

Realice prácticas con los ejercicios de las sentencias "PLOT", "DRAW" y "CIRCLE".

Programa de utilidad

El programa n.º 1 permite realizar uno de los típicos TEST de impresora, en los

Otras impresoras

En el mercado existen diversos Interfaces para poder

Las instrucciones de manejo se encuentran en el propio programa.

Los datos referentes a esa cinta deben ser editados con el programa número "3" que permite confeccionar la información de un máximo de 20 programas.

Los programas deben ser grabados en el siguiente orden:

- El programa número "2" como LINE 6000.
- La tabla de datos que genera el programa número "3".
- A continuación los pro-

```
PROGRAMA N. 2 > QWERTY
CONTADOR > 123
PROGRAMADOR > D.F.G.
FECHA > 23/04/85
```

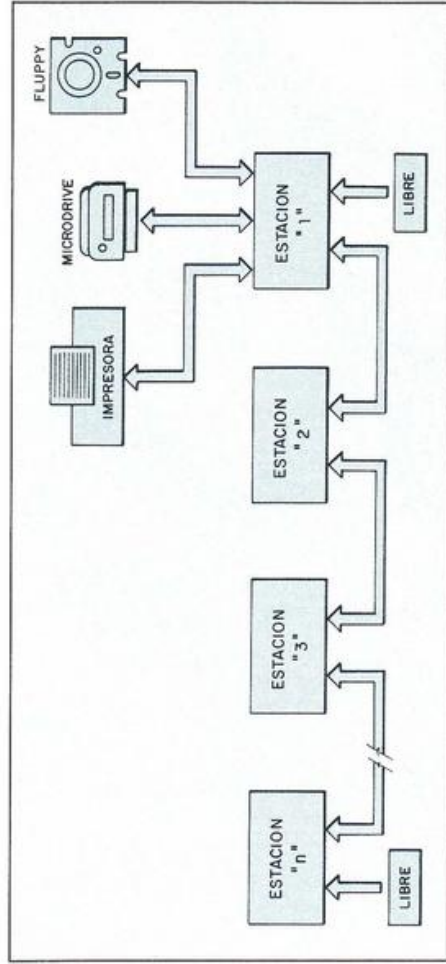
```
COMENTARIOS > ESTE PROGRAMA
EXPLORA EL TECLADO DEL SPECTRUM
```

Edición de datos con el "EDIT/DIR".

gramas correspondientes a esa cinta. Como el programa "directorio" es el primero de la cinta, para ejecutarlo rebo-

bina la cinta y teclea:

LOAD ""



Estructura de conexionado.

de modificar el catálogo para que no se visualice el nombre del programa borrado.

ERASE

Acceso al teclado

WHITE



ERASE

Su estructura general es:

SENTENCIA	ARGUMENTO
ERASE	"m"; unidad; "nombre"

Ejemplos:

— ERASE "m"; 1; "pepe"
— ERASE "m"; 5; "BANK"

Ejecución automática

Utilizando el Microdrive y siguiendo unas cuantas reglas podrá ejecutar de forma automática aquel programa

que más utilice de un cartucho.

El programa deberá grabarse de la siguiente forma:

SAVE "m"; 1; "m" LINE n

donde "n" es el número de línea para su autoejecución. Para utilizar la facilidad de ejecución automática tendrá que:

- Insertar el cartucho en la unidad primera del Microdrive.
- Utilizar el programa después de haber conectado la alimentación del Spectrum o después de haber introducido el comando "NEW".

Teclee el comando "RUN" sin ningún argumento y el programa grabado con el nombre "run" se cargará en la memoria del Spectrum y se autoejecutará.

Protección de ficheros

Básicamente existen tres métodos para proteger sus programas:

- a) Quitar la lengüeta de plástico situada en uno de los

laterales del cartucho. De esta manera no se podrán grabar o borrar ningún programa.

- b) Grabar los programas a proteger con "LINE n". De esta manera no se podrá realizar "MERGE" para visualizar su contenido.

- c) Al realizar la grabación de un fichero anteponga el código 0 al nombre

SAVE "m"; 1; CHR\$ 0 + "nombre"

Cuando intente obtener un catálogo del cartucho, el nombre de ese fichero no se visualizará, por tanto deberá recordarlo para poder cargarlo en otra ocasión.

LOAD "m"; 1; CHR\$ 0 + "nombre"

Ficheros de datos

Aparte de los ficheros de programas, el usuario puede abrir y cerrar ficheros de datos tanto para grabación como para lectura.

US ASCII

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

UK ASCII

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

FRENCH/BELGIAN

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

GERMAN

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

ITALIAN

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

SWEDISH/FINNISH

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

DANISH/NORWEGIAN

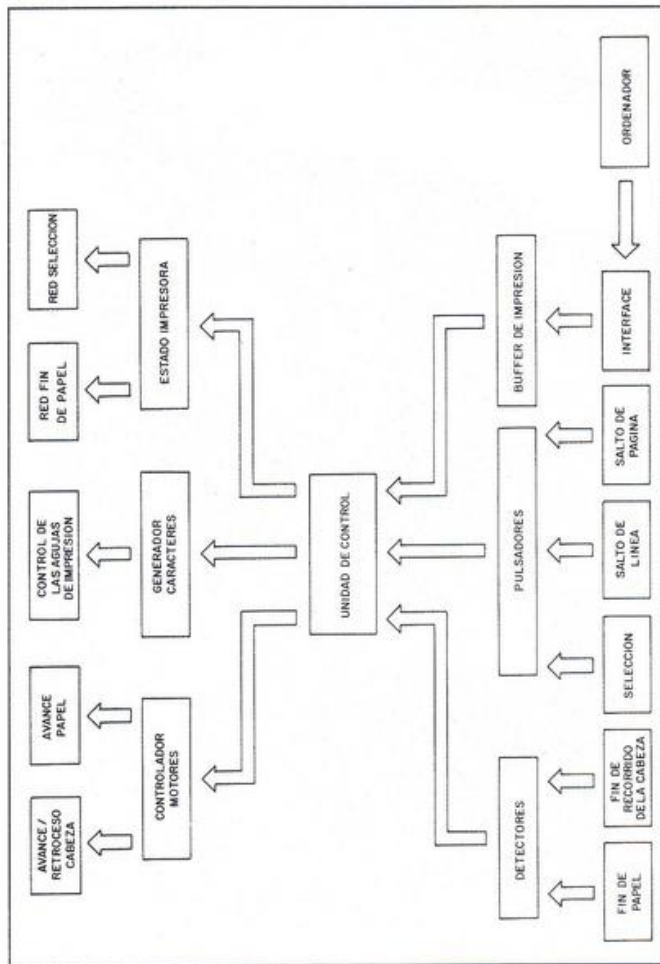
```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

SPANISH

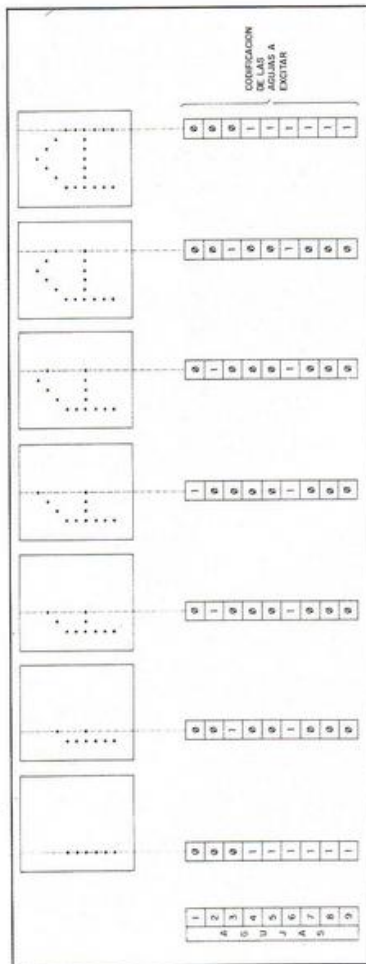
```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`
```

ACTUAL CHARACTERSET SPANISH

Diversos juegos de caracteres de una impresora.



Esquema de bloques de una impresora.



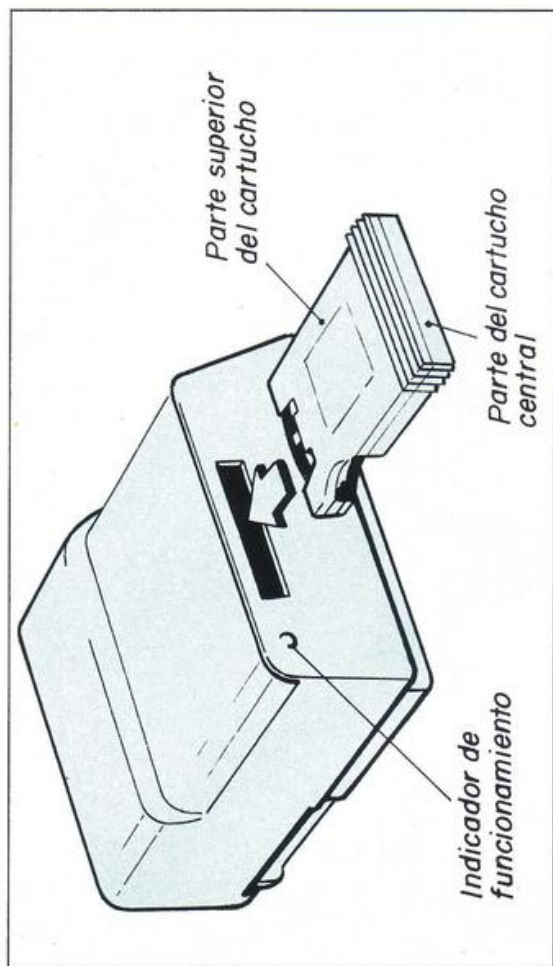
Formación del carácter "A" con una impresora de agujas de 9 x 7.

- GESTION DE IMPRESORAS -

Las impresoras son unos periféricos similares a una máquina de escribir, pero sin teclado incorporado, ya que las órdenes de escritura proceden del ordenador al que están conectados.

Las utilidades de una impresora son diversas, desde la simple obtención en papel de los listados de programas hasta la conexión, por ejemplo, de cartas con un sofisticado "PROCESADOR DE TEXTOS".

Las impresoras son una herramienta imprescindible en un "Procesador de Textos".



Unidad de Microdrive mostrando la correcta inserción del cartucho.

CARGA

Grabación y carga

Ejemplos:

- CAT 5
- CAT j
- CAT (n - 1) * 2
- CAT VAL b\$

La forma de presentar un catálogo o directorio es la siguiente:

- Nombre asignado al formato.

- Lista ordenada por orden alfabético de todos los ficheros.

- Capacidad libre expresada en Kbytes.

Utilizando otra estructura se puede enviar el catálogo a otro canal de comunicación.

Todas las opciones de grabación y carga de programas en cassettes están disponibles para su utilización con el Interface-1, solamente varía la sintaxis de la instrucción.

GRABACION

SENTENCIA	ARGUMENTO
SAVE	* "m"; unidad; "nombre"

Ejemplos:

- SAVE * "m"; 2; "cuat"
- SAVE * "m"; 3; "polo"

El símbolo del asterisco indica al sistema operativo que la grabación debe efectuarse a través del Interface -1, y no del cassette.

VERIFICACION

SENTENCIA	ARGUMENTO
VERIFY	* "m"; unidad; "nombre"

Ejemplos:

- VERIFY * "m"; 8; "vale"
- VERIFY * "m"; 1; "OKEY"

La anterior instrucción envía el catálogo de la unidad Microdrive número 5 a una impresora del tipo ZX.

Ejemplo:

CAT # 3; 5

SENTENCIA	ARGUMENTO
LOAD	* "m"; unidad; "nombre"

Ejemplos:

- LOAD * "m"; 5; "Pelota"
- LOAD * "m"; 3; "ZX"

COMBINACION

SENTENCIA	ARGUMENTO
MERGE	* "m"; unidad; "nombre"

Ejemplos:

- MERGE * "m"; 7; "GO"
- MERGE * "m"; 2; "loro"

Borrado de programas

Una operación engorrosa es el borrado de programas en cassette, sin embargo, cualquier programa grabado en Microdrive puede ser borrado con facilidad utilizando el comando "ERASE", que se encarga, entre otras cosas,

OPEN #8; "n"; 4

c) Asignación de la corriente #15 a una impresora conectada al interface RS-232.

OPEN #15; "1"

d) Para asignar una corriente a cualquiera de los canales: "k" (teclado) "s" (pantalla) o "p" (impresora) debe utilizarse necesariamente el separador "coma"; en los demás canales, puede utilizarse indistintamente "coma" o "punto y coma".

OPEN #5; "s"
OPEN #7; "p"
OPEN #10; "k"

Desactivación de canales y corrientes

Para quitar la asignación de un "canal" a su correspondiente "corriente", se utiliza la sentencia "CLOSE #".

CLOSE #

Acceso al teclado

CYAN



MODO E



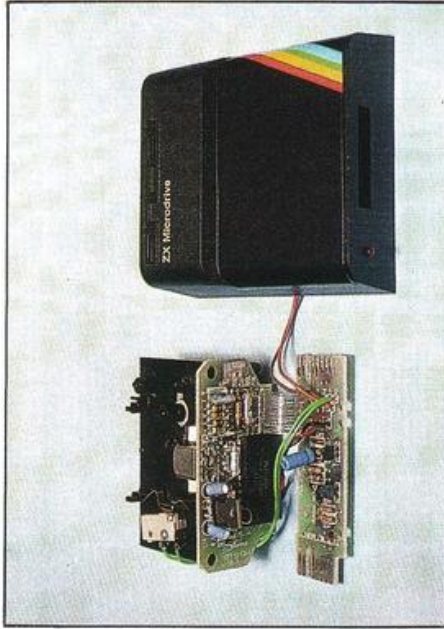
CLOSE #

BOL SHIFT

Es un periférico diseñado para almacenar gran cantidad de datos, con un tiempo de acceso de lectura/escritura superior al proporcionado por un cassette. La información se almacena en unos **cartuchos** intercambiables.

Las operaciones que pueden realizarse con el Microdrive son:

- Formatear o inicializar cartuchos.
- Almacenar y verificar



Interior del Microdrive

Su estructura general es:

SENTENCIA	ARGUMENTO
CLOSE #	corriente

Ejemplos:

- CLOSE # 6
- CLOSE # n
- CLOSE # z + 1
- CLOSE # VAL a\$

Al realizar la desactivación, los datos que estuvieron almacenados en la memoria intermedia serán transmitidos a su correspondiente canal.

El Microdrive

- programas en cartucho.
- Cargar programas.
- Borrar ficheros de datos y programas.
- Combinar los programas residentes en memoria con los archivados en cartucho.
- Catalogar cartuchos, es decir, obtener un *directorio* o listado de los programas grabados en él.
- Abrir y cerrar ficheros de datos.
- Grabar y leer datos de un fichero.
- Mover datos entre ficheros.

FORMAT

Acceso al teclado

BLACK

DELETE

MODO E

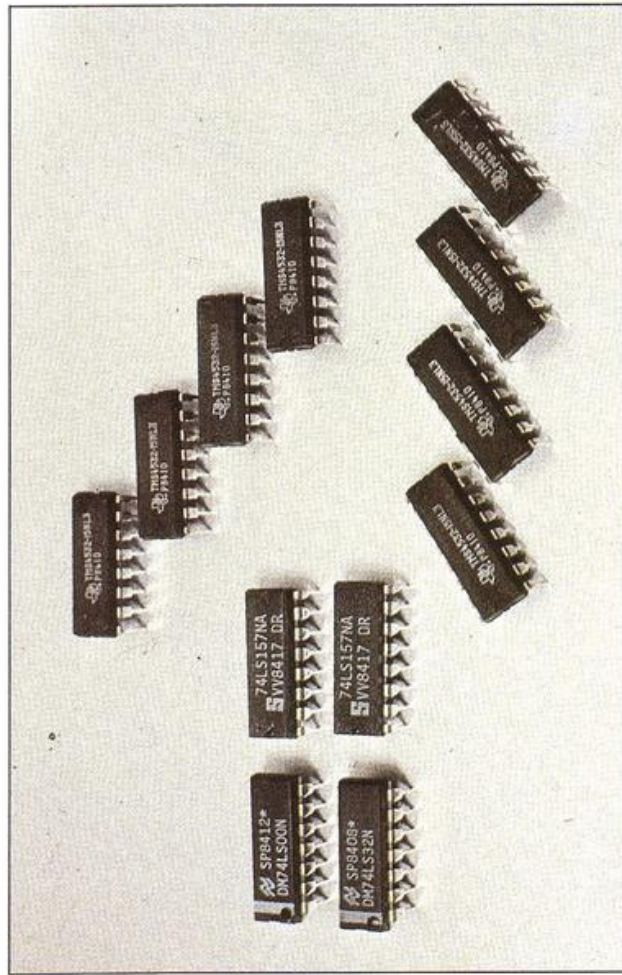


FORMAT

PROGRAMA 1

```
10 REM
*****
** CURSO/BASIC **
*****
** FICHERO **
*****
*****
L5 BORDER 4: PAPER 4: INK 0: C
32 POKE 23658,8
38 LET ultimo=0
100 LET $=(100,40)
500 REM
510 CLS
520 PRINT INVERSE 1; " FICHE
RO BIBLIOTECARIO
522 PRINT INVERSE 1; AT 5,10; "
OPCIONES
530 PRINT AT 9,10; "1. GENERAR"
540 PRINT AT 12,10; "2. ALMACENAR"
550 PRINT AT 15,10; "3. RECUPERAR"
552 PRINT AT 18,10; "4. VISUALIZA
R"
560 PRINT #0; AT 1,4; "pulse la o
pcion deseada"
562 PAUSE 0
570 LET z$=INKEY$
580 IF z$="1" OR z$="4" THEN BE
EP 0.2; 10; GO TO 562
590 BEEP 0.2; 10; GO TO (VAL z$)
600 REM
610 CLS
620 IF ultimo=0 THEN GO TO 1020
630 IF ultimo>0 AND ultimo<=99
THEN GO TO 1015
640 IF ultimo=99 THEN PRINT INU
VERSE 0; AT 1,4; "FICHERO COMPLET
O"
650 INVERSE 0; AT 10,7; "NUEVO
FICHERO"
660 AT 15,7; "MENU OPCIONE
s"
670 AT 1,4; "pulse la opcion de
seada"
680 PAUSE 0; LET z$=INKEY$
690 IF z$="N" THEN BEEP 0.2; 10;
DIM a$(100,40); LET ultimo=0; L
ET $=""; LET f$=""; GO TO 1020
700 IF z$="H" THEN BEEP 0.2; 10;
GO TO 1020
710 IF z$="1" THEN GO TO 1011
720 IF z$="2" THEN GO TO 1011
730 IF z$="3" THEN GO TO 1011
740 IF z$="4" THEN GO TO 1011
750 IF z$="5" THEN GO TO 1011
760 IF z$="6" THEN GO TO 1011
770 IF z$="7" THEN GO TO 1011
780 IF z$="8" THEN GO TO 1011
790 IF z$="9" THEN GO TO 1011
800 IF z$="0" THEN GO TO 1011
810 IF z$="." THEN GO TO 1011
820 IF z$="," THEN GO TO 1011
830 IF z$=";" THEN GO TO 1011
840 IF z$=":" THEN GO TO 1011
850 IF z$="@" THEN GO TO 1011
860 IF z$="A" THEN GO TO 1011
870 IF z$="B" THEN GO TO 1011
880 IF z$="C" THEN GO TO 1011
890 IF z$="D" THEN GO TO 1011
900 IF z$="E" THEN GO TO 1011
910 IF z$="F" THEN GO TO 1011
920 IF z$="G" THEN GO TO 1011
930 IF z$="H" THEN GO TO 1011
940 IF z$="I" THEN GO TO 1011
950 IF z$="J" THEN GO TO 1011
960 IF z$="K" THEN GO TO 1011
970 IF z$="L" THEN GO TO 1011
980 IF z$="M" THEN GO TO 1011
990 IF z$="N" THEN GO TO 1011
1000 IF z$="O" THEN GO TO 1011
1010 IF z$="P" THEN GO TO 1011
1020 CLS
1030 PRINT AT 7,0; "NOMBRE"
1040 PRINT AT 14,0; "AUTOR"
1050 FOR n=1 TO 100
1060 LET n$=STR$ n;
1070 PRINT AT 0,22; INVERSE 1;n$
1080 INPUT "Nombre"; LINE b$;
1090 IF b$="FIN" THEN GO TO 500
1100 LET d$=""; THEN GO TO 1060
1110 IF LEN b$>20 THEN LET b$=b$
(1 TO 20)
1120 INPUT "Go to"; LINE c$;
1130 IF c$="1" THEN GO TO 1060
1140 LET ultimo=ultimo+1
```

```
1150 PAUSE 50
1160 PRINT AT 7,9; "
1170 PRINT AT 14,9; "
1180 NEXT n
1190 GO TO 500
1200 REM
1210 CLS
1220 PRINT INVERSE 1; " GRAB
ACION DE FICHERO
1230 INPUT "Nombre"; AT 15,0; CLAVE
1240 INPUT "Nombre"; LINE c$; B
EEP 0.2; 10; THEN GO TO 2040
2050 IF b$=" " THEN LET b$=b$+
" "
2060 IF LEN b$>6 THEN LET b$=b$+
" "
2070 INPUT "Fecha"; LINE c$; B
EEP 0.2; 10; THEN GO TO 2080
2080 IF c$=" " THEN LET c$=c$+
" "
2090 IF LEN c$>8 THEN LET c$=c$+
" "
2100 PRINT AT 10,9; c$
2110 FOR n=0 TO 9
2120 LET d$=" "
2130 PAUSE 0
2140 IF n=0 THEN LET d$="1"
2150 IF n=1 THEN LET d$="2"
2160 IF n=2 THEN LET d$="3"
2170 IF n=3 THEN LET d$="4"
2180 IF n=4 THEN LET d$="5"
2190 IF n=5 THEN LET d$="6"
2200 IF n=6 THEN LET d$="7"
2210 IF n=7 THEN LET d$="8"
2220 IF n=8 THEN LET d$="9"
2230 IF n=9 THEN LET d$="0"
2240 IF n=10 THEN LET d$="1"
2250 IF n=11 THEN LET d$="2"
2260 IF n=12 THEN LET d$="3"
2270 IF n=13 THEN LET d$="4"
2280 IF n=14 THEN LET d$="5"
2290 IF n=15 THEN LET d$="6"
2300 IF n=16 THEN LET d$="7"
2310 IF n=17 THEN LET d$="8"
2320 IF n=18 THEN LET d$="9"
2330 IF n=19 THEN LET d$="0"
2340 IF n=20 THEN LET d$="1"
2350 IF n=21 THEN LET d$="2"
2360 IF n=22 THEN LET d$="3"
2370 IF n=23 THEN LET d$="4"
2380 IF n=24 THEN LET d$="5"
2390 IF n=25 THEN LET d$="6"
2400 IF n=26 THEN LET d$="7"
2410 IF n=27 THEN LET d$="8"
2420 IF n=28 THEN LET d$="9"
2430 IF n=29 THEN LET d$="0"
2440 IF n=30 THEN LET d$="1"
2450 IF n=31 THEN LET d$="2"
2460 IF n=32 THEN LET d$="3"
2470 IF n=33 THEN LET d$="4"
2480 IF n=34 THEN LET d$="5"
2490 IF n=35 THEN LET d$="6"
2500 IF n=36 THEN LET d$="7"
2510 IF n=37 THEN LET d$="8"
2520 IF n=38 THEN LET d$="9"
2530 IF n=39 THEN LET d$="0"
2540 IF n=40 THEN LET d$="1"
2550 IF n=41 THEN LET d$="2"
2560 IF n=42 THEN LET d$="3"
2570 IF n=43 THEN LET d$="4"
2580 IF n=44 THEN LET d$="5"
2590 IF n=45 THEN LET d$="6"
2600 IF n=46 THEN LET d$="7"
2610 IF n=47 THEN LET d$="8"
2620 IF n=48 THEN LET d$="9"
2630 IF n=49 THEN LET d$="0"
2640 IF n=50 THEN LET d$="1"
2650 IF n=51 THEN LET d$="2"
2660 IF n=52 THEN LET d$="3"
2670 IF n=53 THEN LET d$="4"
2680 IF n=54 THEN LET d$="5"
2690 IF n=55 THEN LET d$="6"
2700 IF n=56 THEN LET d$="7"
2710 IF n=57 THEN LET d$="8"
2720 IF n=58 THEN LET d$="9"
2730 IF n=59 THEN LET d$="0"
2740 IF n=60 THEN LET d$="1"
2750 IF n=61 THEN LET d$="2"
2760 IF n=62 THEN LET d$="3"
2770 IF n=63 THEN LET d$="4"
2780 IF n=64 THEN LET d$="5"
2790 IF n=65 THEN LET d$="6"
2800 IF n=66 THEN LET d$="7"
2810 IF n=67 THEN LET d$="8"
2820 IF n=68 THEN LET d$="9"
2830 IF n=69 THEN LET d$="0"
2840 IF n=70 THEN LET d$="1"
2850 IF n=71 THEN LET d$="2"
2860 IF n=72 THEN LET d$="3"
2870 IF n=73 THEN LET d$="4"
2880 IF n=74 THEN LET d$="5"
2890 IF n=75 THEN LET d$="6"
2900 IF n=76 THEN LET d$="7"
2910 IF n=77 THEN LET d$="8"
2920 IF n=78 THEN LET d$="9"
2930 IF n=79 THEN LET d$="0"
2940 IF n=80 THEN LET d$="1"
2950 IF n=81 THEN LET d$="2"
2960 IF n=82 THEN LET d$="3"
2970 IF n=83 THEN LET d$="4"
2980 IF n=84 THEN LET d$="5"
2990 IF n=85 THEN LET d$="6"
3000 IF n=86 THEN LET d$="7"
3010 IF n=87 THEN LET d$="8"
3020 IF n=88 THEN LET d$="9"
3030 IF n=89 THEN LET d$="0"
3040 IF n=90 THEN LET d$="1"
3050 IF n=91 THEN LET d$="2"
3060 IF n=92 THEN LET d$="3"
3070 IF n=93 THEN LET d$="4"
3080 IF n=94 THEN LET d$="5"
3090 IF n=95 THEN LET d$="6"
3100 IF n=96 THEN LET d$="7"
3110 IF n=97 THEN LET d$="8"
3120 IF n=98 THEN LET d$="9"
3130 IF n=99 THEN LET d$="0"
3140 IF n=100 THEN LET d$="1"
3150 IF n=101 THEN LET d$="2"
3160 IF n=102 THEN LET d$="3"
3170 IF n=103 THEN LET d$="4"
3180 IF n=104 THEN LET d$="5"
3190 IF n=105 THEN LET d$="6"
3200 IF n=106 THEN LET d$="7"
3210 IF n=107 THEN LET d$="8"
3220 IF n=108 THEN LET d$="9"
3230 IF n=109 THEN LET d$="0"
3240 IF n=110 THEN LET d$="1"
3250 IF n=111 THEN LET d$="2"
3260 IF n=112 THEN LET d$="3"
3270 IF n=113 THEN LET d$="4"
3280 IF n=114 THEN LET d$="5"
3290 IF n=115 THEN LET d$="6"
3300 IF n=116 THEN LET d$="7"
3310 IF n=117 THEN LET d$="8"
3320 IF n=118 THEN LET d$="9"
3330 IF n=119 THEN LET d$="0"
3340 IF n=120 THEN LET d$="1"
3350 IF n=121 THEN LET d$="2"
3360 IF n=122 THEN LET d$="3"
3370 IF n=123 THEN LET d$="4"
3380 IF n=124 THEN LET d$="5"
3390 IF n=125 THEN LET d$="6"
3400 IF n=126 THEN LET d$="7"
3410 IF n=127 THEN LET d$="8"
3420 IF n=128 THEN LET d$="9"
3430 IF n=129 THEN LET d$="0"
3440 IF n=130 THEN LET d$="1"
3450 IF n=131 THEN LET d$="2"
3460 IF n=132 THEN LET d$="3"
3470 IF n=133 THEN LET d$="4"
3480 IF n=134 THEN LET d$="5"
3490 IF n=135 THEN LET d$="6"
3500 IF n=136 THEN LET d$="7"
3510 IF n=137 THEN LET d$="8"
3520 IF n=138 THEN LET d$="9"
3530 IF n=139 THEN LET d$="0"
3540 IF n=140 THEN LET d$="1"
3550 IF n=141 THEN LET d$="2"
3560 IF n=142 THEN LET d$="3"
3570 IF n=143 THEN LET d$="4"
3580 IF n=144 THEN LET d$="5"
3590 IF n=145 THEN LET d$="6"
3600 IF n=146 THEN LET d$="7"
3610 IF n=147 THEN LET d$="8"
3620 IF n=148 THEN LET d$="9"
3630 IF n=149 THEN LET d$="0"
3640 IF n=150 THEN LET d$="1"
3650 IF n=151 THEN LET d$="2"
3660 IF n=152 THEN LET d$="3"
3670 IF n=153 THEN LET d$="4"
3680 IF n=154 THEN LET d$="5"
3690 IF n=155 THEN LET d$="6"
3700 IF n=156 THEN LET d$="7"
3710 IF n=157 THEN LET d$="8"
3720 IF n=158 THEN LET d$="9"
3730 IF n=159 THEN LET d$="0"
3740 IF n=160 THEN LET d$="1"
3750 IF n=161 THEN LET d$="2"
3760 IF n=162 THEN LET d$="3"
3770 IF n=163 THEN LET d$="4"
3780 IF n=164 THEN LET d$="5"
3790 IF n=165 THEN LET d$="6"
3800 IF n=166 THEN LET d$="7"
3810 IF n=167 THEN LET d$="8"
3820 IF n=168 THEN LET d$="9"
3830 IF n=169 THEN LET d$="0"
3840 IF n=170 THEN LET d$="1"
3850 IF n=171 THEN LET d$="2"
3860 IF n=172 THEN LET d$="3"
3870 IF n=173 THEN LET d$="4"
3880 IF n=174 THEN LET d$="5"
3890 IF n=175 THEN LET d$="6"
3900 IF n=176 THEN LET d$="7"
3910 IF n=177 THEN LET d$="8"
3920 IF n=178 THEN LET d$="9"
3930 IF n=179 THEN LET d$="0"
3940 IF n=180 THEN LET d$="1"
3950 IF n=181 THEN LET d$="2"
3960 IF n=182 THEN LET d$="3"
3970 IF n=183 THEN LET d$="4"
3980 IF n=184 THEN LET d$="5"
3990 IF n=185 THEN LET d$="6"
4000 IF n=186 THEN LET d$="7"
4010 IF n=187 THEN LET d$="8"
4020 IF n=188 THEN LET d$="9"
4030 IF n=189 THEN LET d$="0"
4040 IF n=190 THEN LET d$="1"
4050 IF n=191 THEN LET d$="2"
4060 IF n=192 THEN LET d$="3"
4070 IF n=193 THEN LET d$="4"
4080 IF n=194 THEN LET d$="5"
4090 IF n=195 THEN LET d$="6"
4100 IF n=196 THEN LET d$="7"
4110 IF n=197 THEN LET d$="8"
4120 IF n=198 THEN LET d$="9"
4130 IF n=199 THEN LET d$="0"
4140 IF n=200 THEN LET d$="1"
4150 IF n=201 THEN LET d$="2"
4160 IF n=202 THEN LET d$="3"
4170 IF n=203 THEN LET d$="4"
4180 IF n=204 THEN LET d$="5"
4190 IF n=205 THEN LET d$="6"
4200 IF n=206 THEN LET d$="7"
4210 IF n=207 THEN LET d$="8"
4220 IF n=208 THEN LET d$="9"
4230 IF n=209 THEN LET d$="0"
4240 IF n=210 THEN LET d$="1"
4250 IF n=211 THEN LET d$="2"
4260 IF n=212 THEN LET d$="3"
4270 IF n=213 THEN LET d$="4"
4280 IF n=214 THEN LET d$="5"
4290 IF n=215 THEN LET d$="6"
4300 IF n=216 THEN LET d$="7"
4310 IF n=217 THEN LET d$="8"
4320 IF n=218 THEN LET d$="9"
4330 IF n=219 THEN LET d$="0"
4340 IF n=220 THEN LET d$="1"
4350 IF n=221 THEN LET d$="2"
4360 IF n=222 THEN LET d$="3"
4370 IF n=223 THEN LET d$="4"
4380 IF n=224 THEN LET d$="5"
4390 IF n=225 THEN LET d$="6"
4400 IF n=226 THEN LET d$="7"
4410 IF n=227 THEN LET d$="8"
4420 IF n=228 THEN LET d$="9"
4430 IF n=229 THEN LET d$="0"
4440 IF n=230 THEN LET d$="1"
4450 IF n=231 THEN LET d$="2"
4460 IF n=232 THEN LET d$="3"
4470 IF n=233 THEN LET d$="4"
4480 IF n=234 THEN LET d$="5"
4490 IF n=235 THEN LET d$="6"
4500 IF n=236 THEN LET d$="7"
4510 IF n=237 THEN LET d$="8"
4520 IF n=238 THEN LET d$="9"
4530 IF n=239 THEN LET d$="0"
4540 IF n=240 THEN LET d$="1"
4550 IF n=241 THEN LET d$="2"
4560 IF n=242 THEN LET d$="3"
4570 IF n=243 THEN LET d$="4"
4580 IF n=244 THEN LET d$="5"
4590 IF n=245 THEN LET d$="6"
4600 IF n=246 THEN LET d$="7"
4610 IF n=247 THEN LET d$="8"
4620 IF n=248 THEN LET d$="9"
4630 IF n=249 THEN LET d$="0"
4640 IF n=250 THEN LET d$="1"
4650 IF n=251 THEN LET d$="2"
4660 IF n=252 THEN LET d$="3"
4670 IF n=253 THEN LET d$="4"
4680 IF n=254 THEN LET d$="5"
4690 IF n=255 THEN LET d$="6"
4700 IF n=256 THEN LET d$="7"
4710 IF n=257 THEN LET d$="8"
4720 IF n=258 THEN LET d$="9"
4730 IF n=259 THEN LET d$="0"
4740 IF n=260 THEN LET d$="1"
4750 IF n=261 THEN LET d$="2"
4760 IF n=262 THEN LET d$="3"
4770 IF n=263 THEN LET d$="4"
4780 IF n=264 THEN LET d$="5"
4790 IF n=265 THEN LET d$="6"
4800 IF n=266 THEN LET d$="7"
4810 IF n=267 THEN LET d$="8"
4820 IF n=268 THEN LET d$="9"
4830 IF n=269 THEN LET d$="0"
4840 IF n=270 THEN LET d$="1"
4850 IF n=271 THEN LET d$="2"
4860 IF n=272 THEN LET d$="3"
4870 IF n=273 THEN LET d$="4"
4880 IF n=274 THEN LET d$="5"
4890 IF n=275 THEN LET d$="6"
4900 IF n=276 THEN LET d$="7"
4910 IF n=277 THEN LET d$="8"
4920 IF n=278 THEN LET d$="9"
4930 IF n=279 THEN LET d$="0"
4940 IF n=280 THEN LET d$="1"
4950 IF n=281 THEN LET d$="2"
4960 IF n=282 THEN LET d$="3"
4970 IF n=283 THEN LET d$="4"
4980 IF n=284 THEN LET d$="5"
4990 IF n=285 THEN LET d$="6"
5000 IF n=286 THEN LET d$="7"
5010 IF n=287 THEN LET d$="8"
5020 IF n=288 THEN LET d$="9"
5030 IF n=289 THEN LET d$="0"
5040 IF n=290 THEN LET d$="1"
5050 IF n=291 THEN LET d$="2"
5060 IF n=292 THEN LET d$="3"
5070 IF n=293 THEN LET d$="4"
5080 IF n=294 THEN LET d$="5"
5090 IF n=295 THEN LET d$="6"
5100 IF n=296 THEN LET d$="7"
5110 IF n=297 THEN LET d$="8"
5120 IF n=298 THEN LET d$="9"
5130 IF n=299 THEN LET d$="0"
5140 IF n=300 THEN LET d$="1"
5150 IF n=301 THEN LET d$="2"
5160 IF n=302 THEN LET d$="3"
5170 IF n=303 THEN LET d$="4"
5180 IF n=304 THEN LET d$="5"
5190 IF n=305 THEN LET d$="6"
5200 IF n=306 THEN LET d$="7"
5210 IF n=307 THEN LET d$="8"
5220 IF n=308 THEN LET d$="9"
5230 IF n=309 THEN LET d$="0"
5240 IF n=310 THEN LET d$="1"
5250 IF n=311 THEN LET d$="2"
5260 IF n=312 THEN LET d$="3"
5270 IF n=313 THEN LET d$="4"
5280 IF n=314 THEN LET d$="5"
5290 IF n=315 THEN LET d$="6"
5300 IF n=316 THEN LET d$="7"
5310 IF n=317 THEN LET d$="8"
5320 IF n=318 THEN LET d$="9"
5330 IF n=319 THEN LET d$="0"
5340 IF n=320 THEN LET d$="1"
5350 IF n=321 THEN LET d$="2"
5360 IF n=322 THEN LET d$="3"
5370 IF n=323 THEN LET d$="4"
5380 IF n=324 THEN LET d$="5"
5390 IF n=325 THEN LET d$="6"
5400 IF n=326 THEN LET d$="7"
5410 IF n=327 THEN LET d$="8"
5420 IF n=328 THEN LET d$="9"
5430 IF n=329 THEN LET d$="0"
5440 IF n=330 THEN LET d$="1"
5450 IF n=331 THEN LET d$="2"
5460 IF n=332 THEN LET d$="3"
5470 IF n=333 THEN LET d$="4"
5480 IF n=334 THEN LET d$="5"
5490 IF n=335 THEN LET d$="6"
5500 IF n=336 THEN LET d$="7"
5510 IF n=337 THEN LET d$="8"
5520 IF n=338 THEN LET d$="9"
5530 IF n=339 THEN LET d$="0"
5540 IF n=340 THEN LET d$="1"
5550 IF n=341 THEN LET d$="2"
5560 IF n=342 THEN LET d$="3"
5570 IF n=343 THEN LET d$="4"
5580 IF n=344 THEN LET d$="5"
5590 IF n=345 THEN LET d$="6"
5600 IF n=346 THEN LET d$="7"
5610 IF n=347 THEN LET d$="8"
5620 IF n=348 THEN LET d$="9"
5630 IF n=349 THEN LET d$="0"
5640 IF n=350 THEN LET d$="1"
5650 IF n=351 THEN LET d$="2"
5660 IF n=352 THEN LET d$="3"
5670 IF n=353 THEN LET d$="4"
5680 IF n=354 THEN LET d$="5"
5690 IF n=355 THEN LET d$="6"
5700 IF n=356 THEN LET d$="7"
5710 IF n=357 THEN LET d$="8"
5720 IF n=358 THEN LET d$="9"
5730 IF n=359 THEN LET d$="0"
5740 IF n=360 THEN LET d$="1"
5750 IF n=361 THEN LET d$="2"
5760 IF n=362 THEN LET d$="3"
5770 IF n=363 THEN LET d$="4"
5780 IF n=364 THEN LET d$="5"
5790 IF n=365 THEN LET d$="6"
5800 IF n=366 THEN LET d$="7"
5810 IF n=367 THEN LET d$="8"
5820 IF n=368 THEN LET d$="9"
5830 IF n=369 THEN LET d$="0"
5840 IF n=370 THEN LET d$="1"
5850 IF n=371 THEN LET d$="2"
5860 IF n=372 THEN LET d$="3"
5870 IF n=373 THEN LET d$="4"
5880 IF n=374 THEN LET d$="5"
5890 IF n=375 THEN LET d$="6"
5900 IF n=376 THEN LET d$="7"
5910 IF n=377 THEN LET d$="8"
5920 IF n=378 THEN LET d$="9"
5930 IF n=379 THEN LET d$="0"
5940 IF n=380 THEN LET d$="1"
5950 IF n=381 THEN LET d$="2"
5960 IF n=382 THEN LET d$="3"
5970 IF n=383 THEN LET d$="4"
5980 IF n=384 THEN LET d$="5"
5990 IF n=385 THEN LET d$="6"
6000 IF n=386 THEN LET d$="7"
6010 IF n=387 THEN LET d$="8"
6020 IF n=388 THEN LET d$="9"
6030 IF n=389 THEN LET d$="0"
6040 IF n=390 THEN LET d$="1"
6050 IF n=391 THEN LET d$="2"
6060 IF n=392 THEN LET d$="3"
6070 IF n=393 THEN LET d$="4"
6080 IF n=394 THEN LET d$="5"
6090 IF n=395 THEN LET d$="6"
6100 IF n=396 THEN LET d$="7"
6110 IF n=397 THEN LET d$="8"
6120 IF n=398 THEN LET d$="9"
6130 IF n=399 THEN LET d$="0"
6140 IF n=400 THEN LET d$="1"
6150 IF n=401 THEN LET d$="2"
6160 IF n=402 THEN LET d$="3"
6170 IF n=403 THEN LET d$="4"
6180 IF n=404 THEN LET d$="5"
6190 IF n=405 THEN LET d$="6"
6200 IF n=406 THEN LET d$="7"
6210 IF n=407 THEN LET d$="8"
6220 IF n=408 THEN LET d$="9"
6230 IF n=409 THEN LET d$="0"
6240 IF n=410 THEN LET d$="1"
6250 IF n=411 THEN LET d$="2"
6260 IF n=412 THEN LET d$="3"
6270 IF n=413 THEN LET d$="4"
6280 IF n=414 THEN LET d$="5"
6290 IF n=415 THEN LET d$="6"
6300 IF n=416 THEN LET d$="7"
6310 IF n=417 THEN LET d$="8"
6320 IF n=418 THEN LET d$="9"
6330 IF n=419 THEN LET d$="0"
6340 IF n=420 THEN LET d$="1"
6350 IF n=421 THEN LET d$="2"
6360 IF n=422 THEN LET d$="3"
6370 IF n=423 THEN LET d$="4"
6380 IF n=424 THEN LET d$="5"
6390 IF n=425 THEN LET d$="6"
6400 IF n=426 THEN LET d$="7"
6410 IF n=427 THEN LET d$="8"
6420 IF n=428 THEN LET d$="9"
6430 IF n=429 THEN LET d$="0"
6440 IF n=430 THEN LET d$="1"
6450 IF n=431 THEN LET d$="2"
6460 IF n=432 THEN LET d$="3"
6470 IF n=433 THEN LET d$="4"
6480 IF n=434 THEN LET d$="5"
6490 IF n=435 THEN LET d$="6"
6500 IF n=436 THEN LET d$="7"
6510 IF n=437 THEN LET d$="8"
6520 IF n=438 THEN LET d$="9"
6530 IF n=439 THEN LET d$="0"
6540 IF n=440 THEN LET d$="1"
6550 IF n=441 THEN LET d$="2"
6560 IF n=442 THEN LET d$="3"
6570 IF n=443 THEN LET d$="4"
6580 IF n=444 THEN LET d$="5"
6590 IF n=445 THEN LET d$="6"
6600 IF n=446 THEN LET d$="7"
6610 IF n=447 THEN LET d$="8"
6620 IF n=448 THEN LET d$="9"
6630 IF n=449 THEN LET d$="0"
6640 IF n=450 THEN LET d$="1"
6650 IF n=451 THEN LET d$="2"
6660 IF n=452 THEN LET d$="3"
6670 IF n=453 THEN LET d$="4"
6680 IF n=454 THEN LET d$="5"
6690 IF n=455 THEN LET d$="6"
6700 IF n=456 THEN LET d$="7"
6710 IF n=457 THEN LET d$="8"
6720 IF n=458 THEN LET d$="9"
6730 IF n=459 THEN LET d$="0"
6740 IF n=460 THEN LET d$="1"
6750 IF n=461 THEN LET d$="2"
6760 IF n=462 THEN LET d$="3"
6770 IF n=463 THEN LET d$="4"
6780 IF n=464 THEN LET d$="5"
6790 IF n=465 THEN LET d$="6"
6800 IF n=466 THEN LET d$="7"
6810 IF n=467 THEN LET d$="8"
6820 IF n=468 THEN LET d$="9"
6830 IF n=469 THEN LET d$="0"
6840 IF n=470 THEN LET d$="1"
6850 IF n=471 THEN LET d$="2"
6860 IF n=472 THEN LET d$="3"
6870 IF n=473 THEN LET d$="4"
6880 IF n=474 THEN LET d$="5"
6890 IF n=475 THEN LET d$="6"
6900 IF n=476 THEN LET d$="7"
6910 IF n=477 THEN LET d$="8"
6920 IF n=478 THEN LET d$="9"
6930 IF n=479 THEN LET d$="0"
6940 IF n=480 THEN LET d$="1"
6950 IF n=481 THEN LET d$="2"
6960 IF n=482 THEN LET d$="3"
6970 IF n=483 THEN LET d$="4"
6980 IF n=484 THEN LET d$="5"
6990 IF n=485 THEN LET d$="6"
7000 IF n=486 THEN LET d$="7"
7010 IF n=487 THEN LET d$="8"
7020 IF n=488 THEN LET d$="9"
7030 IF n=489 THEN LET d$="0"
7040 IF n=490 THEN LET d$="1"
7050 IF n=491 THEN LET d$="2"
7060 IF n=492 THEN LET d$="3"
7070 IF n=493 THEN LET d$="4"
7080 IF n=494 THEN LET d$="5"
7090 IF n=495 THEN LET d$="6"
7100 IF n=496 THEN LET d$="7"
7110 IF n=497 THEN LET d$="8"
7120 IF n=498 THEN LET d$="9"
7130 IF n=499 THEN LET d$="0"
7140 IF n=500 THEN LET d$="1"
7150 IF n=501 THEN LET d$="2"
7160 IF n=502 THEN LET d$="3"
7170 IF n=503 THEN LET d$="4"
7180 IF n
```

Aspecto de la memoria del Spectrum compuesta por 16 chips de RAM dinámica y una de ROM.

SENTENCIA	ARGUMENTO
POKE	Dirección, Dato

Ejemplos:

- POKE 23300,255
- POKE 65342,a
- POKE d+38, -37
- POKE h*2, a-25

La dirección ha de estar comprendida entre 0 y 65535 (si bien, intentar "POKEar" en una dirección menor de 16384, no tiene sentido) y el dato ha de estar entre -255 y 255 (un dato negativo equivale a 256 menos ese número). Si alguno de estos números estuviera fuera de este margen, se produciría el error:

B Integer out of range

La razón de que no tenga sentido "POKEar" en direcciones inferiores a 16384 es que, como hemos dicho an-

tes, el contenido de estas direcciones no se puede alterar.

PEEK

Acceso al teclado

PEEK



Definición

La función PEEK devuelve el contenido de una posición de memoria, cuya dirección es el argumento de PEEK. Su estructura general es:

SENTENCIA	ARGUMENTO
PEEK	Dirección

B Integer out of range

La razón de que no tenga sentido "POKEar" en direcciones inferiores a 16384 es que, como hemos dicho an-

Apertura de ficheros

Utilizando la sentencia "OPEN # " podemos asignar un fichero Microdrive a una corriente; si el fichero no existe, el sistema operativo interpreta que es de *escritura*; por el contrario, si existe, es de *lectura*.

Por ejemplo:

OPEN # 5; "m"; 1; "basic"

Grabación de datos

Podremos grabar datos en un fichero siempre y cuando haya sido abierto para escritura.

Para grabar datos en un fichero se utiliza la sentencia PRINT.

SENTENCIA	ARGUMENTO
PRINT	# corriente; datos

Ejemplos:

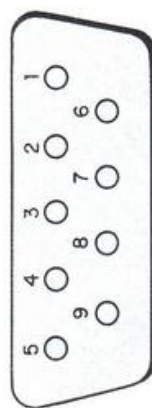
- PRINT # 5; 20
 - PRINT # 5; 3'7'8
 - PRINT # 5; a\$
 - PRINT # 5; 89'b\$'n\$
- Para una posterior lectura, los datos deben ir seguidos del código "ENTER"; para ello introducimos los datos uno a uno.

PRINT # 5; 30
PRINT # 5; 15
PRINT # 5; 7

o utilice el signo del apóstrofe para separarlos

PRINT # 5; 30'15'7

Los datos no se graban directamente en el cartucho, si-



CONECTOR "HEMBRA" TIPO "D" 9 TERMINALES

TERMINAL	NOMBRE
1	NC (NO CONECTADO)
2	TD (TRANSMIT DATA)
3	RD (RECEIVE DATA)
4	DTR (DATA TERMINAL READY)
5	CTS (CLEAR TO SEND)
6	NC
7	MASA
8	NC
9	+9V

Conexión "RS 232".

no que se almacenan en una memoria intermedia de 512 bytes (0.5 Kbytes), cuando ésta se llena, se realiza la transferencia al Microdrive. La grabación se realiza por bloques.

Cierre de ficheros

Cuando la introducción de datos se termina, el fichero debe cerrarse, ya que de lo contrario no podría efectuarse una lectura posterior.

Al cerrar un fichero se transfieren al Microdrive los datos que estuvieran almacenados en la memoria intermedia.

La forma de cerrar un fichero se realiza utilizando la sentencia "CLOSE #".

Ejemplo:

```
CLOSE # 5
```

Lectura de ficheros

Para leer datos de un fichero es necesario que esté abierto para lectura.

Los datos se leen utilizando la sentencia "INPUT".

SENTENCIA	ARGUMENTO
INPUT	# corriente; variables

```
- INPUT # 5; a
- INPUT # 5; b$
- INPUT # 5; c; n$; k
- INPUT # 5; LINE T$
```

Si se leen varios datos separados, estos deben ir separados por "punto y coma".

```
INPUT # 5; d; e; n; k$
```

En la lectura de datos de cadena es conveniente utilizar el siguiente formato:

```
INPUT # 5; LINE c$
```

ya que si la cadena contiene comillas, interpretará que son el final de la cadena.

Ampliación de ficheros

Para realizar una ampliación de un fichero de datos ya creado, deberá seguir los siguientes pasos:

- Abrir el fichero antiguo para lectura.

CANALES DE COMUNICACION

```
"k" = Teclado
"s" = Pantalla
"p" = Impresora
"m" = Microdrive
"n" = Red de area local
"t" = RS232 Texto
"b" = RS232 Binario
```

CORRIENTES

```
#0 Salida y entrada de datos de la parte inferior de la pantalla.
#1
#2 Salida de datos de la parte superior de la pantalla.
#3 Salida de datos a través de la impresora.
#4-15 Libres para ser definidos por el usuario.
```

- Abrir un nuevo fichero para escritura.

- Copiar el fichero antiguo en el nuevo.

- Editar a continuación los datos suplementarios en el fichero nuevo.

- Cerrar los dos ficheros.

Ejemplo:

Tenemos un fichero llamado "DATOS" con 20 datos y deseamos incluir cinco más:

```
10, 84, 32, 15 y 49
```

a) Abrir fichero antiguo

```
OPEN # 4; "m"; 1; "DATOS"
```

b) Abrir fichero nuevo

```
OPEN # 5; "m"; 1; "DATOS.1"
```

c) Copiar fichero

```
FOR n = 1 TO 10:0
INPUT # 4; dato
PRINT # 5; dato
NEXT n
```

LA MEMORIA

Todo lo que introducimos en el ordenador, tiene que almacenarse dentro de alguna forma. Para ello todos los ordenadores disponen de lo que se denomina *memoria*.

Podemos imaginar la memoria de nuestro ordenador como un conjunto de 65536 cajitas, cada una de ellas puede contener un número comprendido entre 0 y 255. Cada cajita tiene, a su vez, un número que actúa como "nombre propio" y la distingue de las demás; la primera casilla se llama "0" y la última "65535". Este número se de-

nomina "Dirección" y el número que contienen se denomina "Dato".

El dato contenido en las primeras 16384 cajitas es fijo, no podemos alterarlo; es como si estas cajitas estuvieran cubiertas por un cristal, podemos ver el número que hay en su interior pero no podemos cambiarlo por otro.

En las restantes cajitas, si podemos alterar el contenido. Veamos ahora las sentencias de Basic que trabajan directamente sobre la memoria.

POKE

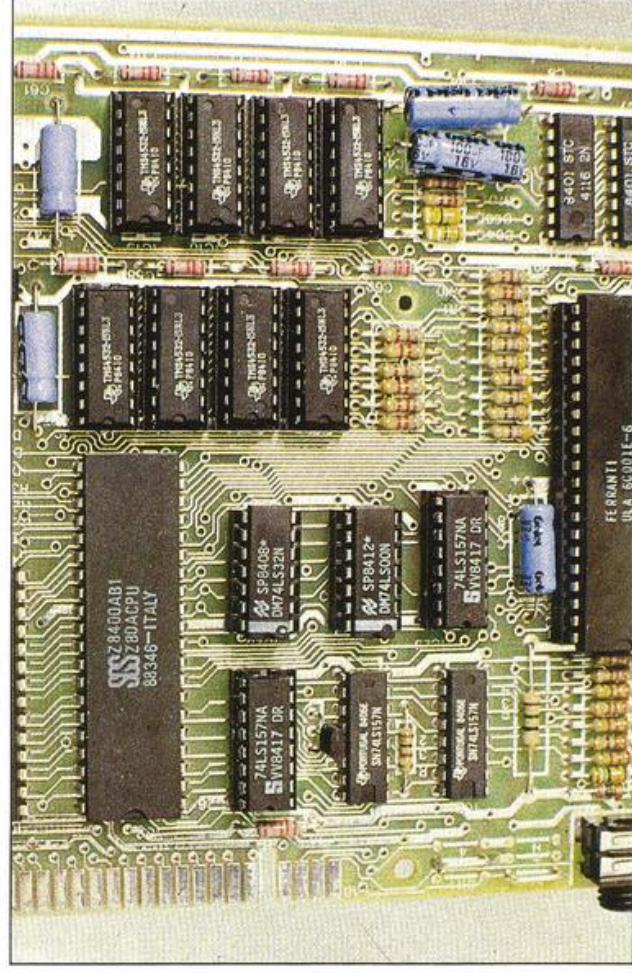
Definición

El comando POKE almacena un dato en una dirección de memoria.

Su estructura general es:

Acceso al teclado

PEEK



Los "chips" que componen la memoria del Spectrum.

velocidad de 9600 baudios,
seria:

```
10 FORMAT "t"; 9600
20 OPEN # 6: "t"
30 PRINT INKEY$ # 6;
40 GOTO 30
```

El canal "b" se utiliza para
enviar los códigos de control
a las impresoras, para utilizar
un *modem*, etc.

Con el canal «b» se pueden
utilizar las sentencias "SAVE"
y "LOAD".

```
SAVE "b"
LOAD "b"
```

dependiendo de si es trans-
misor o receptor.



El Spectrum, con el nuevo dispositivo (interface-1) y dos
unidades de microdrive, puede acceder a 180 K de
memoria externa.

Para enviar un código de
control a una impresora, por
ejemplo el 18, se utilizaría:

```
OPEN # 7: "b"
PRINT # 7: CHR$ 18
CLOSE # 7
```

Catálogo de un cartucho "microdrive".

d) Editar nuevos datos

```
PRINT # 5: 10:84:32
PRINT # 5: 15:49
```

e) Cerrar ficheros

```
CLOSE # 4
CLOSE # 5
```

f) Borrar fichero antiguo

```
ERASE "m": 1: "DATOS"
```

Programa

El programa número "1"
permite generar un fichero Bi-
bliográfico de hasta 100 títu-
los de libros con sus corres-
pondientes autores.

Tanto los nombres de los li-
bros como sus autores debe-
rán tener una longitud máxi-
ma de 20 caracteres.

Para salir de la opción "1"
deberá teclearse la palabra

```
FICHERO
FECHA
```

LIBROS

21/06/85

```
1 LA REGENTA
  CLARIN
2 1984
  ORWELL
3 EL PADRINO
  MARIO PUZO
4 RIMAS Y LEYENDAS
  BECQUER
5 MARGARET
  SHAKESPEARE
```

Programa "Fichero".

nerados.

La opción "3" permite leer
un fichero generado anterior-
mente para su ampliación o
visualización.

Deberemos proporcionar
el nombre sin *extensión* y la
clave.

Si la clave almacenada en
el fichero con *extensión*
"CLV" no coincide con la te-
cleada, el programa se auto-
destruye, no permitiendo el
acceso a personas que no
conozcan la clave.

Por último la opción "4"
permite visualizar un fichero,
bien sea en pantalla o impre-
sora.

MOVE

Acceso al teclado

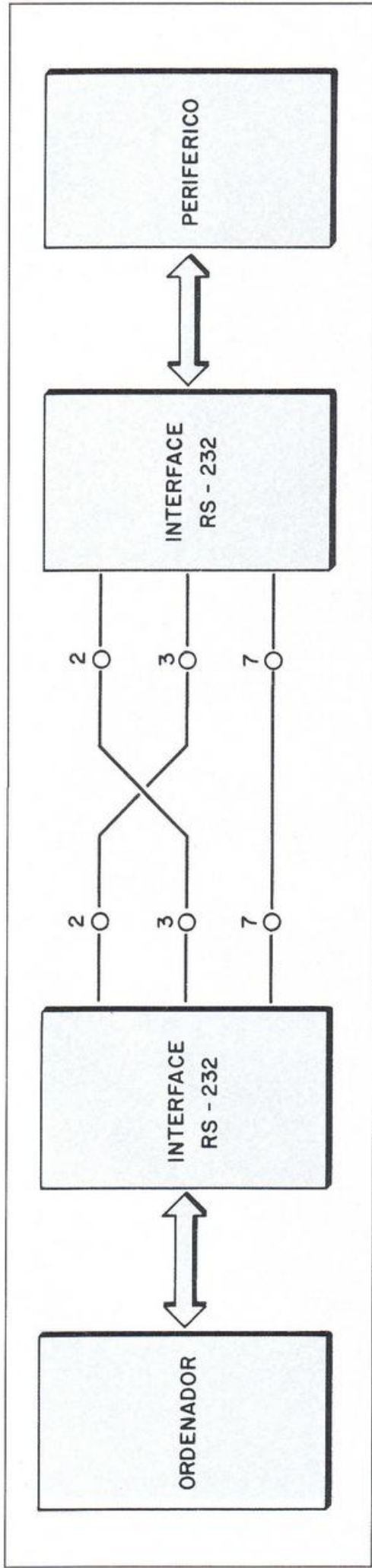
YELLOW

MODO E



MOVE

SYMBOL SHIFT



Conexión simple entre interfaces RS 232.

Definición

La sentencia "MOVE" transfiere los datos de un canal de comunicación (fuente) a otro distinto (destino).
La estructura general es:

SENTENCIA	ARGUMENTO
MOVE	fuente TO destino

Ejemplos:

- MOVE #1 TO #2
- MOVE "m"; 1; "as" TO #3
- MOVE "m"; 1; "Tn" TO "m"; 2; "Tn"
- MOVE #1 TO #3

La "fuente" y el "destino" pueden ser canales o números de corriente.

Esta sentencia es muy útil ya que se puede realizar:

- Copias de ficheros de datos

MOVE "m"; 1; "CAN" TO "m"; 1; "CAN1"
MOVE "m"; 1; "SET" TO "m"; 2; "SET"

- Visualización de ficheros

manera que todas las demás puedan hacer uso de ellos.

Una vez establecida la red, con sus correspondientes cables, es necesario identificar cada estación, para ello utilice la sentencia "FORMAT".

SENTENCIA	ARGUMENTO
FORMAT	"n"; estación

Ejemplos:

- FORMAT "n"; 2
- FORMAT "n"; VAL N\$
- FORMAT "n"; h
- FORMAT "n"; a * 5

La cadena "n" es el indicativo del canal correspondiente a la red, y "estación" en un número entero entre 1 y 64.

Para enviar o recibir programas de la red se utilizan las sentencias "SAVE", "VERIFY", "LOAD" y "MERGE" con las siguientes variaciones:

SAVE "n"; estación destino
VERIFY "n"; estación fuente
LOAD "n"; estación fuente
MERGE "n"; estación fuente

Como podrá observar, la red no utiliza nombres de programas.

Cuando se desea establecer un diálogo entre una estación y todas las demás, simultáneamente, se utiliza el canal de difusión.

Las estaciones receptoras deben teclear:

LOAD "n"; 0

y la transmisora

SAVE "n"; 0

Interface RS-232

Con el uso del Interface RS-232, el Spectrum puede dialogar con cualquier periférico que utilice este protocolo de comunicación serie (Impresoras, modem, otro ordenador, etc.).

Es necesario realizar algunos ajustes en el periférico antes de utilizarlo.

- Desactivar el avance de línea automático.

Desactivar el control de paridad.

- Seleccionar ocho bits.
- Seleccionar un solo bit de parada.
- Seleccionar la velocidad de transmisión, expresada en baudios (bits por segundo), entre cualquiera de las mostradas a continuación.

50	2400
110	4800
300	9600
600	19200
1200	

El Interface RS-232 utiliza dos canales de comunicación:

- El "t" para enviar textos en ASCII.
 - El "b" para enviar códigos de ocho bits en binario.
- Para utilizar cualquiera de los dos canales primero debe indicar al Spectrum la velocidad de transmisión a la que ha ajustado el periférico.

Ejemplos:

FORMAT "t"; 2400
FORMAT "b"; 600

El canal "t" envía el código ASCII de acuerdo al siguiente formato:

- Los códigos del 0 al 31 no se envían, excepto el código 13 (ENTER) que se envía seguido del 10 (cursor abajo) es decir, lo que se conoce por "retorno de carro" y "avance de línea".
- Los códigos del 32 al 127 ("espacio" al "copyright") se envían sin ninguna alteración.
- Los códigos 128 al 164 (gráficos) no se envían, en su lugar se envía el código 63 (?).
- Los códigos del 165 al 255 (tokens) se expanden a los códigos 32-127.

Para enviar un listado a una impresora con Interface RS-232 y velocidad de transmisión de 300 baudios, se utilizaría:

FORMAT "t"; 300
OPEN #5; "t"
LIST #5

Para recibir datos desde un terminal transmisor con una

RENÚMERADOR - LISTADO ASSEMBLER		
Direcc.	Cód. Máq.	Listado Assembler
23296	42,83,92	LD HL, (PROG)
23299	17,10,0	LD DE, +0A
23302	229	BUCLE PUSH HL
23303	237,75,75,92	LD BC, (VARS)
23307	237,56	SBC HL, BC
23309	40,18	JR Z, (FINAL)
23311	225	POP HL
23312	114	LD (HL), D
23313	35	INC HL
23314	115	LD (HL), E
23315	35	INC HL
23316	79	LD C, (HL)
23317	35	INC HL
23318	70	LD B, (HL)
23319	9	ADD HL, BC
23320	35	INC HL
23321	235	EXX DE, HL
23322	1,10,0	LD BC, +0A
23325	9	ADD HL, BC
23326	235	EXX DE, HL
23327	24,229	JR (BUCLE)
23329	207	FINAL RST 08
23330	255	DEFB +FF
		PROG EQU +5C53
		VARS EQU +5C4B

Figura 5. Un sencillo renumerador en Código Máquina para el Spectrum.

mensaje.

4 Out of memory. 0:1

Hemos intentado dimensionar una matriz de 10 por 10, lo que requiere 508 bytes disponibles en el área de variables, pero tenemos tan baja la RAMTOP que no hay espacio suficiente.

Este mensaje se presentará cada vez que intente hacer algo para lo que necesite más memoria que la que tiene disponible.

Siempre que utilice CLEAR en un programa, recuerde que además de bajar la RAMTOP, le borrará la pantalla y todas las variables que hubie-

ra definido hasta ese momento, además de restaurarle las posiciones de PRINT y PLOT y el puntero de DATA, de la misma forma que si hubiera ejecutado un RESTORE y un CLS.

Puede utilizar CLEAR sin argumento, que hará todo excepto modificar la RAMTOP.

tipos de memoria, ROM, RAM, PROM, EPROM, EAROM, etc. Nuestro ordenador dispone de 16384 (16 K) direcciones de memoria ROM y 49152 (48 K) de memoria RAM (en la versión de 16 K, son sólo 16384 posiciones de RAM).

ROM significa "Read Only Memory" (Memoria de sólo lectura), como su nombre indica, es una memoria que sólo se puede leer, y en la que no se puede escribir, su contenido ha sido grabado en fábrica de forma indeleble.

RAM significa «Random Access Memory» (Memoria de acceso aleatorio), en realidad, la ROM también es de acceso aleatorio, ya que podemos leer cualquiera de sus datos sin leer los precedentes, pero a la RAM se la llama así para distinguirla de la ROM, ya que en memoria RAM podemos tanto leer como escribir.

El mapa básico de memoria para ambas versiones sería el representado en la FIGURA 1.

Bit y byte

Hasta ahora hemos dicho que los números se almacenan en las posiciones de memoria, pero no hemos explicado la forma en que se almacena un número determinado.

Se puede imaginar que cada posición de memoria (celda) es una fila de ocho interruptores, a un interruptor encendido le llamamos "1" y a uno apagado le llamamos "0". Cada número comprendido entre "0" y "255" se forma por combinación de "unos" y "ceros", es lo que se

128	64	32	16	8	4	2	1	
1	0	0	1	1	0	1	0	= 128+16+8+2 = 154
0	1	1	0	0	1	0	1	= 64+32+4+1 = 101
1	1	1	1	1	1	1	1	= 128+64+32+16+8+4+2+1 = 255
0	0	0	0	1	1	1	1	= 8+4+2+1 = 15

llama "Notación Binaria". A cada "1" o "0" se le denomina "Bit" (abreviatura de "Binary Digit"), y al conjunto de ocho "unos" o "ceros" se le denomina "Byte", por tanto, cada

Memoria ROM

El motivo de que la memoria ROM venga grabada de fábrica y sus datos no puedan ser

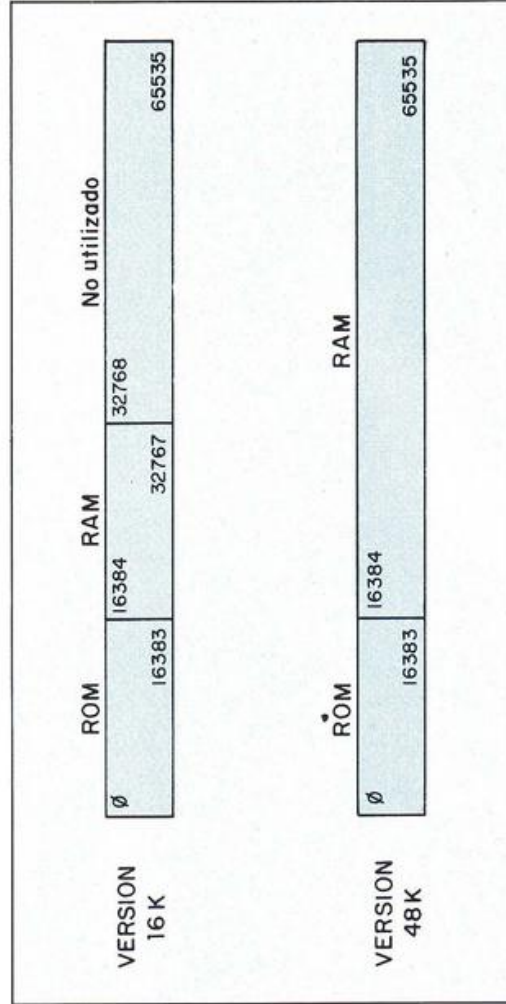


Figura 1. Mapa de memoria básico de ambas versiones del Spectrum.

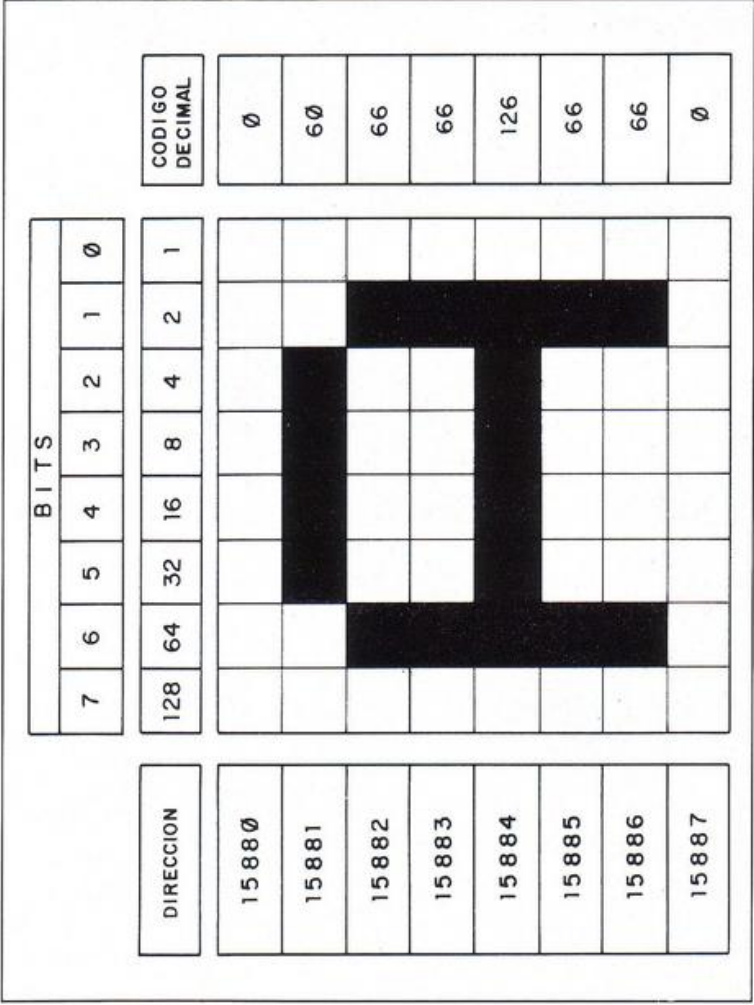


Figura 3. La letra "A" tal como está definida en la ROM del Spectrum

alterados, es que contiene lo que se denomina el "SISTEMA OPERATIVO" del ordenador, un conjunto de programas escritos en código máquina que permiten al ordenador operar desde el mismo momento en que se conecta.

También contiene la ROM el juego de caracteres, se encuentran a partir de la dirección 15616 y están definidos de la misma forma que los UDG. Como ejemplo, la letra "A" está definida de la forma que puede verse en la FIGURA 3.

Parte del Sistema Operativo del ordenador, lo constituye el denominado "INTERPRETE DE BASIC" que es precisamente, el programa que nos

permite utilizar este lenguaje para programar el ordenador.

La memoria RAM

En las direcciones que están a continuación de la ROM, se encuentra la memoria RAM. No toda ella está a nuestra disposición, ya que una parte la necesita el ordenador para sus propios datos.

Los primeros 6144 bytes (desde la dirección 16384 hasta la 22527) están ocupados por el archivo de presentación visual, en esta zona se encuentra almacenado todo lo que vemos en la pantalla. Los 768 bytes siguientes

(desde la dirección 22528 hasta la 23295) contienen el archivo de atributos, que almacena los colores de todos los caracteres de la pantalla. Entre la dirección 23296 y la 23551 (256 bytes) se encuentra la memoria intermedia de impresora. A continuación, entre la 23552 y la 23733 (182 bytes) están las variables del sistema.

A partir de la dirección 23734 se encuentra el área de información para canales; en la versión básica, esta zona ocupa 21 bytes, pero se expande al conectar el INTERFACE 1 y trabajar con MICRODIVE o con la ZX-NET.

A continuación, viene la zona donde almacenamos el

TOP. Es la dirección más alta que puede utilizar el Basic, algo así como un "límite" del sistema. Podemos variar la posición de este límite, dentro de unos márgenes, lo cual puede ser muy útil en determinados casos. Normalmente, por encima de RAMTOP sólo se encuentran los Gráficos Definidos por el Usuario, pero podemos bajar la RAMTOP y hacer sitio para colocar algo que no deseamos que el Basic pueda borrar, por ejemplo, un programa en código máquina. Lo que coloquemos por encima de RAMTOP queda a salvo de borrados incluso con NEW. Si desea saber en qué dirección se encuentra la RAMTOP de su ordenador, teclee: **PRINT PEEK 23730+256*PEEK 23731**

La respuesta normal será: 32599 para el modelo de 16 K, y 65367 para el de 48 K. Tenga en cuenta que algunos interfaces pueden alterar este valor.

Si desea bajar la RAMTOP teclee el comando CLEAR seguido de la nueva dirección. Por ejemplo, supongamos que su versión es de 16 K, en ese caso la RAMTOP estará en 32599. Supongamos ahora que quiere conseguir un espacio libre de 300 bytes, teclee:

CLEAR 32299

Ahora las direcciones libres van desde la 32300 hasta la 32599, ambas inclusive.

La dirección más baja que puede usar como argumento de CLEAR es 23821, y la más

alta es 32767 en la versión de 16 K y 65535 en la de 48 K. Si el argumento de CLEAR está fuera de este margen, obtendrá el informe:

M RAMTOP no good.

En este caso, CLEAR habrá hecho todo (borrar variables, restaurar punteros, etc.) menos cambiar la RAMTOP.

Cuando se llena la memoria

Es posible que haya tenido ya alguna experiencia de lo que ocurre cuando se llena la memoria de su ordenador, si jugamos un poco con CLEAR bajando mucho la RAMTOP, seremos capaces de verlo claramente. Si tiene el INTERFACE 1, desconéctelo para hacer estas pruebas. Teclee:

CLEAR 23850

Ahora intente pulsar cualquier tecla, verá que el ordenador no responde, y en su lugar, emite un "pitido" que dura un par de segundos. Hemos bajado tanto la RAMTOP que no le hemos dejado sitio al Basic para trabajar. No tenemos más remedio que desconectar el ordenador y volverlo a conectar de nuevo para que todo vuelva a la normalidad.

El "pitido" que hemos oído es la señal de alarma del Spectrum, indica que la memoria está totalmente llena. Ahora teclee:

CLEAR 23900

Ahora introduzca las siguientes líneas:

10 REM *****
20 REM *****
30 REM *****

No se asuste, si cuando ha ido a introducir la línea 30 el ordenador no la ha aceptado, y le ha respondido con el mensaje:

G No room for line, 0:1

Esto le indica que el área de Basic está llena, y no cabe ninguna línea más. Ahora haga NEW y teclee:

10 REM *****
20 REM *****
30 REM *****

Esta vez, no sólo no ha podido introducir la línea 20, sino que ni siquiera ha podido terminarla, cuando llevaba 22 "x" pulsadas e iba a pulsar la 23, el ordenador se ha bloqueado de nuevo con el "pitido", la memoria está tan llena que no hay sitio suficiente en el área de edición para construir la nueva línea. Borre toda la línea 20 pulsando DELETE, e introduzca:

CLEAR 24000: NEW

Ahora teclee:

DIM a(10:10)

Al pulsar ENTER, el ordenador le responderá con el

línea	10	00001010
= 10	0	00000000
sent.	1	00000001
= 1		

Matriz de números

Los dos bytes siguientes al nombre, almacenan la longitud de la variable más sus dimensiones, a partir de ahí y de dos en dos bytes, se almacenan las dimensiones y a continuación, los elementos en coma flotante, ordenados según los subíndices.

Ejemplo:

```
DIM A(2,2): LET a (1,1)=15:
LET a(1,2)=26:
LET a(2,1)=12: LET a(2,2)=7
```

a	129	100000001
long.	25	00011001
= 25	0	00000000
dim. = 2	2	00000010
1 dim.	2	00000010
2 dim.	0	00000000
a(1,1)	0	00000000
= 15	0	00000000
	15	00001111
	0	00000000
a(1,2)	0	00000000
= 26	0	00000000
	26	00011010
	0	00000000
a(2,1)	0	00000000
= 12	0	00000000
	12	00001100
	0	00000000
a(2,2)	0	00000000
= 7	0	00000000
	7	00000111
	0	00000000
	0	00000000

Matriz de caracteres

Los dos primeros bytes que siguen al nombre, indican la longitud de la variable más sus dimensiones, el siguiente indica el número de dimensiones, a continuación vienen las dimensiones, y finalmente, el texto.

Ejemplo:

```
DIM a$(2,8): LET a$(1)="BASIC";
LET a$(2)="SINCLAIR"
```

a	193	110000001
long.	21	00010101
= 21	0	00000000
dim. = 2	2	00000010
1 dim.	2	00000010
2 dim.	0	00000000
B	66	01000010
A	65	01000001
S	83	01010011
I	73	01001001
C	67	01000011
--	32	00100000
--	32	00100000
S	83	01010011
I	73	01001001
N	78	01001110
C	67	01000011
L	76	01001100
A	65	01000001
I	73	01001001
R	82	01010010

Borrado de variables

Ejecute el siguiente programa:

```
10 LET a=27
20 CLEAR
30 PRINT a
```

Verá que se detiene con el informe:

```
2 Variable not found, 30:1
```

La línea 30 no encuentra la variable "a" a pesar de haber sido definida en la línea 10. Lo que ocurre es que la variable ha sido borrada en la línea 20, ésta es una de las utilidades del comando CLEAR.

CLEAR

Acceso al teclado



Definición

El comando CLEAR borra la pantalla, las variables, restaura la posición de PRINT a la esquina superior izquierda, restaura la posición de PLOT a la esquina inferior izquierda, restaura el puntero de DATA y, caso de tener un argumento numérico, cambia la dirección de RAMTOP si ello fuera posible.

Su estructura general es:

SENTENCIA	ARGUMENTO
CLEAR	Nueva RAMTOP

Vuelva a mirar la figura 2 donde se muestra el mapa de memoria, casual final verá una dirección apuntada por una variable que se llama RAM-

programa Basic y las variables, la longitud de estas zonas no es fija ya que se van expandiendo a medida que vamos almacenando datos en ellas. Las direcciones de inicio de cada una de estas zonas están contenidas en las variables del sistema CHANS, PROG y VARS que se actualizan continuamente por el sistema operativo.

Hay una serie de zonas más cuyas longitudes y direcciones de inicio tampoco son fijas, pero se encuentran anotadas en las correspondientes variables del sistema. Estas zonas son: Área de edición, zona de trabajo y pila del calculador.

A continuación viene una zona llamada de reserva, que contiene los bytes que no han sido utilizados, y por tanto, se contrae cada vez que se expande cualquiera de las otras zonas. Este área es considerablemente mayor en la versión de 48 K que en la de 16 K.

Todas las zonas que hemos visto hasta ahora se expanden hacia "arriba", pero hay dos zonas por encima del área de reserva que se expanden hacia "abajo", se trata de las pilas de máquina y GO SUB.

Finalmente, los últimos 168 bytes de la RAM se encuentran ocupados por los Gráficos Definidos por el Usuario (UDG).

En la FIGURA 2 se puede ver el mapa de memoria del Spectrum, los números situados a la izquierda son direcciones, y los situados a la derecha indican la cantidad de bytes ocupados por la zona correspondiente en el momento de conectar el ordenador.

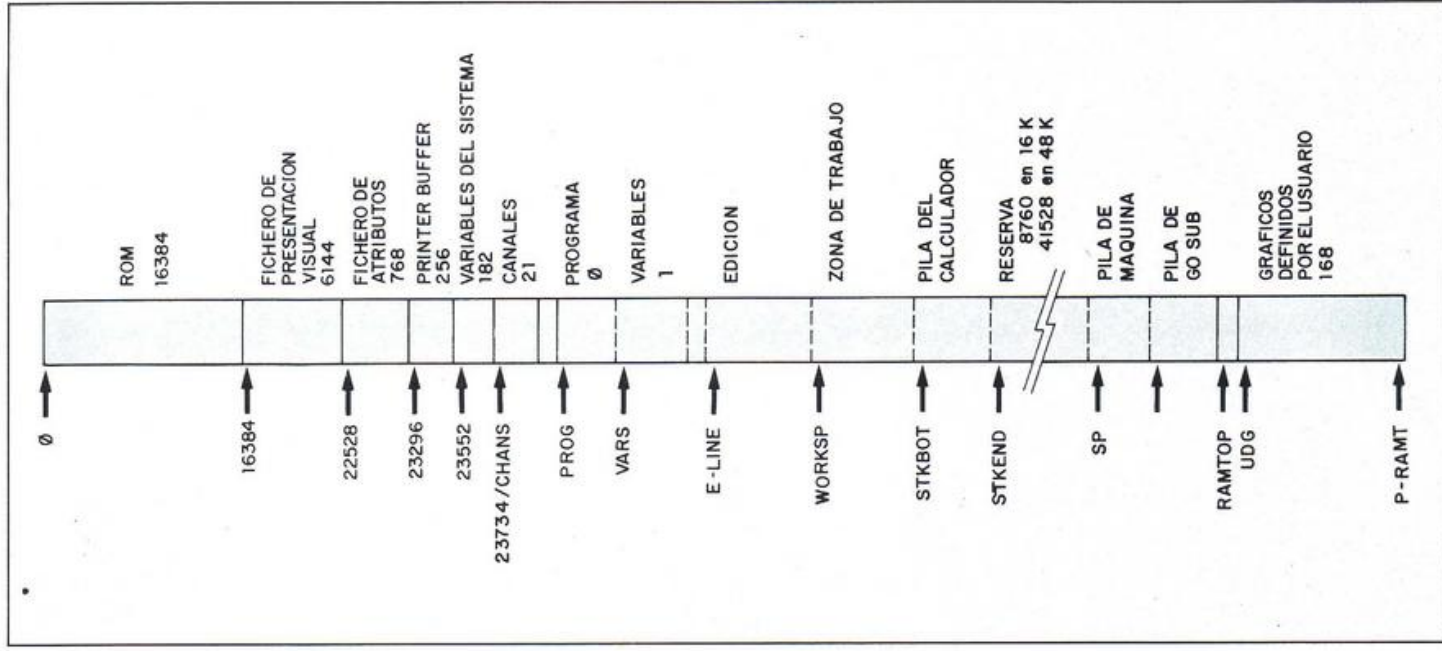


Figura 2. Mapa completo de memoria del Spectrum.

Almacenamiento de programas

Ahora vamos a ver como se almacenan en la memoria las líneas de Basic y las variables que introducimos cuando escribimos o cargamos un programa.

A partir de la dirección apuntada por la variable del sistema PROG, se almacena la primera línea del programa. En primer lugar, el número de línea que ocupa dos bytes (el más significativo primero); a continuación, viene la longitud de la línea que también ocupa dos bytes, pero con el orden invertido.

Los siguientes bytes constituyen el texto de la línea propiamente dicha, donde hay un número, está primero su representación ASCII, a continuación el código 14 y después la representación del número en coma flotante, que es la que realmente utiliza el ordenador.

Supongamos la siguiente línea:

10 LET a=27

En el interior de la memoria esta línea quedaría almacenada como se ve en la FIGURA 4.

Variables

La forma en que se almacenan las variables es algo más compleja, ya que depende del tipo de variable de que se trate.

El tipo de variable viene dado por la configuración de los tres primeros bits del primer byte, de la forma siguiente:

DIRECCION	DATO	
PROG	0	Número de línea
PROG + 1	10	
PROG + 2	12	
PROG + 3	0	Longitud de la línea + ENTER (12 bytes)
PROG + 4	241	
PROG + 5	97	Código de LET
PROG + 6	61	Código de "a"
PROG + 7	50	Código de "="
PROG + 8	55	Representación ASCII de "27"
PROG + 9	14	Indicador de número
PROG + 10	0	
PROG + 11	0	
PROG + 12	27	Representación en coma flotante de "27"
PROG + 13	0	
PROG + 14	0	
PROG + 15	13	Código de ENTER (fin de línea)

Figura 4. Una línea de programa en Basic, tal como se almacena en la zona de programa de la memoria RAM.

Variable de cadena de caracteres

Los dos bytes que siguen al nombre indican la longitud. Ejemplo:

a\$="HOLA"

A	65	0	1	0	0	0	0	0	1
Long	4	0	0	0	0	0	1	0	0
= 4	0	0	0	0	0	0	0	0	0
H	72	0	1	0	0	1	0	0	0
O	79	0	1	0	0	1	1	1	1
L	76	0	1	0	0	1	1	0	0
A	65	0	1	0	0	0	0	0	1

Variable de control de bucle FOR-NEXT

Los cinco primeros bytes que siguen al nombre indican el valor inicial en coma flotante, los cinco siguientes indican el límite, los cinco siguientes el "paso", los dos siguientes indican la línea donde se ha definido el bucle y el último indica el número de sentencia dentro de la línea.

Ejemplo:

10 FOR a=1 TO 7 STEP 2

a	238	1	1	1	0	1	1	1	0
valor	= 1	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		1	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
límite	= 7	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		7	0	0	0	0	0	1	1
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
paso	= 2	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		2	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0

Variable numérica cuyo nombre es una sola letra: 011
Variable numérica cuyo nombre son varias letras: 101
Variable de cadena de caracteres: 010
Variable de control de bucle FOR-NEXT: 111
Matriz de números: 100
Matriz de caracteres: 110

A continuación, vamos a ver cada una detenidamente y con algún ejemplo.

Variable numérica cuyo nombre es una sola letra

Ejemplo:

a=27

a	97	0	1	1	0	0	0	0	1
27	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	27	0	0	0	1	1	0	1	1
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Variable numérica cuyo nombre son varias letras

En este caso, la última letra tiene el primer bit a "1" para indicar que es el fin del nombre.

Ejemplo:

abcd=27

a	161	1	0	1	0	0	0	0	1
b	98	0	1	1	0	0	1	0	0
c	99	0	1	1	0	0	1	1	1
d	228	1	1	1	0	0	1	0	0
27	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	27	0	0	0	1	1	0	1	1
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

VARIABLES DEL SISTEMA

De la misma manera que un programa Basic utiliza una serie de variables, el Sistema Operativo (que de hecho es un programa escrito en código máquina) utiliza las suyas; son lo que se denomina "Variables del Sistema".

Las Variables del Sistema están todas juntas, y ocupan direcciones de memoria fijas. Tienen nombres, pero el ordenador no los reconoce, su única finalidad es servir a efectos nemotécnicos, para recordarnos su función. El verdadero nombre por el que se hace referencia a una variable en concreto es la dirección de la posición de memoria que ocupa. La TABLA 1 es

una lista de todas las variables del sistema ordenadas alfabéticamente, con su dirección en decimal y hexadecimal y el número de bytes que ocupan.

Cuando una variable ocupa más de un byte, el primero contiene el octeto menos significativo y el último, el más significativo, por ejemplo, si el contenido de una variable de dos bytes de longitud fuera "3B4C" (en hexadecimal), el primer byte contendría "4C" y el segundo "3B". Justo al revés de lo que parecería normal, pero éste es el formato que necesita el microprocesador para poder leer los números correctamente.

La mayor parte de las variables ocupan dos bytes. Si desea leer el contenido de una variable cuya dirección es "d", utilice:

```
PRINT PEEK d+256*PEEK (d+1)
```

Y si desea almacenar el número "n" en una variable cuya dirección es "d", utilice:

```
POKE d,n-256*INT (n/256):
POKE d+1,INT (n/256)
```

El PROGRAMA 1 sirve para imprimir el contenido de cualquier variable del sistema, para ello pregunta primero el nombre de la variable, que deberá teclearse tal como aparece en la TABLA 1.

PROGRAMA 1

```
10 REM *****
    LECTURA DE
    VARIABLES DEL
    SISTEMA
    *****
20 DEF FN a(d)=PEEK d+256*PEEK
(d+1)
30 DEF FN b(d)=FN a(d)+65536*P
EEK d+1
40 REM CARGA DATOS
50 PEEK 36588:POKE 36588,6:
60 PEEK 36589:POKE 36589,6:
70 FOR n=1 TO 68: READ a$(n):
NEXT n
80 FOR n=1 TO 68: READ b$(n):
NEXT n
90 REM PIDE VARIABLE
210 POKE 23658,8
220 INPUT "Que variable desea e
xplicar? ";c$(1)
230 PRINT AT 19,0:
240 POKE 23658,0
250 IF c$(1)="" THEN CLS
260 GO TO 9999
300 REM BUSCA VARIABLE
310 FOR n=1 TO 68:
1) THEN GO TO 340
```

Programando en código máquina

El Spectrum, al igual que la mayoría de los ordenadores, permite llamar desde el Basic a rutinas escritas en código máquina. El código máquina no es realmente un lenguaje de programación (el lenguaje correspondiente es el Assembler) sino el conjunto de números que, almacenados en las posiciones de memoria, le indican al microprocesador las operaciones que debe ir ejecutando.

En lenguajes de alto nivel, como el Basic, cada comando desencadena la ejecución de cientos de instrucciones en código máquina, pero puede haber cosas que no se pueden hacer en Basic, o que se hacen más deprisa en código máquina, para ello se pregunta primero el nombre de la variable, que deberá teclearse tal como aparece en la TABLA 1.

digo máquina, para ello se ha previsto la función USR.

USR

Acceso al teclado

USR



MODO E

ATTR

Definición

La función USR con argumento numérico, ejecuta las instrucciones en código máquina correspondientes a los

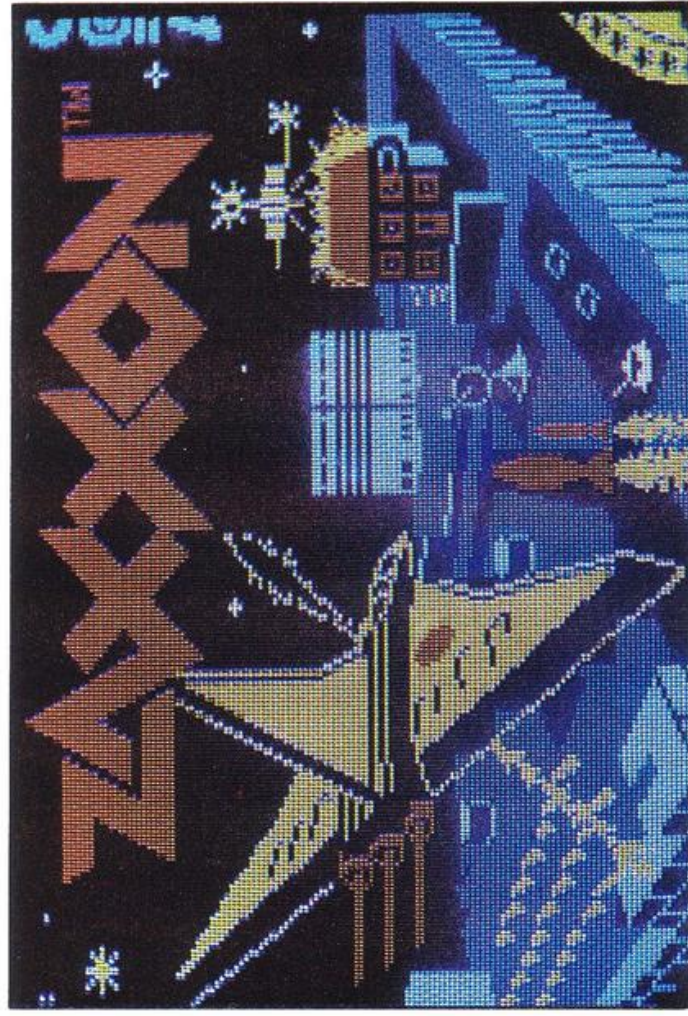
números almacenados a partir de la dirección apuntada por el argumento, hasta que se encuentre una instrucción RET (código 201), momento en el que devuelve como resultado el contenido del par de registros BC del microprocesador.

Su estructura general es:

FUNCIÓN	ARGUMENTO
USR	Dirección

Los programas en código máquina, normalmente, se escriben primero en Assembler, y luego se traducen por medio de un programa que se conoce con el nombre de "Ensamblador".

En este caso, el mismo ensamblador se encarga de introducir el programa en el ordenador. No obstante, si no



La perfección alcanzada en los juegos comerciales sólo es posible con un dominio absoluto del código máquina.

se dispone de ensamblador, también es posible hacer la traducción "a mano". En este caso, será necesario escribir un pequeño programa en Basic que se encargue de introducir el código máquina que, normalmente, se encontrará en sentencias DATA.

Hay dos formas de almacenar un programa en código máquina, una es bajar la RAMTOP (con CLEAR) y almacenar el programa por encima de ésta, con lo que quedará a salvo de borrados accidentales. Este es el sistema más usado, pero en determinados casos, puede ser interesante meter un programa corto en la memoria intermedia de impresora, si bien hay que tener en cuenta que, en este caso, será borrado por cualquier comando que utilice la impresora, o bien por el comando NEW.

Para ilustrar la velocidad y posibilidades del código máquina, hemos desarrollado una pequeña rutina que permite reenumerar las líneas del programa Basic, empezando por la línea 10 y siguiendo de 10 en 10.

En la FIGURA 5 se muestra el listado del programa en lenguaje Assembler, a la izquierda está la traducción a código máquina.

En este caso hemos preferido almacenar el programa en la memoria intermedia de impresora, con el fin de que las direcciones sean las mismas para 16 y 48 K. No obstante, el programa es "reubicable", lo que quiere decir que puede correr en cualquier posición de memoria.

El siguiente programa en Basic se encarga de introducir el código máquina en memoria:

```
10 LET d1c=23296
20 FOR n=1 TO 35
30 READ a: POKE d1c,n
40 NEXT n
50 LET d1c=d1c+1
60 DATA 35,115,267,25,35,74,9,35,235,1,1
70 DATA 6,8,9,235,24,229,267,255
```

Cuando lo haya ejecutado, puede salvarlo en cinta teclando:

```
SAVE "renum." CODE 23296,35
```

Cuando tenga un programa Basic en el que las líneas no estén numeradas de 10 en 10, cargue este reenumerador con:

```
LOAD "CODE"
```

Y cuando lo tenga, teclee:

```
RANDOMIZE USR 23296
```

Deberá obtener el mensaje:

```
0 OK, 0:1
```

que le indicará que todo ha ido correctamente; si ahora hace un listado, verá que las líneas están numeradas de 10 en 10 y empezando por la 10. Tenga en cuenta, no obstante, que los GO TO y GO SUB no habrán sido reenumerados, por lo que deberá hacerlo manualmente.

Si desea que la primera línea sea la 100 y que se numeren de 50 en 50, teclee:

```
POKE 23300,100
POKE 23323,50
```

En general, la dirección 23300 almacena el número de la primera línea (entre 0 y 255) y la dirección 23323 el incremento (también entre 0 y 255).

Si desea que un programa en código máquina se autoejecute, deberá utilizar un pequeño cargador en Basic de la forma:

```
10 LOAD "CODE:RANDOMIZE
USR (dirección)
```

que salvará en cinta con SAVE...LINE 10 antes del programa en código máquina.

```
10 DIM a(5): PRINT AT 0,0:
20 FOR n=0 TO 7
30 LET a(n+1)=IN (254+256*(255-2))
40 PRINT "a(n)=a(n+1)";a(n+1)
50 NEXT n: GO TO 10
```

Ejecute el PROGRAMA 1 y pulse varias teclas, le servirá para ver qué dato entrega cada una en su port correspondiente.

El port 254 es, sin duda, el más usado en la versión básica del Spectrum, la función IN 254 nos sirve para leer la entrada EAR, la señal está presente en el bit D6 del bus de datos.

Si lo utilizamos como salida, podremos controlar el color del borde con los bits D0, D1 y D2 del bus de datos, ejecute el siguiente programa:

```
10 REM PROGRAMA 2
20 LET x=128: LET y=88: LET i=
100 PLOT INVERSE i,x,y
110 LET a=IN 223: GO TO 150-30*(a<0)
120 LET i=(a>16): LET a=a-16*(a>16)
```

PROGRAMA 2

```
130 LET x=x+((a=1 OR a=9 OR a=5)
AND x<255)-((a=2 OR a=10 OR a=6)
AND x>0)
140 LET y=y+((a=8 OR a=9 OR a=10)
AND y<175)-((a=4 OR a=5 OR a=6)
AND y>0)
150 PLOT x,y
160 GO TO 100
```

El port 254 sirve también para hacer sonar el altavoz interno, la señal deberá estar presente en el bit D4. Por último,

```
10 FOR n=0 TO 7: OUT 254,n
20 NEXT n: GO TO 10
```

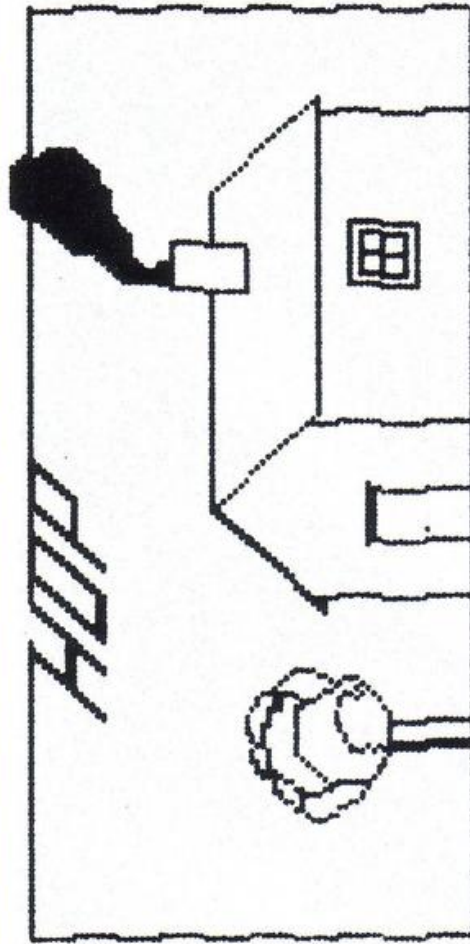
Tenga en cuenta que el color del borde es temporal, por lo que desaparecerá al pulsar cualquier tecla. Con el siguiente programa podrá conseguir un efecto curioso:

```
10 INPUT "Color del borde ";a
20 OUT 254,a: GO TO 10
```

mo, el bit D3 excita la salida MIC.

Otro port muy importante del Spectrum es el 223 ya que nos sirve para leer un joystick Kempston. Si tiene un interface joystick de este tipo, ejecute el PROGRAMA 2. Se trata de un "Tele-Sketch" con el que podrá dibujar por la pantalla, también puede borrar si mantiene apretado el pulsador de "disparo".

En la versión básica del Spectrum, los bits A5, A6 y A7 del bus de direcciones no se usan, por lo que podrá utilizarlos si decide construir su propio interface. Las direcciones adecuadas serán aquellas que hagan que A0-A4 sean todos "unos", estas direcciones son: 31, 63, 95, 127, 159, 191, 223 y 255.



Joystick Kempston y un poco de paciencia.

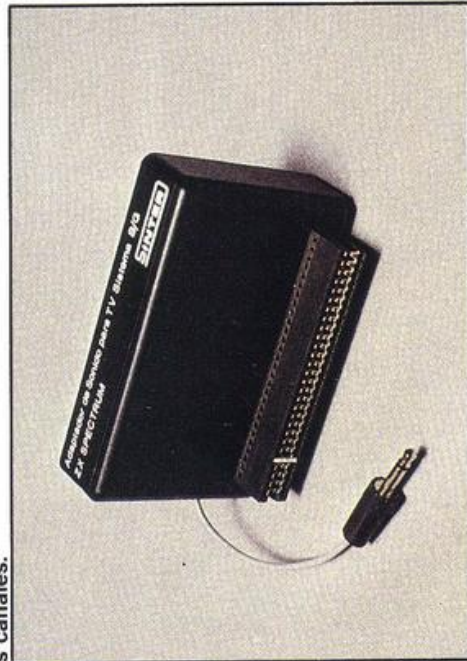


Las posibilidades sonoras del Spectrum quedan notablemente mejoradas con el uso de este sintetizador musical de tres canales.

cada uno indica una semifila (las cinco teclas derechas o izquierdas de una fila horizontal). El bit correspondiente a la fila deberá ser "0" mientras que los demás permanecerán a "1".

En la FIGURA 1 podemos ver las ocho semi-filas del teclado, a la izquierda está el número de port que se utiliza para leer cada semifila. Este dato varía según se trate de un modelo "ISSUE 2" o "ISSUE 3B". Los datos representados entre paréntesis corresponden al "ISSUE 3B" (Spectrum Plus). Si no hay ninguna tecla pulsada, el dato obtenido sería 255 en el "ISSUE 2" y 191 en el "ISSUE 3B".

En la TABLA 1 se ve la configuración binaria del bus de direcciones para cada uno de estos ports. Finalmente, el



La señal de ancho del Spectrum puede mezclarse con la de video para ser reproducida simultáneamente en T.V.

65278 :	11111110	11111110
65022 :	11111101	11111110
64510 :	11111011	11111110
63486 :	11110111	11111110
61438 :	11101111	11111110
57342 :	11011111	11111110
49150 :	10111111	11111110
32766 :	01111111	11111110

PROGRAMA 1 es un bucle que permite leer las ocho semi-filas devolviendo los datos de cada una en el "array" a (8) y presentándolos ordenados en la pantalla.

LOS PERIFERICOS

Un ordenador se compone, básicamente, de una CPU (Unidad Central de Proceso) y de una cierta cantidad de memoria. En el Spectrum, la CPU es el microprocesador Z-80. Este núcleo debe comunicarse con el exterior, para lo cual se sirve de los dispositivos periféricos.

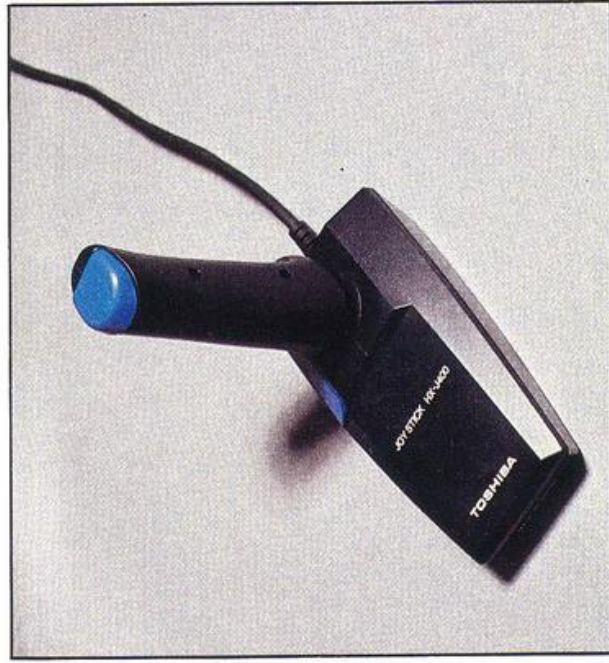
Un periférico es todo dispositivo que se une al ordenador, excepto la CPU y la memoria. Son ejemplos de periféricos, el teclado, la pantalla, el cassette, el joystick, la impresora, el Microdrive, etc.

La CPU se comunica con la memoria a través de los buses de direcciones y de datos, indicando en el bus de control, que quiere acceder a la memoria. Igualmente, para comunicarse con los periféricos utiliza los buses de direcciones y datos, pero esta vez, el bus de control indica que se está accediendo a un periférico. La pantalla es una excepción, ya que la comunicación se hace mediante la ULA que funciona como una segunda "pseudo-CPU" con mayor prioridad.

De la misma forma que cada posición de memoria tiene una dirección, los periféricos tienen también uno o varios números que los definen. Estos números se denominan "ports" (en inglés, "puertos") y cumplen la misma función que las direcciones en la me-



Entre los sistemas de almacenamiento de datos alternativos al cassette se encuentra el wafadrive.



El joystick es especialmente adecuado para los juegos.

moría. Aunque el Z-80 sólo puede direccionar 256 ports de entrada/salida, el Spectrum se las arregla de forma ingeniosa para trabajar con números de port superiores a 255.

Las instrucciones que envían y reciben datos a y desde los ports tienen una sintaxis muy similar a las de la memoria (POKE y PEEK). Vamos a verlas a continuación.

OUT

Acceso al teclado

PEEK



Definición

El comando OUT escribe un dato en un port de salida, el número de port se indica mediante la dirección.

Su estructura general es:

SENTENCIA	ARGUMENTO
OUT	Dirección, Dato

Ejemplos:

- OUT 254,16
- OUT d,27
- OUT 25+d,a
- OUT 254,a*8

La dirección ha de estar comprendida entre 0 y 65535, y el dato, ha de estar entre -255 y 255 (un número negativo equivale a 256 menos ese número).

Si alguno de estos números estuviera fuera de este margen, se produciría el error:

B Integer out of range

Acceso al teclado

CODE + SIMBOL SHIFT

Definición

La función IN tiene como argumento la dirección de un port, y devuelve como resultado el dato que se encuentra en ese momento en el port.

Su estructura general es:

SENTENCIA	ARGUMENTO
IN	Dirección

Ejemplos:

- PRINT IN 254
- LET a=IN 32766
- PRINT 27+IN 223
- LET a=IN b

La dirección puede ser cualquier número entero comprendido entre 0 y 65535. Si estuviera fuera de este margen, se produciría el error:

B Integer out of range

En el Spectrum no se puede utilizar arbitrariamente cualquier port, cada uno tiene su función concreta y existen

números de port que no tienen ningún sentido. Para comprender el funcionamiento de los ports es imprescindible atender a la configuración binaria de los buses de direcciones y datos cuando se llama a cada port.

El bus de direcciones está compuesto por 16 bits, y cada uno maneja un determinado periférico. Los bits se numeran del 0 al 15 empezando por la derecha, y precedidos de una "A" para indicar que se trata del bus de direcciones. Así el bit de más a la derecha se denomina "A0", el siguiente "A1", y así sucesivamente hasta el de más a la izquierda que se denomina "A15".

La configuración binaria que se produce en el bus de direcciones, traducida a decimal, constituye la dirección del port, por ejemplo, la configuración binaria 00000000 01111110 corresponde al port 254 (uno de los más usados en el Spectrum). El bus de datos está compuesto por 8 bits, la configuración binaria de estos bits, traducida a decimal, constituye el dato que se almacena en el port o que se lee del mismo.

Los ocho bits de la derecha del bus de direcciones (A0 a A7), indican a qué periférico se quiere acceder. Esto se indica poniendo este bit a "0" mientras que los demás permanecen a "1", sólo uno de estos bits debe ser "0" a la vez, ya que de lo contrario, se podría crear confusión en la ULA al intentar acceder a varios periféricos simultáneamente.

Los ocho bits de la izquierda (A8 a A15) deben ser normalmente cero, sólo se utilizan cuando se desea acceder al teclado, en este caso,

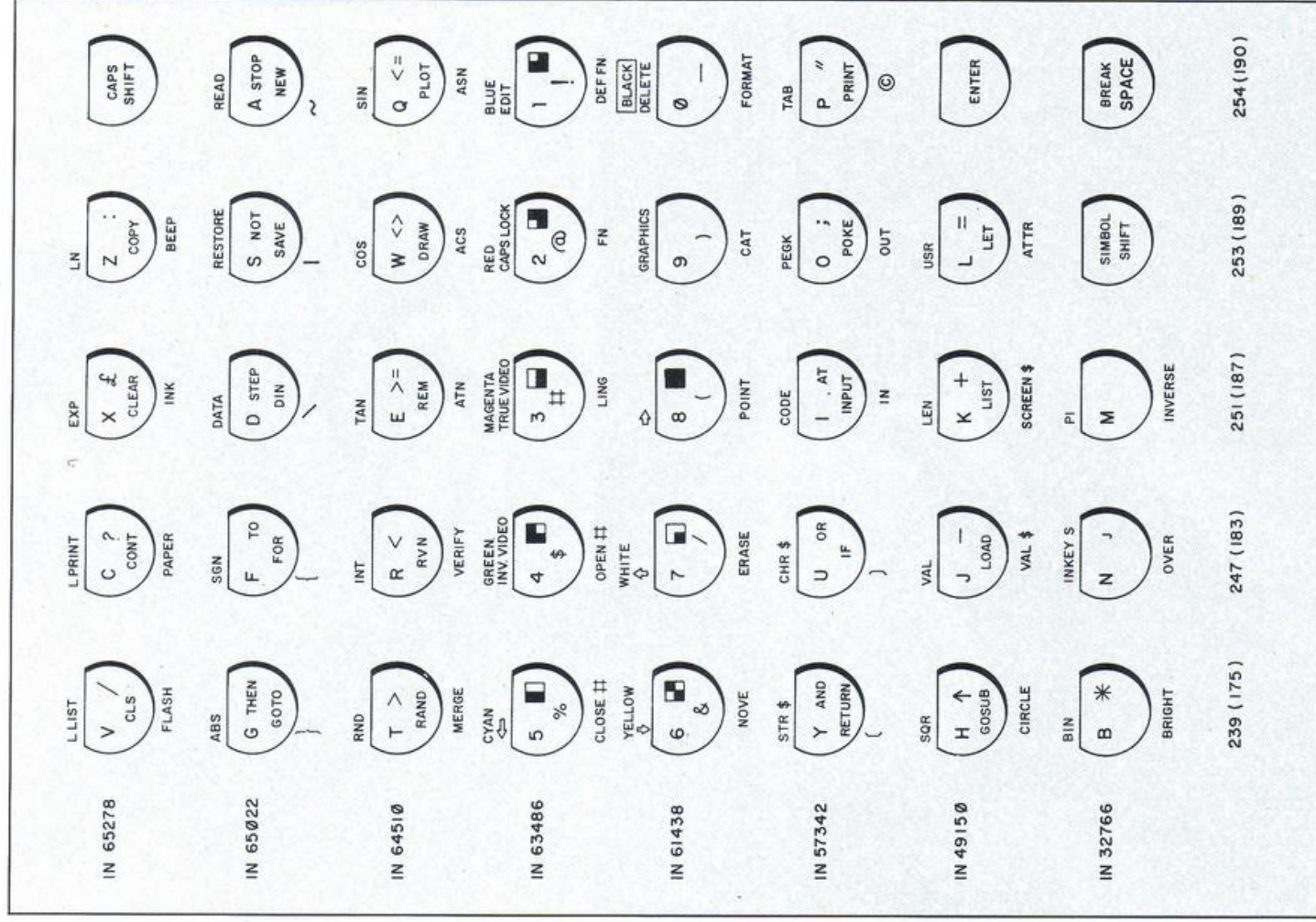


Figura 1. Los ports del teclado en el Spectrum.

Capítulo 6

OPERADORES

Introducción	33
Operadores aritméticos	33
Expresiones aritméticas	33
Cálculo de expresiones	33
Operadores de relación	34
Operadores lógicos	34
Función «AND»	34
Función «OR»	35
Función «NOT»	35
Ejercicio	36

Capítulo 7

CODIGO ASCII

Introducción _____ 37
Manejo de la tabla _____ 37
Organización del ASCII _____ 38
Transmisión del ASCII _____ 40

Capítulo 8

OPERACIONES CON CADENAS

Concatenación de cadenas	42
Subcadenas	43
Fragmentación	43
Fragmentación específica	43
Asignación de subcadenas	45
Comparación de cadenas	45
Ordenación de cadenas	47
Prioridades	48

Capítulo 9

ELABORACION DE PROGRAMAS

Introducción	50
Análisis	50
Síntesis	51
Representación gráfica	51
Programa	52

Parte II

Capítulo 10

EL JUEGO DE SENTENCIAS

Clasificación	60
Comandos de control	60
Comandos de programación	60
Comandos de entrada/salida	60
Manejo de cadenas	61
Funciones aritméticas	61
Funciones lógicas	61
Comandos de dibujo	61
Comandos de control de color	62
Sonido	62
Manejo de impresora	62
Interface 1	63
Manejo Microdrive	63
Auxiliares	64
Programa «COLOREAR»	64
Programa «GRAFICAS»	64
Programa «GDU»	66
Programa «DIBUJANDO»	66
Programa «BIPBIP»	68

Capítulo 17

COMANDOS BASICOS

REM	69
Funciones de Video	70
LET	70
Representación gráfica	71
PRINT	71
Desplazamientos	72
Formatos	72

```

0010 2000  VARTABLE TPO 5
0020 0000  SUB TIME TPO 5
0030 0000  SUB TIME TPO 5
0040 0000  SUB TIME TPO 5
0050 0000  SUB TIME TPO 5
0060 0000  SUB TIME TPO 5
0070 0000  SUB TIME TPO 5
0080 0000  SUB TIME TPO 5
0090 0000  SUB TIME TPO 5
0100 0000  SUB TIME TPO 5
0110 0000  SUB TIME TPO 5
0120 0000  SUB TIME TPO 5
0130 0000  SUB TIME TPO 5
0140 0000  SUB TIME TPO 5
0150 0000  SUB TIME TPO 5
0160 0000  SUB TIME TPO 5
0170 0000  SUB TIME TPO 5
0180 0000  SUB TIME TPO 5
0190 0000  SUB TIME TPO 5
0200 0000  SUB TIME TPO 5
0210 0000  SUB TIME TPO 5
0220 0000  SUB TIME TPO 5
0230 0000  SUB TIME TPO 5
0240 0000  SUB TIME TPO 5
0250 0000  SUB TIME TPO 5
0260 0000  SUB TIME TPO 5
0270 0000  SUB TIME TPO 5
0280 0000  SUB TIME TPO 5
0290 0000  SUB TIME TPO 5
0300 0000  SUB TIME TPO 5
0310 0000  SUB TIME TPO 5
0320 0000  SUB TIME TPO 5
0330 0000  SUB TIME TPO 5
0340 0000  SUB TIME TPO 5
0350 0000  SUB TIME TPO 5
0360 0000  SUB TIME TPO 5
0370 0000  SUB TIME TPO 5
0380 0000  SUB TIME TPO 5
0390 0000  SUB TIME TPO 5
0400 0000  SUB TIME TPO 5
0410 0000  SUB TIME TPO 5
0420 0000  SUB TIME TPO 5
0430 0000  SUB TIME TPO 5
0440 0000  SUB TIME TPO 5
0450 0000  SUB TIME TPO 5
0460 0000  SUB TIME TPO 5
0470 0000  SUB TIME TPO 5
0480 0000  SUB TIME TPO 5
0490 0000  SUB TIME TPO 5
0500 0000  SUB TIME TPO 5
0510 0000  SUB TIME TPO 5
0520 0000  SUB TIME TPO 5
0530 0000  SUB TIME TPO 5
0540 0000  SUB TIME TPO 5
0550 0000  SUB TIME TPO 5
0560 0000  SUB TIME TPO 5
0570 0000  SUB TIME TPO 5
0580 0000  SUB TIME TPO 5
0590 0000  SUB TIME TPO 5
0600 0000  SUB TIME TPO 5
0610 0000  SUB TIME TPO 5
0620 0000  SUB TIME TPO 5
0630 0000  SUB TIME TPO 5
0640 0000  SUB TIME TPO 5
0650 0000  SUB TIME TPO 5
0660 0000  SUB TIME TPO 5
0670 0000  SUB TIME TPO 5
0680 0000  SUB TIME TPO 5
0690 0000  SUB TIME TPO 5
0700 0000  SUB TIME TPO 5
0710 0000  SUB TIME TPO 5
0720 0000  SUB TIME TPO 5
0730 0000  SUB TIME TPO 5
0740 0000  SUB TIME TPO 5
0750 0000  SUB TIME TPO 5
0760 0000  SUB TIME TPO 5
0770 0000  SUB TIME TPO 5
0780 0000  SUB TIME TPO 5
0790 0000  SUB TIME TPO 5
0800 0000  SUB TIME TPO 5
0810 0000  SUB TIME TPO 5
0820 0000  SUB TIME TPO 5
0830 0000  SUB TIME TPO 5
0840 0000  SUB TIME TPO 5
0850 0000  SUB TIME TPO 5
0860 0000  SUB TIME TPO 5
0870 0000  SUB TIME TPO 5
0880 0000  SUB TIME TPO 5
0890 0000  SUB TIME TPO 5
0900 0000  SUB TIME TPO 5
0910 0000  SUB TIME TPO 5
0920 0000  SUB TIME TPO 5
0930 0000  SUB TIME TPO 5
0940 0000  SUB TIME TPO 5
0950 0000  SUB TIME TPO 5
0960 0000  SUB TIME TPO 5
0970 0000  SUB TIME TPO 5
0980 0000  SUB TIME TPO 5
0990 0000  SUB TIME TPO 5
1000 0000  SUB TIME TPO 5

```

Bytes		dirección		Contenido
Nombre	dec.	hexa.		
ATTR P	1	23693	5C8D	Atributos permanentes en curso.
ATTR T	1	23695	5C8F	Atributos temporales en curso.
BORDCR	1	23624	5C48	Atributos de la parte inferior de la pantalla.
BREG	1	23655	5C67	Registro "B" del calculador.
CH ADD	2	23645	5C5D	Dirección del siguiente carácter que ha de ser interpretado por el intérprete de Basic.
CHANS	2	23631	5C4F	Dirección del área de información para canales.
CHARS	2	23606	5C36	Dirección del juego de caracteres, menos 256.
COORDS	2	23677	5C7D	El primer byte indica la coordenada "x" del último punto PLOTado, y el segundo byte, la coordenada "y".
CURCHL	2	23633	5C51	Dirección del canal en curso.
DATADD	2	23639	5C57	Puntero de DATAS.
DEFADD	2	23563	5C0B	Dirección del argumento de una función definida por el usuario, si se está valorando alguna, en otro caso, vale 0.
DEST	2	23629	5C4D	Dirección de la variable en una asignación.
DF CC	2	23684	5C84	Dirección de la posición de PRINT en el archivo de pantalla.
DF SZ	1	23659	5C68	Número de líneas, incluida una en blanco, de la parte inferior de la pantalla.
DFCLL	2	23686	5C86	Como "DF CC", pero para la parte inferior de la pantalla.
E LINE	2	23641	5C59	Dirección del comando que está siendo tecleado.
E PPC	2	23625	5C49	Número de la línea a la que apunta el cursor del programa ">".
ECHD E	2	23682	5C82	Número de las 33 columnas, y de las 24 líneas del final del buffer de entrada.
ERR NR	1	23610	5C3A	Código del informe, menos 1.
ERR SP	2	23613	5C3D	Dirección del elemento de la pila de máquina que se usa como retorno en caso de error.
FLAGS	1	23611	5C3B	Indicadores de control del Sistema.
FLAGS; 2	1	23658	5C6A	Cursor "L" o "C". "L" = 0, "C" = 8.
FLAG	1	23665	5C71	Indicadores de control del Sistema.
FRAMES	3	23672	5C78	Reloj en tiempo real, se incrementa cada 50 milisegundos.
K CUR	2	23643	5C5B	Dirección del cursor.

Bytes		dirección		Contenido
Nombre	dec.	hexa.		
K DATA	1	23565	5C0D	Segundo byte de los controles de color introducidos por el teclado.
K STATE	8	23552	5C00	Ocho variables intermedias, usadas para leer el teclado.
LAST K	1	23560	5C08	Código de la última tecla pulsada.
LISTSP	2	23615	5C3F	Dirección de retorno tras un listado automático.
MASK P	1	23694	5C8E	Máscara para colores transparentes.
MASK T	1	23696	5C90	Como "MASK P", pero temporal.
MEM	2	23656	5C68	Dirección del área usada como memoria del calculador.
MEMBOT	2	23698	5C92	Base de la memoria del calculador.
MODE	1	23617	5C41	Modo de cursor "K"=0, "E"=1, "G"=2
NEWPPC	2	23618	5C42	Número de línea a la que hay que saltar.
NMI	2	23728	5C80	Vector de interrupción no enmascarable (anulada para facilitar la protección del software comercial).
NSPPC	1	23620	5C44	Número de sentencia dentro de una línea, a la que hay que saltar.
NXTLIN	2	23637	5C55	Dirección de la siguiente línea de programa.
OLDPPC	2	23662	5C6E	Línea a la que salta CONTINUE.
OSPPC	1	23664	5C70	Número de sentencia, dentro de la línea a la que salta CONTINUE.
P FLAG	1	23697	5C91	Indicadores para la impresión.
P POSN	1	23679	5C7F	Número de las 33 columnas de la posición de la impresora.
P RAMT	2	23732	5CBA	Dirección del último byte de RAM física.
PIP	1	23609	5C39	Duración del tono emitido al pulsar una tecla.
PPC	2	23621	5C45	Contador del programa en Basic (número de línea).
PR CC	1	23680	5C80	Byte menos significativo de la dirección de la siguiente posición de LPRINT.
PROG	2	23635	5C53	Dirección de inicio del programa Basic.
RAMTOP	2	23730	5C82	Dirección del último byte del área de memoria ocupada por el sistema Basic.
RASP	1	23608	5C38	Duración de la señal emitida al llenarse la memoria.
REPDEL	1	23561	5C09	Tiempo (en 1/50 de segundo) que ha de estar pulsada una tecla para que comience a repetirse.
REPPER	1	23562	5C0A	Tiempo (en 1/50 de segundo) entre sucesivas repeticiones de una tecla.
S POSN	2	23688	5C88	El primer byte contiene el número de 33 columnas de la posición de PRINT, el segundo contiene el número de 24 líneas.
S TOP	2	23660	5C6C	Número de la línea superior en un listado automático.
SCR CT	1	23692	5C8C	Contador de (Scroll), es siempre 1 más que el número de líneas que se han de subir antes de preguntar (scroll?).
SEED	2	23670	5C76	Origen de las operaciones para generar un número pseudo-aleatorio.
SPOSML	2	23690	5CBA	Como (S POSN), pero para la parte inferior de la pantalla.
STKBOT	2	23651	5C63	Dirección del fondo de la pila del calculador.
STKEND	2	23653	5C65	Dirección de la parte superior de la pila del calculador, e inicio del área de reserva.
STRLIN	2	23666	5C72	Longitud de la cadena de destino en una asignación.
STRMS	38	23568	5C10	Canales unidos a las corrientes abiertas.
SUBPPC	1	23623	5C47	Contador de programa Basic (número de sentencia dentro de la línea).
T ADDR	2	23668	5C74	Dirección del siguiente elemento de la tabla sintáctica.
TVDATA	2	23566	5C0E	Bytes de color y controles AT y TAB que van al televisor.
TVFLAG	1	23612	5C3C	Indicadores asociados con el televisor.
UDG	2	23675	5C7B	Dirección del área de gráficos definidos por el usuario.
VARS	2	23627	5C4B	Dirección del área de variables.
WORKSP	2	23649	5C61	Dirección del área de trabajo.
X PTR	2	23647	5C5F	Dirección del carácter que sigue al signo (?).

INDICE

INTRODUCCION

El lenguaje del Spectrum _____

Parte I

Capítulo 1

TECLADO DEL SPECTRUM

Acceso al teclado _____
 Modo **K** _____
 Modo **L** _____
 Modo **C** _____
 Modo **E** _____
 Modo **G** _____

Capítulo 2

CONFECCION DE PROGRAMAS

Edición de programas _____
 Corrección de errores _____
 Ejercicio _____

Capítulo 3

EL SPECTRUM PLUS

Teclado del «ZX Spectrum +» _____
 Modos **L** **C** _____

Modo **E** _____
 Modo **G** _____
 Edición de programas _____

Capítulo 4

ALMACENAMIENTO DE PROGRAMAS

Introducción _____
 Verificación _____
 Recuperación de programas _____
 Protección de programas _____
 Conservación de cintas _____
 Oscilación _____
 Ajuste _____

Capítulo 5

CONSTANTES Y VARIABLES

Introducción _____
 Constantes numéricas _____
 Notación entera _____
 Notación decimal _____
 Notación exponencial _____
 Notación binaria _____
 Decimal-Binario _____
 Binario-Decimal _____
 Ejercicio _____
 Constantes alfanuméricas _____
 Variables numéricas _____
 Variables alfanuméricas _____

Código		Caracter
Dec.	Hexa	
128	80	.
129	81	.
130	82	.
131	83	.
132	84	.
133	85	.
134	86	.
135	87	.
136	88	.
137	89	.
138	8A	.
139	8B	.
140	8C	.
141	8D	.
142	8E	.
143	8F	.
144	90	U06 "A"
145	91	U06 "B"
146	92	U06 "C"
147	93	U06 "D"
148	94	U06 "E"
149	95	U06 "F"
150	96	U06 "G"
151	97	U06 "H"
152	98	U06 "I"
153	99	U06 "J"
154	9A	U06 "K"
155	9B	U06 "L"
156	9C	U06 "M"
157	9D	U06 "N"
158	9E	U06 "O"
159	9F	U06 "P"
160	A0	U06 "Q"
161	A1	U06 "R"
162	A2	U06 "S"
163	A3	U06 "T"
164	A4	U06 "U"

Figura 6. Caracteres gráficos.

Código		Token
Dec.	Hexa	
165	A5	RND
166	A6	INKEY\$
167	A7	PI
168	A8	FN
169	A9	POINT
170	AA	SCREEN\$
171	AB	ATTR
172	AC	AT
173	AD	TAB
174	AE	VAL\$
175	AF	CODE
176	B0	VAL
177	B1	LEN
178	B2	SIN
179	B3	COS
180	B4	TAN
181	B5	ASN
182	B6	ACS
183	B7	ATN
184	B8	LN
185	B9	EXP
186	BA	INT
187	BB	SQR
188	BC	SGN
189	BD	ABS
190	BE	PEEK
191	BF	IN
192	C0	USR
193	C1	STR\$
194	C2	CHR\$
195	C3	NOT
196	C4	BIN
197	C5	OR
198	C6	AND
199	C7	<=
200	C8	>=
201	C9	<>
202	CA	LINE
203	CB	THEN
204	CC	TO
205	CD	STEP
206	CE	DEF FN
207	CF	CAT
208	D0	FORMAT
209	D1	MOVE
210	D2	ERASE

Código		Token
Dec.	Hexa	
211	D3	OPEN #
212	D4	CLOSE #
213	D5	MERGE
214	D6	VERIFY
215	D7	BEEP
216	D8	CIRCLE
217	D9	INK
218	DA	PAPER
219	DB	FLASH
220	DC	BRIGHT
221	DD	INVERSE
222	DE	OVER
223	DF	OUT
224	E0	LPRINT
225	E1	LLIST
226	E2	STOP
227	E3	READ
228	E4	DATA
229	E5	RESTORE
230	E6	NEW
231	E7	BORDER
232	E8	CONTINUE
233	E9	DIM
234	EA	REM
235	EB	FOR
236	EC	GO TO
237	ED	GO SUB
238	EE	INPUT
239	EF	LOAD
240	F0	LIST
241	F1	LET
242	F2	PAUSE
243	F3	NEXT
244	F4	POKE
245	F5	PRINT
246	F6	PLOT
247	F7	RUN
248	F8	SAVE
249	F9	RANDOMIZE
250	FA	IF
251	FB	CLS
252	FC	DRAW
253	FD	CLEAR
254	FE	RETURN
255	FF	COPY

Figura 7. Códigos de «Tokens».

EL JUEGO DE CARACTERES

Cada uno de los signos, letras o números que puede imprimir el Spectrum se corresponde con un número comprendido entre 0 y 255, a este número se le denomina código.

Existen varios sistemas normalizados de codificación de caracteres, el más utilizado en informática se denomina ASCII, y es el que utiliza el Spectrum, con ligeras variaciones.

EL ASCII

La palabra ASCII está compuesta por las siglas de «American Standard Code for Information Interchange» (Código Normalizado Americano para Intercambio de Información).

El ASCII se elaboró en los Estados Unidos a finales de la década de los 60. En principio se pensó para su uso en teletipos, estos aparatos transmiten 7 bits de código más uno de «paridad», por lo que el ASCII utiliza códigos comprendidos entre 0 y 127. La decodificación en los teletipos se hacía de forma mecánica, por lo que este proceso tuvo que ser simplificado al máximo. El ASCII utiliza los cinco bits de menos peso para designar un carácter determinado, y los dos siguientes, para indicar si se trata de

	001.....	010.....	011.....
...00000	espacio		
...00001	!	@	f
...00010	"	A	a
...00011	#	B	b
...00100	\$	C	c
...00101	%	D	d
...00110	&	E	e
...00111	'	F	f
...01000	(G	g
...01001)	H	h
...01010	*	I	i
...01011	+	J	j
...01100	,	K	k
...01101	-	L	l
...01110	.	M	m
...01111	/	N	n
...10000	0	O	o
...10001	1	P	p
...10010	2	Q	q
...10011	3	R	r
...10100	4	S	s
...10101	5	T	t
...10110	6	U	u
...10111	7	V	v
...11000	8	W	w
...11001	9	X	x
...11010	:	Y	y
...11011	;	Z	z
...11100	<	[{
...11101		\	
...11110			
...11111			

un código no imprimible (de control), un número o signo, una mayúscula o una minúscula.

Los 32 primeros códigos de control, y no producían la impresión de ningún carácter. Los 96 restantes constituyen propiamente, el juego de caracteres del ASCII, el código 127 se usa con frecuencia, para indicar el borrado del último carácter impreso. En la FIGURA 1 se pueden ver todos los caracteres del ASCII, ordenados según la configuración binaria de su código.

El juego de caracteres del Spectrum

El Spectrum utiliza una variante del ASCII, los 32 primeros códigos son también de control, si bien cumplen funciones bastante diferentes a las asignadas por el ASCII. Los 96 siguientes son idénticos a los caracteres ASCII, salvo el código 127 que se utiliza para el signo de "Copyright".

Por otro lado, el Spectrum no necesita comprobación de paridad, por lo que los ocho bits están disponibles. Ello permite duplicar el número de códigos utilizables.

Los 37 códigos siguientes al 127, se han utilizado para caracteres gráficos, de los cuales, los primeros 16 están definidos, y los 21 restantes son definibles por el usuario (los famosos UDG).

Los últimos 91 códigos han sido asignados a los "Tokens" que utiliza el Spectrum. La función CODE aplicada sobre una cadena, da como resultado el código del primer carácter que la compone. Su

...11101	=]	}
...11110	>	^	~
...11111	?	-	

Figura 1. Configuración binaria del ASCII.

Código		Funcion
Dec.	Hexa	
0	00	No Utilizado
1	01	No utilizado
2	02	No utilizado
3	03	No utilizado
4	04	No utilizado
5	05	No utilizado
6	06	CAPS LOCK y efecto de "coma" en impresión.
7	07	EDIT
8	08	Cursor Izquierda
9	09	Cursor Derecha
10	0A	Cursor Abajo
11	0B	Cursor Arriba
12	0C	DELETE (Borrado)
13	0D	ENTER (Nueva línea)
14	0E	CAPS SHIFT + SYMBOL SHIFT y "número" dentro de un programa.
15	0F	No Utilizado
16	10	INK (Control de tinta).
17	11	PAPER (Control de papel).
18	12	FLASH (Control de parpadeo).
19	13	BRIGHT (Control de brillo)
20	14	INVERSE (Control de inversión de video).
21	15	OVER (Control de sobreimpresión).
22	16	AT (Control de posicionamiento).
23	17	TAB (Control de tabulación)
24	18	No Utilizado
25	19	No Utilizado
26	1A	No Utilizado
27	1B	No Utilizado
28	1C	No Utilizado
29	1D	No Utilizado
30	1E	No Utilizado
31	1F	No Utilizado

Figura 2. Códigos de Control.

inversa, CHR\$ aplicada sobre un número, da como resultado el carácter que tiene ese número como código.

La FIGURA 2, muestra la lista de códigos de control utilizados en el Spectrum. Las FIGURAS 3, 4 y 5 muestran las

tres partes de juego de caracteres. La FIGURA 6 corresponde a los caracteres gráficos y por último, la FIGURA 7 muestra la lista de "Tokens"

con sus códigos.

En todos los casos, los códigos han sido representados en decimal como en hexa.

Codigo		Caracter
Dec.	Hexa	
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D]
94	5E	^
95	5F	_

Figura 4. Letras mayúsculas.

Codigo		Caracter
Dec.	Hexa	
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	
125	7D	}
126	7E	~
127	7F	(c)

Figura 5. Letras minúsculas.

TAB	73
AT	73
Canales de comunicación	75
INPUT	77
INPUT TAB y AT	79
INPUT LINE	80
Otra aplicación	80
Programas de repaso	81
Programa «GRANJA»	81
Programa «EDUCACION»	83
Programa «INTERES»	84
Programa «GRADOS»	86
Programa «FICHA»	86

Introducción	108
DO WHILE	108
REPEAT UNTIL	108
Diferencias	108
FOR/NEXT	109
STEP	111
Bucles anidados	112
Errores	113
Programas	113

Capítulo 15

Capítulo 12

COMANDOS DE CONTROL

RUN	89
BREAK	90
STOP	91
Ruptura del «INPUT»	92
CONTINUE	94
Informes de pantalla	96
NEW	97
CLS	97
LIST	98
LIST y EDIT	98

Capítulo 13

SALTOS INCONDICIONALES Y CONDICIONALES

Introducción	99
GO TO	99
IF... THEN...	101
Evaluación de las condiciones	103
Programa	107

Capítulo 14

BUCLES

INKEY\$	142
PAUSE	142
Programas	144

SUBROUTINAS

Introducción	122
GO SUB	122
RETURN	122
Utilización de «GOSUB» y «RETURN»	122
Tipos de subrutinas	124
Subrutinas anidadas	125
Error	126
Programas	127

Capítulo 16

DATOS DE UN PROGRAMA

Introducción	134
READ	134
DATA	134
Utilización de «READ» y «DATA»	134
RESTORE	135
Errores	137
Programas	140

Capítulo 17

LECTURA DEL TECLADO Y TEMPORIZACIONES

INKEY\$	142
PAUSE	142
Programas	144

Capítulo 18

FUNCIONES

Introducción	148
Funciones numéricas	148
ABS	148
INT	148
SGN	149
SQR	149
BIN	150
PI	150
El radián	151
SIN	151
COS	152
TAN	152
ASN	152
ACS	152
ATN	152
Aplicación de la trigonometría	153
Función exponencial	154
EXP	154
Función logarítmica	155
LN	155
Definición de funciones	159
DEF FN	159
FN	159
Errores	160

Capítulo 19

FUNCION ALEATORIA

Introducción	161
RND	161
Programa «BARQUITOS»	164
RANDOMIZE	164
Programa «TABLA»	168

Capítulo 20

FUNCIONES DE CADENA

Introducción	169
--------------	-----

LEN	170
STR\$	171
VAL	171
VAL\$	172
Conversiones de código	172
CHR\$	172
CODE	174
Funciones definidas de cadena	175
Errores	175
Programa	177

Capítulo 21

MATRICES

Introducción	179
Dimensionado de matrices	179
DIM	180
Matrices numéricas	180
Asignación y visualización	182
Manejo de tablas	185
Matrices de cadena	195
Asignación	196
Fragmentación	196
Errores	199
Grabación de datos	200
Programa	202

Capítulo 22

DEPURACION DE PROGRAMAS

Introducción	206
Errores	206
Depuración	207
STOP y CONTINUE	209
Programa «Depurador»	209
Ejercicio	210

Capítulo 23

COLOR

Introducción	211
--------------	-----

PROGRAMAS «MICROBASIC»

EJER1	10, 11, 15 y 16	CARTAS	189, 190 y 191
CODEBIN	25, 26 y 27	11 ERRORES	209
ASCII	37	SIN ERRORES	210
ASCII/DECIMAL	39	ATRIBUTOS	224
CHR\$	40	DEPURACION	225
CONCATENACION	42	CARTA COLOR	225
FRAGMENTACION	44 y 45	COLORES 1	225
COMPARACION	47	COLORES 2	225 y 226
ORDENA	48 y 49	DIBUJO PEZ	227 y 228
LISTIN	51, 52, 53 y 59	MICROHOBBY	230
COLOREAR	61	ABSTRACTO	232
GRAFICAS	62	GRAFICO	233
G.D.U.	64	RECTAS	234
DIBUJANDO	65	MAPA	237, 238 y 239
BIPBIP	66	OVER	239
GRANJA	74 y 75	LABERINTO	240
EDUCACION 2 GRADO	75 y 76	ALFABETO ESPAÑOL	246
INTERES SIMPLE	77	NOTAS GRAFICAS	247
GRADOS	78 y 79	LECTURA «GDU»	250
FICHA	79 y 80	PALITROQUE	251, 252 y 253
CALCULADORA	92 y 93	NAVIDAD	259
iiiNEW!!!	93 y 94	DIATONICA	259
AGENDA	102	CROMATICA	260
AREAS Y PERIMETROS	105 y 106	DOS CRUCES	260 y 261
ESTADISTICA	114 y 115	EFFECTO 1	263
HISTOGRAMAS	116	EFFECTO 2	263
ADIVINO	126	EFFECTO 3	263
LONGITUD	130	EFFECTO 4	263
GEOGRAFIA 1	138	NAVIDAD (TREMOLLO)	264
GEOGRAFIA 2	139 y 140	LISTADOR	268
MAQUINA	145	DIRECTORIO	270 y 271
MOVIMIENTO	157	EDIT/DIR	273
iiiAGUA!!!	165	IMPRESORA (TEST)	277
LA TABLA	166	FICHERO	285 y 286
BUSQUEDA	169	RENUMERADOR	304 y 306
INSERTAR	169	SCAN DE TECLADO	311
ANULAR	171	SKETCH	311
INPUT	174	LECTURA DE VARIABLES	312
MANEJO DE TABLAS	179 y 180		

GESTION DE IMPRESORA

Introducción	276
LPRINT	276
Programa TEST DE IMPRESORA	277
LLIST	277
COPY	278
Ejemplo de COPY	278
Otras impresoras	278
Tipos de impresoras	279
Elección de una impresora	280
Juego de caracteres	281

Capítulo 29

INTERFACE 7

Introducción _____ 283
Canales y corrientes _____ 283
Asignación de canales
y corrientes _____ 283
OPEN # _____ 283

Desactivación de canales

y corrientes	284
CLOSE #	284
El Microdrive	284
FORMAT	284
Programa FICHERO	285
CAT	286
Grabación y carga	287
Borrado de programas	287
ERASE	288
Ejecución automática	288
Protección de ficheros	288
Ficheros de datos	288
Grabación de datos	289
Apertura de fichero	289
Cierre de ficheros	289
Lectura y ampliación de ficheros	290
Red de área local	292
Interface RS-232	293

Capítulo 30

LOS PERIFERICOS

Introducción	307
OUT	308
IN	308
Ports del teclado	309
Programa SCAN DEL TECLADO	311
Programa SKETCH	311

Capítulo 32

VARIABLES DEL SISTEMA

312	Introducción
312	Programa LECTOR DE
313	VARIABLES
	Tabla de variables del Sistema

SENTENCIAS Y FUNCIONES

A		ERASE	288
		EXP	154
<hr/>			
ABS	148		
ACS	152		
AND	34		
ASN	152		
AT	73		
ATN	152	FLASH	221
ATTR	222	FN	159
		FOR	109
		FORMAT	284

BEEP _____ 258
BIN _____ 150
BORDER _____ 214
BRIGHT _____ 219
GO TO _____ 122
GOSUB _____ 99

C	CAT	286	IF	101
	CHR\$	172	IN	308
	CIRCLE	234	INK	215
	CLEAR	302	INKEY\$	142
	CLOSE #	284	INPUT	77
	CLS	97	INT	148
	CODE	174	INVERSE	220
	CONTINUE (CONT)	94 y 209		
	COPY	278		
	COS	152		

D

DATA _____ 134

DEF FN _____ 159

DIM _____ 180

DRAW _____ 229

LEN _____ 170

LET _____ 70

LINE _____ 80

LIST _____ 98

LLIST _____ 277

LN _____ 155

LOAD _____ 272