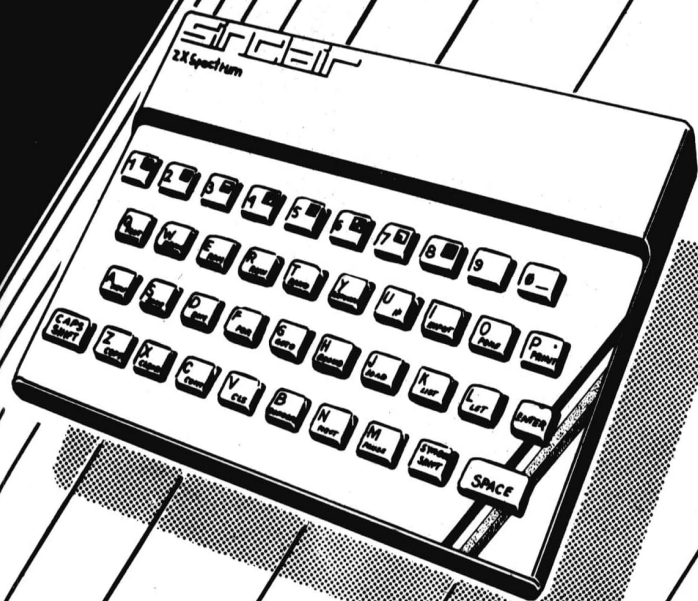


Bosetti

ZX Spectrum

Consejos y Trucos

UN LIBRO DATA BECKER
EDITADO POR FERRE MORET, S.A.



Bosetti
ZX
Spectrum
Consejos y Trucos

UN LIBRO DATA BECKER
EDITADO POR FERRE MORET, S.A.

Los programas y utilidades que se encuentran en este libro pueden ser adquiridos en Cassette solicitándolos directamente a FERRE MORET, S.A., o en sus distribuidores

Este libro ha sido traducido por Dn. Juan Jiménez Molina Ingeniero Industrial y experto conocedor del ZX Spectrum

APSSA, ROCA UMBERT, 26 - L'HOSPITALET DE LL. (Barcelona)

ISBN 84-86437-10-5

Deposito Legal B-30.843-1985

Copyright (C) 1984 DATA BECKER GmbH
Merowingerstr. 30
4000 Düsseldorf

Copyright (C) 1985 FERRE MORET, S.A.
Tuset n.8 ent. 2
08006 Barcelona

Reservados todos los derechos. Ninguna parte de este libro podrá ser reproducida de algún modo (impresión, fotocopia o cualquier otro procedimiento) o bien, utilizado, reproducido o difundido mediante sistemas electrónicos sin la autorización previa de FERRE MORET, S.A.

Advertencia importante

Los circuitos, procedimientos y programas reproducidos en este libro, son divulgados sin tener en cuenta el estado de las patentes. Están destinados exclusivamente al uso amateur o docente, y no pueden ser utilizados para fines comerciales.

Todos los circuitos, datos técnicos y programas de este libro, han sido elaborados o recopilados con el mayor cuidado por el autor y reproducidos utilizando medidas de control eficaces. No obstante, es posible que exista algún error. FERRE MORET, S.A. se ve por tanto obligada a advertirles, que no puede asumir ninguna garantía, ni responsabilidad jurídica, ni cualquier otra responsabilidad sobre las consecuencias atribuibles a datos erróneos. El autor les agradecerá en todo momento la comunicación de posibles fallos.

I N D I C E

P R O L O G O

CAPITULO 1:

ASPECTOS GENERALES DEL SPECTRUM

Introducción.....	3
Del ZX 81 al ZX Spectrum.....	7

CAPITULO 2:

CONSEJOS PARA LA PROGRAMACION EN BASIC

Aspectos generales.....	11
Programar efectivamente.....	18

CAPITULO 3:

PEEKs Y POKES Y USRS

La memoria del Spectrum.....	22
Las variables del sistema.....	25
Algunos consejos y trucos.....	30

CAPITULO 4:

RUTINAS UTILES Y PROGRAMAS CORTOS

Letra a doble tamaño.....	35
64 caracteres por línea.....	39
Mastermind.....	49
Listado de reloj tipo "telediarlo".....	55
Listado calendario.....	56
Listado biorritmo.....	58

CAPITULO 5:

UN JUEGO PARA MENTES DESPIERTAS : ADIVINAR SIMBOLOS

Explicación del juego.....	60
Desarrollo del programa.....	62
El listado.....	73

CAPITULO 6:

SPECTRUM ESPECTACULAR : DIA Y NOCHE EN LA TIERRA

Desarrollo del programa.....	79
El listado.....	85

CAPITULO 7:

APARATOS ADICIONALES Y AMPLIACIONES

Introducción.....	92
El Interface 1.....	94
Impresoras en serie.....	98
Lápiz óptico.....	101
Procesado de texto.....	103
Compilador.....	110

CAPITULO 8:

REPRESENTACIONES GRAFICAS

Introducción.....	112
Desarrollo del programa Busygraph.....	113
Trabajar con el programa.....	122
El listado.....	126

CAPITULO 9:

EL TABLON DE ANUNCIOS : PUBLICIDAD CON EL SPECTRUM

Introducción.....	130
El desarrollo del programa.....	134
El listado.....	150

CAPITULO 10:

EL SPECTRUM EN LA EMPRESA PEQUEÑA

Aspectos generales.....	155
El programa de comando.....	157
El fichero de clientes Busydat.....	159
Aplicación de Busydat.....	169
Programa Busyman de administración de almacén y volumen de ventas.....	171
Aplicación de Busyman.....	187

APENDICE:

1. El juego de comandos del Spectrum.....	190
2. Indicaciones de error.....	206

P R O L O G O

Si el proceso de desarrollo de la aviación en los últimos años hubiese sido paralelo al de la industria de los ordenadores, hoy sería posible volar por un precio inferior a 3000 Ptas, en menos de media hora, de Frankfurt a Tokyo.

A causa de este desarrollo tan importante, los microordenadores han conquistado el mundo y, hoy por hoy, nuestra vida es inimaginable sin microprocesadores.

Clive Sinclair ha tenido una gran participación en la expansión de los ordenadores, basada en el desarrollo del ZX-81 y ZX Spectrum, pues ambos permiten una introducción económica, pero rentable, en este campo.

Este libro va dirigido a todos los actuales y futuros propietarios de un ordenador Spectrum, que ya estén un poco familiarizados con ordenadores. Estos conocerán ya los términos elementales de BASIC y posiblemente hayan creado algunos programas, deseando ahora sacar más provecho de su ordenador.

Este libro contiene muchas sugerencias, consejos, trucos y programas, los cuales pueden resumirse en forma de biblioteca, y así facilitar la creación de programas propios.

Después de algunos programas cortos hemos desarrollado un primer programa de juego más largo, en el cual lo más interesante no es matar el mayor número de enemigos, sino solucionar un problema de forma inteligente.

En el siguiente programa se aprovechará la capacidad gráfica del Spectrum. Cuando esté terminado, la pantalla de la televisión, no sólo mostrará un mapamundi, sino también la situación del sol, así como los sectores de día y noche en la tierra.

Al surgir los Microdrives como memoria de masa económica y rápida, las posibilidades de aplicación de Spectrum se han ampliado considerablemente, sobre todo teniendo en cuenta que muchos fabricantes ofrecen una amplia gama de accesorios (sobre todo teclados). Sobre aparatos adicionales se informará ampliamente.

Por lo demás se desarrollarán ideas y programas, los cuales traen consigo una considerable simplificación del trabajo, aún en empresas pequeñas. Estos programas son de administración de volumen de ventas, administración de almacén, un fichero de clientes y un programa de gráficos. Además, el Spectrum se utiliza como tablón de anuncios: palabras y textos de cualquier tipo pueden producirse en la pantalla de diferentes tamaños y colores.

Las explicaciones de este libro van destinadas, de una parte, a ayudar e impulsar al lector dentro de este campo y, de otra, a ofrecerle satisfacción y diversión.

CAPITULO 1: ASPECTOS GENERALES DEL SPECTRUM

Introducción

El ZX Spectrum ofrece la posibilidad de adquirir de forma económica un ordenador versátil y de conseguir, de este modo, no sólo divertirse con juegos, sino aprender "un montón" sobre ordenadores.

Si Ud. es un principiante, le recomendamos una lectura intensa del manual de manejo del Spectrum ya que, aparte de estar escrito de forma clara y concisa, contiene importantes informaciones (y no sólo para el "novato").

En este libro queremos echar un vistazo al Spectrum, con el objeto de ser completo, de manera que "ascendientes" del ZX 81 se sitúen rápidamente.

El corazón, o mejor dicho el cerebro, del Spectrum es el procesador Z80, producido por la empresa Zilog. Este procesador de 8 bits, puede direccionar 64 KByte de memoria. 16 KByte de esta memoria están ocupados por el sistema operativo, el cual está alojado en una ROM.

¿Qué es un sistema operativo? ¿Que es una ROM? El sistema operativo es un programa, o sea software, que tiene el objeto de relacionar el procesador con el mundo exterior. El mundo exterior en este caso está representado por el usuario del ordenador, así como por otros aparatos como son el televisor o el cassette, pero también el teclado del mismo ordenador. De alguna forma, el Spectrum tiene que saber lo que queremos de él.

Esto se lo comunicamos pulsando diversas teclas. El sistema operativo puede reconocer el significado de cada entrada y elaborarla. Como el usuario debe aplicar ciertas normas al realizar las entradas, el programa puede decidir si en este momento se está escribiendo un programa, o se trata de un comando a cumplir directamente.

Si se trata del primer caso, hay que conseguir que este programa se almacene en memoria trabajando después con él según las necesidades. El lenguaje, con el cual programamos al Spectrum, se llama BASIC. Se trata de la abreviación de "Beginners Allpurpose Symbolic Instruction Code". Pero desgraciadamente, el procesador no entiende este lenguaje. El sólo entiende su propio lenguaje máquina. Por ello, un programa escrito en BASIC debe ser traducido.

También este lenguaje - llamado Interpretador - está incluido en el Spectrum y se puede considerar como parte del sistema operativo. En muchos ordenadores el sistema operativo propiamente dicho y el Interpretador son cosas separadas, pero en el Spectrum van estrechamente ligadas.

Estos programas son, naturalmente, muy importantes para el usuario, y deben formar parte inalterable del ordenador. Por ello, están depositados en una parte de la memoria que sólo puede ser utilizada para su lectura. De aquí el nombre "Read Only Memory", o abreviado ROM, lo que significa "sólo para leer".

Pero si toda la memoria del ordenador se utilizara únicamente para leer, no podríamos escribir programas propios en la memoria, y el aparato no sería un verdadero ordenador. Por este motivo existe un segundo tipo de memoria, que puede ser ocupado y borrado a voluntad. Esta memoria se denomina "Random Access Memory" o abreviado RAM.

En el Spectrum los programas en el ROM ocupan 16 KByte. Para la memoria RAM quedan libres hasta 48 KByte. De éstos, unos 40,5 KByte se reservan para la programación en BASIC. Se inician a partir de la dirección 23685. Las direcciones desde 16385 hasta 23685 se necesitan para la memoria de pantalla, variables de sistema y cosas similares.

Ya que el Spectrum dispone de una resolución de 256 por 176 puntos, y cada matriz de 8 x 8 puede disponerse con un atributo de color, se utilizan 6144 direcciones para la memoria de la pantalla, y otras 704 para la información de color.

El repertorio de símbolos del Spectrum está compuesto por 256 símbolos, cuyos códigos están numerados desde 0 hasta 255. Los primeros 128 símbolos corresponden en general a la Norma ASCII. Los símbolos de los códigos desde 165 hasta 255 representan los comandos de BASIC disponibles en el Spectrum.

Esto permite introducir los comandos pulsando únicamente una tecla o una combinación de teclas. Esto no es únicamente ventajoso por la poca similitud del teclado del Spectrum con el de una máquina de escribir, sino que tiene además otras dos ventajas: Por una parte, uno no se puede equivocar al introducir comandos y por otra, estos comandos (también llamados token) necesitan solamente un Byte de memoria. Otros ordenadores necesitan frecuentemente un Byte para cada letra.

¿Cómo sabe el cerebro del Spectrum, el Z80, qué tecla ha sido pulsada y lo que esto significa? Todas las teclas están interconectadas internamente en forma de red, compuesta de 5 x 8 líneas. Una tecla corresponde exactamente a una combinación de estos grupos de líneas. Esta combinación se transmite a un Chip concreto, la ULA. Esto es una abreviación de 'Uncommitted Logical Array'. Este chip determina de qué tecla se trata.

En las variables del sistema, el Z80 encuentra la información sobre el modo actual de funcionamiento de aquél, es decir, la función que una determinada tecla tiene en este momento. Cada tecla está ocupada con diferentes funciones. Cuando se ha determinado la función, es cuando esta información se transmite a la pantalla.

No sólo es posible programar el Spectrum en BASIC. También es factible trabajar con el lenguaje máquina si ello se domina.

De todas formas, en este libro no vamos a centrarnos en el lenguaje máquina, de modo que nos dedicaremos al Basic recalcando, antes que nada, las diferencias esenciales entre el buen ZX 81 y el Spectrum. Esto reducirá el tiempo de introducción, tanto al que "asciende" del ZX 81, como al que cambia de sistema.

Del ZX 81 al ZX Spectrum

El que 'asciende' del ZX 81 al Spectrum, no tendrá grandes problemas con el nuevo ordenador, ya que está familiarizado con la filosofía de Sinclair. Unicamente debe acostumbrarse a que tiene una aplicación más amplia del BASIC.

A diferencia al ZX 81, aquí es posible introducir varios comandos en una línea, o separados por un doble punto en el modo de funcionamiento directo. Ya que en el Spectrum no se activa el Z80 para mantener la pantalla como sucede en el caso del ZX 81, no existe la diferencia entre 'SLOW' y 'FAST'. El Spectrum trabaja siempre en 'FAST'.

Tampoco es necesario activar el comando 'SCROLL' desde el BASIC. Cuando la pantalla está llena, pasa automáticamente a preguntar ¿'SCROLL'?, tras lo cual se efectúa la salida después de pulsar cualquier tecla. Si se quiere dar la orden 'SCROLL', también se puede realizar a través de una rutina ROM (RAND USR 3282).

Debido a la mayor resolución gráfica del Spectrum, hay que tener en cuenta las diferencias en la orden 'PLOT'. El comando 'UNPLOT' no existe en el Spectrum, pero sí en cambio una serie de órdenes gráficas que no se encuentran en el ZX 81. (Hay una lista completa de todos los comandos BASIC con aclaraciones en el apéndice).

En el Spectrum existen menos símbolos gráficos definidos que en el ZX 81, pero se dispone de 21 caracteres, que pueden definirse a voluntad y disponer de ellos libremente. También es posible, con un pequeño truco, definir un juego de caracteres completamente nuevo, o utilizar varios juegos de caracteres paralelamente.

Trataremos de nuevo este tema en el tercer y sobre todo en el cuarto capítulo, cuando desarrollemos una rutina, con la cual, el Spectrum es capaz de editar 64 símbolos por línea sobre la pantalla.

El Spectrum utiliza, en contraposición al ZX 81, el código ASCII.

En el ZX Spectrum es posible rebajar la RAMTOP a voluntad por medio de la orden CLEAR, indicando detrás el valor decimal máximo que se debe alcanzar con el BASIC. Los símbolos definidos por el propio usuario, los UDGs, están en un campo protegidos del NEW.

La velocidad, con la cual se memorizan programas en el cassette, es 6 veces superior al ZX 81. Aparte de ello, el proteger los programas es mucho más sencillo, y el proceso SAVE puede ser comprobado con el comando VERIFY. De esta manera se excluye en lo posible la pérdida de datos. Los programas pueden ser protegidos con LINE lo cual conduce, al cargar un programa, a una ejecución automática que se inicia en la correspondiente línea. Al cargar, el Spectrum escribe los nombres de los programas sobre la pantalla.

Además de los ocho colores de los que dispone el Spectrum, también existe algo para el oído. Con el comando 'BEEP', puede emitir sonidos. Pero al determinar este comando no estuvieron presentes ni Pink Floyd ni la Filarmónica de Viena. Pero a pesar de todo, con un poco de empeño, se pueden conseguir 'ruidos' bastante interesantes. Incluso en un programa de ajedrez es posible emitir sonidos que tienen cierta similitud con el habla humana.

Con el comando SCREEN\$, es posible en el Spectrum, memorizar en cassette el contenido completo de la pantalla, aún cuando se trate de un gráfico con alta resolución. Esto es posible porque, aparte de programas en BASIC, el Spectrum es capaz de memorizar y de cargar Bytes.

El comando `SAVE "nombre" SCREEN$` no significa otra cosa que `SAVE "nombre" código 16384,6912`. A partir de la dirección 16384, se encuentra un fichero de pantalla de una longitud de 6912 Bytes. Este es, a diferencia del ZX 81, un fichero de longitud fija.

Pero la orden `SCREEN$` también puede aplicarse de otra forma. Si se le provee con dos argumentos, como son el número de línea y el número de columna, se obtiene como resultado el signo que se encuentre en la pantalla en este punto. Una función similar se puede realizar con `POINT` (línea, columna). Con esto se puede preguntar, si un Pixel tiene en este momento el color de fondo o el color de primer plano. En este caso, la línea y la columna representan los valores del sistema gráfico, o sea, números entre 0 y 255.

También se pueden introducir números en código binario con el comando `BIN`. Así, con el comando `PRINT BIN 00000100` el ordenador responde con el valor 4. Esta posibilidad es bastante cómoda cuando se quieren definir símbolos. Se dibuja el símbolo sobre un pautado de 8 x 8, y se introduce un 1 en las partes determinadas. Así se suprime la transformación a números decimales, lo cual significa un trabajo muy laborioso cuando hay muchos símbolos. Concretaremos el método con un pequeño ejemplo. Queremos definir la S de Sinclair, para lo cual se introducirá:

```
BIN 00000000
BIN 11111111
BIN 10000000
BIN 10000000
BIN 11111111
BIN 00000001
BIN 00000001
BIN 11111111
```

en las correspondientes memorias, por ejemplo `USR "a"` hasta `USR "a+7"`. `Print USR "a"` representa entonces un signo de la forma correcta.

En el Spectrum es posible memorizar datos con órdenes DATA y leerlos con READ. Por medio de RESTORE, es posible fijar la línea a partir de la cual se ha de leer.

Ya que la ROM de 16 KByte es aún más potente que en el ZX 81, se encuentran las variables del sistema en otros sectores de la memoria. Las posibilidades y las variables de sistema que se ofrecen a través de PEEK y POKE, serán tratadas en el capítulo tercero.

Exteriormente hay, naturalmente, algunas diferencias entre el ZX 81 y el Spectrum. Lo más notable es el teclado, que alegrará al propietario de un ZX 81, mientras que los propietarios de otros ordenadores no tendrán para el mismo más que una sonrisa despectiva. Pero con el Spectrum Plus, este problema habrá quedado solucionado.

Las posibilidades de ampliación en el Spectrum, dentro de la empresa Sinclair, son bastante amplias. Con el Interface 1 y los Microdrives se puede ampliar el sistema de tal manera, que no debe temer comparaciones con sistemas más caros. Con los Microdrives se dispone de un sistema de memoria económico y relativamente rápido (más rápido que algunos sistemas de diskettes). A través de la conexión RS 232 es posible conectar muchos tipos de impresoras, igualmente lo es la conexión a otros ordenadores, directamente o a través de un Modem. Además existe la peculiaridad de una red local, en la cual se pueden interconectar hasta 64 Spectrum (y también Sinclair QLs). De estos temas se tratará en capítulo séptimo.

CAPITULO 2:

CONSEJOS PARA LA PROGRAMACION EN BASIC

Aspectos generales

BASIC es la abreviación de Beginners Allpurpose Symbolic Instruction Code, o sea, Código General Símbolico para Principiantes. Este lenguaje iba inicialmente dirigido hacia los novatos. Pero como todos los demás lenguajes, también BASIC se ha ido perfeccionando, y las versiones que se aplican hoy en diversos ordenadores pequeños se caracterizan por su gran flexibilidad.

Sin duda alguna, se puede aplicar el BASIC para la solución de muchos programas complejos. Cuanto más amplio sea un programa, mayor es la posibilidad de cometer errores, y mayor debería ser la preparación previa para la elaboración de un programa. Es necesario, después de pasar la fase de 'programar a lo loco', llegar a la máxima autodisciplina al programar.

Intentar programar en uno u otro sentido, sin determinación, no trae consigo más que un resultado desastroso. No hay cosa más desagradable que volver a comprobar por trigésima vez un programa propio, sin poder asimilar la lógica y encontrar los errores.

Para eludir este trabajo innecesario y evitarse al máximo las faltas en los programas es aconsejable, en casi todos los casos, detallar el problema con lápiz sobre el papel. Cada problema se puede seccionar en una serie de cuestiones parciales que, por sí solos, son más fáciles de resolver.

Programadores profesionales elaboran el programa en forma de los así denominados diagramas de flujo. Tales diagramas comienzan siempre con la casilla START y terminan con la casilla STOP. Entre estas dos casillas se encuentra la parte interesante. A todos los problemas parciales del programa se les asigna una casilla, y éstas se unen por medio de

flechas. Las flechas indican todos los posibles transcurso del programa, y las casillas contienen el problema de esta parte del programa.

Como ya se ha dicho, para programas amplios es aconsejable elaborar estos diagramas de flujo, o al menos, listar todos los problemas a solucionar. En el capítulo 4 se realizará un ejemplo de diagrama de flujo con el ejemplo del programa Mastermind. Cuando se ha realizado este trabajo previo, se puede conectar el ordenador.

Es útil programar primero una serie de líneas REM. Aquí no se debería señalar solamente las diferentes partes del programa, sino también explicar lo que representa cada variable utilizada. Si luego hiciera falta la memoria, se pueden borrar los REMs. Pero hasta que se llegue a tal punto pueden ser muy útiles.

Una propiedad del lenguaje Pascal es el así denominado concepto de procesos. Estos se definen en el programa bajo cualquier nombre y pueden ser activados en cualquier momento bajo este mismo nombre. En algunas versiones más modernas de BASIC, también se ha realizado este concepto. En el Spectrum todavía no, pero podemos conseguir algo similar cuando al principio de un programa introducimos lo siguiente:

```
LET menu = 1900
LET problema 1 = 1500
```

```
•
•
•
```

```
LET problemas = 6666
```

Cuando cada parte del programa se ha cerrado con RETURN, podemos activar cada problema con el comando GOSUB problema 1. Esto hace que la programación sea más clara. También tiene la ventaja de que al efectuar modificaciones en el programa, sólo es necesario cambiar el número de línea para un problema, y todos los demás comandos de programa siguen siendo correctos.

Generalmente se recomienda dar a los programas una determinada estructura y aplicarla siempre que sea posible. Una posibilidad para programas dirigidos por menús es empezar el menú siempre en la línea 9000. Entonces los demás problemas parciales pueden comenzar en las líneas 1000, 2000 etc. Las líneas 100 hasta 900 se utilizan para subrutinas, y las líneas 20 hasta 99 para la reserva de memoria para variables. Las primeras líneas pueden determinar el color para la Impresión, papel, etc, pero también la asignación de las líneas para los diferentes problemas parciales.

Exponemos un ejemplo que explica esta configuración:

```
1  CLEAR 65535
2  INK 0: PAPER 6: BORDER 3: CLS
3  LET margen = 100: LET falso = 150: LET color = 200
10 LET menú = 9000 : GOTO menú
20 DIM a(100) : DIM b(32)
30 DIM g$(17,3)
•
•
90 GOTO menú
```

```

100 REM margen lateral
      .
      .
140 RETURN
150 REM entrada falsa
      .
      .
190 RETURN
200 REM let it be beepen
      .
      .
290 RETURN
      .
      .
1000 REM primer concepto del menú
      .
      .
1990 GOTO menú
      .
      .
      .
      .
9000 CLS : PRINT "menú principal"
      .
      .

```

Para reservar en este programa el espacio de memoria para variables, se ha de introducir al principio el comando directo:

```
GOTO 20
```

El programa se puede salvar (SAVE) con LINE 1, y se inicia automáticamente después de ser cargado, sin borrar las variables. También se puede volver a iniciar con GOTO 1, después de una posible desconexión del sistema. Si esto se aplica en todos los programas, sólo hay que aprenderse una vez esta dirección.

Para modificar programas de manera relativamente fácil, o buscar errores, se sugiere no sólo programar de forma clara. También se pueden conseguir listados de programas de aspecto atractivo.

Esto se consigue con una cantidad suficiente de indicaciones en línea y también introduciendo color en los listados, visualizando líneas de forma parpadeante y corriendo partes del programa hacia la derecha.

Así indicaciones REM especiales pueden ser:

```
100 REM*****
o
100 REM
```

Así subrutinas cortas pueden introducirse :

```
100 REM Borrar las líneas 10-15
110   FOR I=10 TO 15
120     PRINT AT 10,I,,
130   NEXT I
140 RETURN
```

En la página 87 del manual del Spectrum se encuentran los códigos de control para los colores. Estos pueden determinar el color de una cadena en un comando PRINT, cuando se quiere prescindir de INK y PAPER. Pero también se puede operar con ellos en el listado en casi cualquier parte. De esta forma es posible recalcar los comienzos de un programa en un color rojo, cuando antes de la línea determinada se pulsan las teclas CAPS SHIFT y SYMBOL SHIFT y 2. Al final de línea hay que eliminar el color, activando el color normal de forma idéntica. De esta forma es igualmente posible visualizar líneas, incluso el listado completo, de forma intermitente.

Estos códigos se aprenden fácilmente. Primero hay que entrar en el "Modo Extendido". Entonces se pulsa una cifra de 0 a 7: preparación de un color determinado para el fondo, 8 y 9 influyen en el contraste. Si al mismo tiempo se pulsa la tecla CAPS SHIFT, entonces el color es válido para la escritura, y las cifras 8 y 9 inician o terminan la visualización intermitente.

Cuando Vd. haya confeccionado un programa y funcione, emplee un poco de tiempo en perfeccionarlo. Intente confeccionar el programa lo más atractivo posible. El color y el sonido (ó mejor el BEEP) no están sin motivo incorporados en el Spectrum y no deberían desaprovecharse. Incluso un televisor blanco y negro representa los colores en diferentes matices grises, lo que permite igualmente una composición más diferenciada de la pantalla.

Programas con letra verde ó amarilla sobre fondo negro no son sólo más profesionales, sino que esta combinación de color exige menos esfuerzo de los ojos que letra negra sobre fondo blanco. Y naturalmente, diferentes colores presentan la información más clara, sobre todo cuando el mismo color se utiliza para el mismo tipo de información.

No existe ningún programa de cierta extensión en el cual no se encuentre, en alguna parte, un pequeño error aunque es agradable que su número sea el mínimo. La mayoría de las veces se cometen los errores en situaciones extremas, como por ejemplo cuando un fichero está completo y se ha de realizar una operación determinada. Aún cuando esta operación funciona en casi todos los casos, se introducen en estas situaciones los "Bugs" (=errores de programación que aparecen precisamente en casos excepcionales).

Sobre todo, cuando otras personas hayan de utilizar sus programas, invierta más tiempo para comprobarlos. Es muy aconsejable dejar trabajar con el programa a alguien que no esté familiarizado con el mismo, y que trabaje sin ningún tipo de instrucción. Cuando uno mismo ha escrito el programa, no se le ocurren las posibilidades más sencillas como, por ejemplo, el hacer entradas falsas.

Si la capacidad de memoria no empieza a escasear, se deberían utilizar las máximas ayudas para la entrada de datos, y éstas escribirlas en la pantalla.

Ocurre a veces, que después de algún tiempo no se sabe qué es lo que se ha escrito en el programa y agradece estas ayudas.

Programar efectivamente

Todas estas reflexiones traen consigo una ocupación más o menos intensiva de la memoria. Pero también hay otras posibilidades de ser más ahorrativo con esta capacidad de memoria. Ya hemos visto un pequeño ejemplo. Supongamos que Vd. quiere borrar una línea, por ejemplo la línea 10. Esto se puede realizar con el comando:

```
PRINT AT 10,0;"
```

quiere decirse, escribiendo una cadena con 32 elementos. Con el comando

```
PRINT AT 10,0,,
```

se consigue el mismo fin y se ocupan 32 Bytes menos de memoria. Los números se memorizan en el Spectrum con 5 Bytes, palabras clave sin embargo, sólo con 1 Byte. También las variables ocupan 1 Byte. Con ello es posible dar el comando más eficaz, por ejemplo:

```
PRINT AT 10, VAL "0",,  
o PRINT AT 10,NOT PI,,
```

La indicación PI/PI está por '1', así como 'NOT PI' está por '0'. Si Vd. se fija cuantas veces se utiliza el '0' y el '1' en un programa largo, podrá calcular fácilmente cuánta capacidad de memoria se puede ahorrar en un caso de necesidad, siempre que se sustituyan estas variables numéricas.

Si se utiliza frecuentemente una dirección de salto, por ejemplo la línea donde está el menú, se gana capacidad denominando a esta línea con una variable lo más corta posible como puede ser 'm', y se escribe GOTO m en lugar de GOTO 9000.

Para ser justo hay que decir que este método es un poco más lento que la indicación directa del número, ya que el Spectrum ha de buscarse cada vez el valor para cada variable. Pero por regla general, la diferencia es mínima.

Es aconsejable utilizar los mismos nombres de variable lo más frecuente posible, ya que cada variable a definir ocupa espacio. Claro, que no se sobreesciban de esta manera las magnitudes que más tarde puedan hacer falta.

Si en un programa se han de memorizar muchos números, cuyo valor no sobrepasa 255, lo más efectivo es memorizar en lugar del número (que ha de ser entero) el carácter que corresponde a este número. Esto sólo ocupa 1 Byte en lugar de 5 , que son necesarios para cada número. A través del comando CODE se recupera el número cuando sea necesario.

Incluso hasta el número 65536 es más rentable memorizarlos a través del carácter con dos Bytes. Un Byte basta para la magnitud INT (número 256), y el segundo Byte para 'Número-256*INT(número/256). Este es justamente el método que utiliza el Spectrum para direccionar sus 65536 espacios de memoria. En el capítulo próximo utilizaremos más a menudo este sistema de direccionamiento.

Con o sin Interface 1 existe la posibilidad de utilizar junto al comando simple PRINT, igualmente el comando PRINT#, para programación más efectiva, siempre que se disponga de una Impresora.

Esto es debido a que el concepto de corrientes y canales descrito en el manual del Interface 1, está incluido ya en el Interface 1 sin ROM adicional, pero esto de forma más íntima. En el Spectrum, la entrada y la salida se realizan a través de diferentes corrientes, las cuales llegan por los así denominados 'canales' a su punto de determinación. En el ROM normal se encuentran tres corrientes diferentes, señalizadas con 1,2 y 3. También existen tres canales; uno, denominado 'k', hacia la parte inferior de la pantalla a través del cual se transmiten los comandos INPUT e Indicaciones del Spectrum. El segundo es denominado 's' y es responsable de la parte superior de la pantalla, normalmente la parte más activada. Este canal es activado con PRINT. El tercer canal es utilizado por la impresora.

Si se tiene el Interface 1, se dispone de más canales, como son el 'm' para Microdrives, 't' y 'b' para comunicación en serie y 'n' para la red. Estos pueden ser activados a través de 16 corrientes (de 0 a 15). El comando PRINT "Hola" es en realidad una abreviación de PRINT# 2, "Hola". Igualmente, el comando LPRINT "Hola" es una abreviación de PRINT#3,"Hola". Idénticamente se comporta LIST y LLIST.

Esto se puede aprovechar de forma ventajosa en muchos programas, para traspasar una información de la pantalla a la impresora. El siguiente ejemplo lo demostrará:

```

100 REM salida
110 LET salida = 2
120 FOR i=1 TO 200
130 PRINT# salida, a (i),b(i)
140 NEXT i
150 IF salida = 2 THEN INPUT "¿imprimir? S/N",m$
160 IF m$ = "S" THEN LET salida = 3: GO TO 120
170 LET salida = 2

```

Cuando se llega al bucle, el parámetro 'salida' se ha dirigido a 2, y se efectúa a través de la pantalla. En la línea 150 se cuestiona si se ha de aplicar la impresora. En este caso, la 'salida' se dirige hacia 3, por lo cual se efectúa a través de la impresora, como se deseaba.

Este método es seguro, sobre todo es muy efectivo cuando se trata de comandos PRINT muy amplios (y ahorra tiempo al introducirlo). También es posible comprobar las salidas en la pantalla antes de imprimirla y malgastar así el papel.

CAPITULO 3: PEEKs, POKES y USRS

La memoria del Spectrum

A algunos lectores el título de este capítulo les resultará un poco extraño. Los "enterados" saben, naturalmente, de qué se trata. Si Vd. no lo sabe: no tenga miedo. Después de leer este capítulo, Vd. lo sabrá.

Ahora nos ocuparemos un poco de la memoria del Spectrum. Cada espacio en la memoria tiene una dirección. Cuando se quiera saber lo que hay memorizado en un espacio determinado, por ejemplo el espacio 1000, se introduce el siguiente comando:

```
PRINT PEEK 1000
```

El Spectrum contesta entonces con un número entre 0 y 255 en la pantalla (en este caso el 9).

El comando PEEK significa: "mira qué es lo que hay en ese espacio". Si se quiere memorizar un número determinado en un espacio determinado, por ejemplo el 6, hay que introducir:

```
POKE 30000,6
```

Naturalmente sólo se puede trabajar con POKES en la memoria RAM. Si esto fuera posible con el ROM, el sistema operativo quedaría desconcertado.

Las direcciones de la memoria se han elegido de la manera más sencilla.

Empiezan con el 0 y llegan, según la versión del Spectrum, hasta 32767 (16K) o hasta 65535 (48K). Los primeros 16K, hasta la dirección 16383, están ocupados por el ROM. Aquí no es posible operar con el POKE, sino únicamente con el PEEK. Pero podemos aplicar algunas rutinas del sistema operativo, activándolo desde nuestro programa. Cómo se hace esto y dónde están memorizadas algunas rutinas interesantes, se describirá a continuación.

En los espacios del 16384 al 22527 se encuentra el fichero de presentación en pantalla, o sea, la información que se vé en la pantalla. Esto es exactamente $256 \times 192 / 8 = 6144$ Bytes. A cada matriz de 8×8 , se le puede asignar un atributo de color. Estos $32 \times 24 = 768$ Bytes se encuentran a continuación del mencionado fichero en los espacios 22528 hasta 23295 y constituyen el fichero de atributos.

256 Bytes de memoria intermedia para la impresora se encuentran en los espacios 23296 hasta 23551. A continuación de esta memoria se encuentran las variables del sistema, que apoyan al Spectrum en la organización de ficheros importantes para un programa. No todas las magnitudes están en direcciones fijas. A veces es más interesante tener una sola dirección, en la cual se encuentra la dirección de determinadas magnitudes.

Esto es similar a un viajante comercial que va continuamente de un lugar a otro. Siempre que llega a un nuevo destino, telefona a su oficina de origen y comunica su número de teléfono actual. Sus compañeros no saben nunca donde se encuentra exactamente. Pero cuando quieren contactar con él, sólo han de llamar a su oficina de origen para conseguir el número actual de teléfono.

A continuación de las variables de sistema que, como se ha dicho, dan información sobre el paradero de muchas magnitudes, se encuentran los mapas de microdrive y la información de canales. Y finalmente empieza el sector BASIC en el cual, primero se memoriza el programa y, después, las variables utilizadas en el mismo. Por encima de todo el sector disponible al BASIC, se encuentran también los símbolos User Defined Graphics.

Las variables de sistema

Las variables de sistema se encuentran en los espacios 23552 hasta 23733 en el Spectrum sin Interface, además están ocupados los espacios del 23734 hasta el 23791 con el Interface al utilizar Microdrives, con la conexión RS 232 y la red. Algunas de estas variables pueden ser activadas con PEEKS y POKES, en otros aplicando el comando POKES de forma inoportuna, se consiguen resultados sorprendentes, como por ejemplo una ruptura del sistema.

Las variables están provistas de nombres que no corresponden a los nombres en BASIC. Para alterar una variable no se puede introducir el comando: LET VAR = 0. Es necesario asignar, mediante 'POKE' el número correspondiente a la dirección correcta.

Si se trata de una variable que ocupa dos Bytes, entonces el primer Byte es el menos significativo. Si hay que escribir el número decimal "d" en la dirección "z", hay que dar el siguiente comando:

```
POKE z,d-256*INT(d/256)
POKE z+1,INT(d/256)
```

Al leer el valor correspondiente se introduce:

```
PRINT PEEK z+256*PEEK(z+1)
```

y obtiene el valor decimal como resultado.

Las variables de sistema están explicadas en el manual del Spectrum. Aquí vamos a detallar más profundamente algunas de ellas, y describir algunos efectos de los POKES.

23560 LAST K En este lugar se memoriza el código de la última tecla pulsada. Cuando Vd. pregunte por este valor en el modo directo, recibirá como respuesta siempre el 13.

- 23561 REPDEL Indica el tiempo en fracciones de 1/50 segundos antes de que la tecla pulsada sea repetida automáticamente. Al iniciar nuevamente el sistema se encuentra aquí el valor 35. Si Vd. "pokea" aquí el valor 1, resulta un efecto interesante. Este es ampliado, si Vd. asigna, mediante "POKE" igualmente un 1 en la siguiente dirección. Esto es debido a que aquí se encuentra :
- 23562 REPPER El tiempo entre dos repeticiones, igualmente en fracciones de 1/50 de segundo. Normalmente esto es un 5. Es más o menos imposible efectuar una entrada razonable cuando en 23561 y 23562 se encuentra 1. Lo único que se puede hacer es sacar el enchufe de la corriente.
- 23606 - 23607 AQUI se encuentra la dirección de inicio del juego de caracteres menos 256. El juego de caracteres comienza con un espacio libre (CHR\$ 32) y finaliza con CHR\$ 127. La cantidad de 256 se resta, ya que 32×8 es justamente 256. Esta dirección se puede modificar, mediante "POKE", de forma adecuada, para definir un propio juego de caracteres. En el capítulo 4 haremos uso de esta posibilidad, para poder visualizar 64 símbolos por línea.
- 23608 RASP Duración del sonido de emergencia. Este, por ejemplo, suena cuando la memoria está llena, y no puede admitir ninguna entrada más. Normalmente se encuentra aquí un 64. Tiene poco sentido cambiar este valor.
- 23609 PIP Duración de los clics del teclado. Normalmente se encuentra aquí un 0. En algún programa puede ser útil modificar, mediante "POKE", un número, para confirmar acústicamente una entrada.

23613 - 23614 ERR SP	Aquí se encuentra la dirección, a la cual se salta a causa de un error.
23617 MODE	Indica el modo en cual se encuentra el sistema. En el caso corriente es 0.
23626 BORDR	Color del borde multiplicado por 8. También están memorizados los atributos para la mitad inferior de la pantalla. Pokear números como el 132,176,198,etc., trae consigo buenos resultados. Lo más sencillo es realizar un bucle de 0 hasta 255 y tener cuidado de que, en cada valor, se escriba en las dos líneas inferiores. Con esto se pueden conseguir bonitos efectos.
23627 - 23628 VARS	Indica la dirección de las variables y el fin del programa BASIC.
23635 - 23636 PROG	Indica el comienzo del programa BASIC. Conjuntamente con VARS, permite calcular la longitud del programa más $256 * (\text{PEEK } 23628 - \text{PEEK } 23636) +$ $\text{PEEK } 23627 - \text{PEEK } 23635$
23659 DF SZ	Indica el número de líneas en la parte inferior de la pantalla. En el caso corriente son 2. Si aquí se asigna, mediante 'POKE', un 0, no es posible la indicación de error. Pulsando la tecla Break se origina el bloqueo del sistema y elimina la posibilidad de listar el programa.
23672 - 23674 FRAMES	Cuenta los Frames, o sea, aumenta en uno cada 20 milisegundos. Esto abre la posibilidad de llevar a cabo una medición de tiempo real, ya que el aumento sólo se interrumpe con Load o Save. La aplicación se muestra claramente en el programa reloj tipo - telediarlo.

23675 - 23675 UDG	Dirección del primer carácter gráfico propio definido. Esta dirección también puede ser encontrada con el comando PRINT USR "a". Si el espacio de memoria es ya muy reducido y ninguno de estos caracteres se ha de utilizar, esto es una posibilidad de ganar un poco de memoria a través de POKE.
23677 COORDS 3678	Estas posiciones dan las coordenadas x e y del último punto representado. A veces es útil asignar valores a éstos mediante "POKE" aún cuando PLOT consigue el mismo efecto.
23684 S POSN	Indica el número de columna de la posición de impresión. Aquí no se cuenta desde 0 como con AT, sino a partir de 33.
23685	Análogo a 23684, pero para el número de línea. Puede ser muy útil leer estas magnitudes mediante PEEK.
23692 SCR CT	Indica la cantidad de desplazamiento hacia arriba(SCROLL)antes de que con la pregunta se interrumpa el proceso. Si aquí se pokea continuamente un número mayor que 1 la pantalla no cesa de "enrollarse".
23693 ATTR P	Especifica los colores actuales. Como con 23624 es muy efectivo utilizar aquí el comando "POKE". Por ejemplo 155,156 ó 0.
23730 - 23731 RAMTOP	Indica la dirección de los últimos Bytes que están disponibles para el BASIC. También puede ser modificado con CLEAR.

23732 - 23733
PARAMT

Indica la dirección del último Byte de la memoria. Esta dirección se puede leer mediante ''PEEK'' para saber si se trata de la versión 16 ó 48KByte del Spectrum. A través de ello se puede, siempre que sea necesario, realizar una adaptación automática en un programa.

23791 COPIES

Indica la cantidad de copias que se realizan al utilizar Microdrives. Si aquí se introduce un número alto, se puede reducir el tiempo de acceso a un Fichero.

Algunos consejos y trucos

En la última parte de este capítulo vamos a señalar y resumir algunos consejos, que representan una gran ayuda para la redacción de un programa, o que pueden ser aplicados para proteger los programas contra listado por personas no autorizadas.

Como ya se ha dicho, el Spectrum puede ser programado en el código máquina. Estos programas o partes de programa son activados desde un programa BASIC a través de los comandos

```
RAND USR n  
o PRINT USR n
```

en los cuales, n significa la dirección de comienzo del programa. El final de tales programas es siempre el comando "RETURN" pero, claro está, en el código máquina.

La ROM está totalmente escrita en código máquina. Este programa contiene una cantidad de subprogramas, la mayoría de los cuales están finalizados con un comando Return. Podemos activar estas rutinas siempre que tengan el efecto deseado, y aplicarlas.

Cuando Vd. esté cansado de desconectar la corriente para efectuar un nuevo inicio del sistema, lo cual a largo plazo no es beneficioso para el Spectrum, puede sustituir esto por una llamada directa a la ROM. Lo único que tenemos que hacer realmente es iniciar el sistema total desde el principio. ¿Cómo se hace esto? Muy fácil. El comando

```
RAND USR 0
```

cumple este objetivo.

Podemos probar el principio en otra parte sin peligro, por ejemplo allí donde se borra la pantalla. Esto se realiza a partir del Byte 3435. O sea

RAND USR 3435

borra la pantalla.

Ahora bien, esto lo podíamos haber hecho fácilmente con 'CLS'. Pero no habríamos conseguido el efecto deseado. Ahora sólo queremos borrar la parte inferior de la pantalla. Esto se realiza con

RAND USR 3280

Ya que estamos en el sector de ROM que es responsable de la pantalla, nos acordamos de que el Spectrum pregunta "¿Scroll?" en un momento determinado. Eso significa que tiene que existir una rutina que realice este scrolling. Según en qué parte de la rutina entremos, conseguiremos diferentes efectos.

RAND USR 3280

realiza el ascenso una sola línea.

RAND USR 3582

realiza el ascenso una sola línea con borde

RAND USR 3652

realiza el ascenso la parte inferior de la pantalla, y

RAND USR 3583

realiza el ascenso la parte inferior de la pantalla con borde.

Por lo contrario

RAND USR 3330

realiza el ascenso hasta la línea 1.

Ya vimos en el último párrafo cómo se consigue el color total del borde y de las dos últimas líneas inferiores de la pantalla, e incluso el parpadeo. Esto se realizaba con el comando

POKE 23624,n

con valores para n desde 0 hasta 255, en lo cual, los valores más altos conseguían los efectos más espectaculares. Análogamente vale lo mismo para los colores de la pantalla cuando se introduce

POKE 23693,n

Si se quieren proteger los programas contra listados por personas no autorizadas, hay que inventarse bastantes trucos. Existen empresas completas que se ocupan de la protección del software. Y claro está, la protección ideal aún no se ha encontrado. Esto es debido a que, aunque mejoran los sistemas de protección, también mejoran los sistemas de burlar a estas protecciones.

A nadie se le ocurriría hoy día utilizar la protección única de escribir con tinta del mismo color que el papel, de manera que la escritura sea ilegible. Por otra parte podría ser una buena protección, ya que nadie cree en esto.

Por lo menos se puede introducir el signo de Copyright con el propio nombre. Si esto se efectúa en el número de línea 1, y a continuación

POKE 23756,0

la línea se transforma en línea 0 y no sale en el listado.

Una forma divertida, muy efectiva contra principiantes, y que les hace dudar sobre el buen funcionamiento de su ordenador, es la introducción de:

POKE 23606,20

o POKE 23606,111

o valores similares, ya que esto hace que la escritura sea ilegible. Esto ocurre porque la dirección para el juego de caracteres se ha "'mandado a paseo'". Si se quiere invertir este paso, hay que asignar un 0 mediante "'POKE'".

Naturalmente se puede aplicar este cambio de dirección del juego de caracteres, para definir un juego de caracteres propio. Esto se explicará detalladamente en el próximo capítulo, cuando consigamos que el Spectrum escriba 64 signos por línea en la pantalla.

Un método sencillo para evitar una salida accidental del programa consiste en cambiar el número de líneas en la parte inferior de la pantalla. En el caso normal son dos, a través de

POKE 23659,0

se ponen a 0, y ya no hay espacio para una indicación de error, como también origina un BREAK, y entoces llega el bloqueo del sistema.

Analógicamente funciona el comando

POKE 23613,0: POKE 23614,0

ya que aquí el programa origina un inicio nuevo del sistema después de una indicación de error. La única desventaja es, que todas estas variables no se conservan así como fueron introducidas.

El proceso ha de repetirse, por tanto, varias veces.

CAPITULO 4:

RUTINAS UTILES Y PROGRAMAS CORTOS

Letra a doble tamaño

Ha llegado la hora de ocuparnos de asuntos serios. Vamos a empezar con algunas rutinas que influyen en el tamaño de la letra en la pantalla.

Primero vamos a intentar representar el contenido de un campo, por ejemplo y\$, a doble tamaño. Este proceso sobresaltaría títulos o comunicados importantes en un programa.

Vamos a empezar primeramente con un sólo carácter. Primero conseguimos el código de este carácter. Este comienza en el lugar $8 * \text{CODE } y\$ + \text{PEEK } 23606 + 256 * \text{PEEK } 23607 = 8 * \text{CODE } y\$ + 15360$ y está compuesto por ocho números. Tomamos los primeros cuatro números y los introducimos dos veces consecutivas en USR "a", y los segundos cuatro números los introducimos análogamente en USR "b". Ahora sólo tenemos que imprimir estos dos caracteres, autodefinidos uno encima del otro en el lugar deseado, y ya hemos terminado.

o sea:

```
800 LET x=8*CODE y$ + 15360
810 FOR z=0 TO 3
820 LET a=PEEK (x+z)
830 LET b=PEEK (x+z+4)
840 POKE USR "a"+2*z,a:POKE USR "a"+2*z+1,a
850 POKE USR "b"+2*z+b:POKE USR "b"+2*z+1,b
860 NEXT z
```

y por ejemplo

```
900 PRINT AT 9,9;"A"  
910 PRINT AT 10,9;"B"
```

para lo cual hay que elegir los correspondientes caracteres gráficos en las dos últimas líneas.

Si ahora introducimos

```
700 INPUT y$
```

y 990 GO TO 700

seguido de RUN, podemos comprobar la rutina.

Si todo sigue su orden normal, vamos a ampliar la rutina de tal manera, que y\$ pueda contener más de un sólo carácter. Para ello formamos un bucle de la longitud de y\$:

```
750 FOR i=1 TO LEN y$
```

Debido a esto tenemos que cambiar las líneas 800,900 y 910

```
800 LET x=8*CODE y$(i)+15360  
900 PRINT AT 9,i;"A"  
910 PRINT AT 10,i;"B"
```

y añadimos

```
920 NEXT i
```

Eligiendo por ejemplo la posición de impresión xpos,ypos y xpos+1,ypos para 'Gráfico A' y 'Gráfico B' y añadimos RETURN; esto es una subrutina útil, que puede convertir muchos programas en más accesibles.

En conjunto, la rutina sería:

```
750 FOR i=1 TO LEN y$
```

```

800 LET x=8*CODE y(i)+15360
810 FOR z=0 TO 3
820 LET a=PEEK (x+z)
830 LET b=PEEK (x+z+4)
840 POKE USR "a"+2*z,a:POKE USR "a"+2*z+1,a
850 POKE USR "b"+2*z,b:POKE USR "b"+2*z+1,b
860 NEXT z
900 PRINT AT xpos,ypos+i-1;"A"
910 PRINT AT xpos+1,ypos+i-1;"B"
920 NEXT i
930 RETURN

```

La rutina es relativamente lenta, pero esto no debe tomarse como molestia. Si se quiere hacer un poco más rápida, hay que elegir la siguiente versión:

```

800 LET c=USR "a":LET d=USR "b"
810 FOR i=1 TO LEN y$
820 IF y$(i)=" " THEN PRINT AT xpos+1,ypos+i-1;" ":GO
    TO 900
830 LET x=8*CODE y$(i)+15360
840 FOR z=0 TO 3
850 POKE c+2*z,PEEK(x+z):POKE d+2*z,PEEK(x+z+4)
860 POKE c+2*z+1,PEEK (x+z):POKE d+2*z+1,PEEK(x+z+4)
870 NEXT z
880 PRINT AT xpos,ypos+i-1;"A"
890 PRINT AT xpos+1,ypos+i-1;"B"
900 NEXT i
910 RETURN

```

Aquí hacen falta menos cálculos en los bucles, de manera que esta versión es casi el doble de rápida. Compare Vd. ambas rutinas. Pero piense que hay que definir las magnitudes xpos y ypos.

Si se quieren representar caracteres aún más grandes, llegaremos pronto a los límites del BASIC. Las rutinas tardan mucho tiempo. En estos casos se debería

programar en el código máquina. En el capítulo 9 utilizaremos una rutina en código máquina, para escribir letras de cualquier tamaño. Pero Vd. puede escribir sus rutinas en BASIC, y luego traducirlas al lenguaje máquina mediante un compilador. Sobre este compilador hablaremos en el capítulo 7.

64 caracteres por línea

Después de descubrir la posibilidad de representar caracteres mayores a lo normal, vamos a intentar conseguir caracteres más pequeños sobre la pantalla. ¿Por qué esto? Bien, a veces los 32 caracteres por línea no son suficientes, o al menos son incómodos cuando hay que escribir columnas de números o mucho texto. Por algo existen en el mercado, a un elevado precio, las llamadas tarjetas de 80 columnas para algunos ordenadores.

El programa procesador de textos Tasword II es capaz de representar 64 caracteres por línea. De este programa hablaremos en el capítulo 7. Aquí vamos a desarrollar un método sencillo, para poder representar 64 caracteres por línea en nuestros propios programas. El trabajo vale la pena pues permite mostrar el doble de información que en el modo normal.

¿Como funciona todo esto? Cada carácter esta representado en el Spectrum en una matriz de 8 x 8 puntos o Pixels. Pero es posible representar los caracteres en una matriz de 4 x 8, de forma que aún son legibles. ¿Probémoslo ! Para ello utilizamos los UDGs, o sea los User defined Graphics Symbole.

Probémoslo con la letra 'A'. Esta letra se puede representar en código binario de la siguiente manera:

```
BIN 00000000
BIN 00000100
BIN 00001010
BIN 00001010
BIN 00001110
BIN 00001010
BIN 00001010
BIN 00000000
```

Seguramente reconoce Vd. la muestra de la letra. Representémoslo en la pantalla. Para ello introducimos los siguientes comandos:

```
10 DATA BIN 00000000,BIN 00000100,BIN 00001010,
    BIN 00001010,BIN 00001110,BIN 00001010,BIN 00001010,
    BIN 00000000
20 FOR i=0 TO 7
30 READ x
40 POKE USR "a"+i,x
50 NEXT i
100 CLS
110 FOR i=1 TO 300
120 PRINT "A";
130 NEXT i
140 STOP
```

En la línea 120 se introduce naturalmente gráfica A, o sea CHR\$ 144. Si Vd. introduce este pequeño programa y pulsa RUN verá 300 As estrechas sobre la pantalla. Hasta aquí bien. Esto no es nada más que un nuevo juego de caracteres como ya hemos tratado en el capítulo 3. ¿Cómo podemos juntar más los caracteres de manera que, de verdad, nos quepan más en la pantalla?

El siguiente paso es una pequeña ampliación del programa anterior. Introduzcamos lo siguiente:

```
15 DATA BIN 00000000,BIN 00100000,BIN 01010000,
    BIN 01010000,BIN 01110000,BIN 01010000,BIN 01010000,
    BIN 00000000
60 FOR i=0 TO 7
70 READ x
80 POKE USR "b"+i,x
90 NEXT i
```

```
125 PRINT "B";
```

Nuevamente la letra visualizada es el carácter gráfico. Ahora introduzca otra vez RUN y verá 600 As, utilizando una vez la parte izquierda, y otra vez la parte derecha de la matriz 8 x 8. Con esto aún no hemos llegado a la meta, pero ya estamos mucho más cerca.

Ahora tenemos que conseguir que el Spectrum ponga ambos caracteres en la misma posición, sin que se borren mutuamente. Lo último se puede realizar fácilmente a través de OVER 1. O sea, si después de cada carácter que ocupe la parte izquierda de la matriz, decrementamos en uno la posición de Print, se ha solucionado el problema. El retroceder se efectúa sencillamente con PRINT CHR\$ 8. En nuestro ejemplo tendríamos que empezar con el carácter gráfico B ya que ésta ocupa la parte izquierda de la matriz. Cambiemos el programa de la siguiente manera:

```
120 PRINT "B";CHR$ 8;  
125 PRINT OVER 1;"A";
```

Y ahora introducimos RUN. Bien, ¿que le parece?

Ya esta marcado el camino a seguir: vamos a definir un juego completo de caracteres propios, en los cuales sólo se utiliza la parte izquierda de la matriz, y otro juego completo para la parte derecha. Ambos juegos los memorizaremos en cualquier lugar por encima del RAMTOP. Cuando queramos visualizar algo con 64 caracteres por línea, ponemos el indicador sobre el juego de caracteres, que está situado en las variables de sistema 23606 y 23607, cambiando de un juego a otro, mientras que visualizamos el texto sobre la pantalla según nuestro pequeño programa.

Los caracteres que queremos representar, abarcan aquellos con los códigos de 32 hasta 143, o sea 112 caracteres.

Para ello necesitamos 8×112 números por cada juego, para ambos juegos serían 2×896 números. El juego de caracteres normal es direccionado con la posición de memoria 15360. Esto significa que en la posición de memoria 23606 hay un 0, y en 23607 un 60, ya que el resultado de $0 + 256 \times 60$ es 15360. El juego de caracteres propiamente dicho comienza en la dirección 15516, y los caracteres imprimibles empiezan 256 posiciones después.

Es aconsejable iniciar los juegos de caracteres nuevos en direcciones que sean múltiplos de 256, ya que así al conmutar, sólo hay que pokear la dirección 23607.

Supongamos por un momento que ya hemos definido los dos juegos de caracteres y ocupado las posiciones de memoria 63488 hasta 64383 y 64512 hasta 65408. Entonces aseguramos con el comando:

```
SAVE "64MC" CODE 63488,1921 o bien
SAVE "*"m";1;"64MC" CODE 63488,1921 respectivamente
```

Las direcciones de la memoria están elegidas de tal manera, que con el comando POKE 23607,247 y POKE 23607,251 se puedan activar los nuevos juegos de caracteres. Con POKE 23607,60 se puede volver de nuevo a los caracteres normales. Vamos a escribir ahora una pequeña rutina que nos represente cualquier cadena con 64 caracteres por línea sobre la pantalla.

```
5 CLEAR 63487: LOAD*"m";1;"64MC" CODE
10 LET pr64=9700
20 LET xpos=10
30 LET ypos=0
50 CLS
60 INPUT " Por favor introducir texto",z$
70 GO SUB pr64
80 PAUSE 0
90 GO TO 50
9700 PRINT AT xpos,ypos;CHR$8;
9710 FOR i=1 TO LEN z$ STEP 2
9720 POKE 23607,251
9730 PRINT z$(i);
9740 IF i=LEN z$ THEN GO TO 9790
9750 PRINT CHR$ 8;
9760 POKE 23607,247
9770 PRINT OVER 1;z$(i+1);
9780 NEXT i
9790 POKE 23607,60
9800 RETURN
```

Bien, ahora sólo nos faltan los dos juegos de caracteres y empezar. La definición de los caracteres es algo muy laborioso. El que disponga del programa Tasword II lo tiene más fácil, ya que en este programa ya hay un juego definido.

Cuando se haya cargado el programa, este juego se encuentra en las direcciones 61184 hasta 62079 y puede ser salvado fácilmente cuando se vuelva al BASIC con:

SAVE "64MC" CODE 61184,896 o bien

SAVE "*"m";1;"64MC" CODE 61184,896

Después puede ser cargado con LOAD"64MC" CODE 63488,896 en las posiciones utilizadas por nosotros. El que no disponga del programa Tasword II, tiene que echar toda la carne en el asador e introducir el código. El programa para la formación de este juego de caracteres se compone de la siguiente manera:

```
10    CLEAR 63487
20    RESTORE
32    DATA 0,0,0,0,0,0,0,0
33    DATA 0,2,2,2,2,0,2,0
34    DATA 0,5,5,0,0,0,0,0
35    DATA 0,5,7,5,5,7,5,0
36    DATA 0,2,7,4,7,1,7,2
37    DATA 0,4,4,1,2,4,1,1
38    DATA 0,2,5,2,6,11,15,0
39    DATA 0,2,4,0,0,0,0,0
40    DATA 0,2,4,4,4,4,2,0
41    DATA 0,4,2,2,2,2,4,0
42    DATA 0,0,5,2,7,2,5,0
43    DATA 0,0,2,2,7,2,2,0
44    DATA 0,0,0,0,0,2,2,4
45    DATA 0,0,0,0,7,0,0,0
46    DATA 0,0,0,0,0,6,6,0
47    DATA 0,1,1,2,2,4,4,0
48    DATA 0,7,5,5,5,5,7,0
49    DATA 0,2,6,2,2,2,7,0
50    DATA 0,2,5,1,2,4,7,0
51    DATA 0,6,1,6,1,1,6,0
52    DATA 0,1,3,5,5,7,1,0
53    DATA 0,7,4,6,1,1,6,0
```

54 DATA 0,3,4,6,5,5,2,0
55 DATA 0,7,1,2,2,4,4,0
56 DATA 0,7,5,2,5,5,7,0
57 DATA 0,2,5,5,3,1,6,0
58 DATA 0,0,0,2,0,0,2,0
59 DATA 0,0,2,0,0,2,2,4
60 DATA 0,0,1,2,4,2,1,0
61 DATA 0,0,0,7,0,7,0,0
62 DATA 0,0,4,2,1,2,4,0
63 DATA 0,2,5,1,2,0,2,0
64 DATA 3,4,2,5,2,1,6,0
65 DATA 0,2,5,5,7,5,5,0
66 DATA 0,6,5,6,5,5,6,0
67 DATA 0,2,5,4,4,5,2,0
68 DATA 0,6,5,5,5,5,6,0
69 DATA 0,7,4,6,4,4,7,0
70 DATA 0,7,4,7,4,4,4,0
71 DATA 0,2,5,4,7,5,2,0
72 DATA 0,5,5,7,5,5,5,0
73 DATA 0,7,2,2,2,2,7,0
74 DATA 0,1,1,1,5,5,2,0
75 DATA 0,5,5,6,6,5,5,0
76 DATA 0,4,4,4,4,4,7,0
77 DATA 0,5,7,7,7,5,5,0
78 DATA 0,7,5,5,5,5,5,0
79 DATA 0,2,5,5,5,5,2,0
80 DATA 0,6,5,5,6,4,4,0
81 DATA 0,7,5,5,5,7,7,1
82 DATA 0,7,5,5,6,6,5,0
83 DATA 0,3,4,2,1,1,6,0
84 DATA 0,7,2,2,2,2,2,0
85 DATA 0,5,5,5,5,5,7,0
86 DATA 0,5,5,5,5,5,2,0
87 DATA 0,5,7,7,7,7,2,0
88 DATA 0,5,5,2,2,5,5,0
89 DATA 0,5,5,5,2,2,2,0
90 DATA 0,7,1,2,2,4,7,0
91 DATA 0,5,2,5,7,5,5,0
92 DATA 0,5,0,7,5,5,7,0
93 DATA 0,5,0,5,5,5,7,0

94 DATA 0,2,7,2,2,2,2,0
 95 DATA 0,0,0,0,0,0,0,15
 96 DATA 0,2,5,4,15,4,15,0
 97 DATA 0,0,6,1,7,5,7,0
 98 DATA 0,4,4,6,5,5,6,0
 99 DATA 0,0,3,4,4,4,3,0
 100 DATA 0,1,1,3,5,5,3,0
 101 DATA 0,0,2,5,6,4,3,0
 102 DATA 0,3,4,6,4,4,4,0
 103 DATA 0,0,3,5,5,3,1,6
 104 DATA 0,4,4,6,5,5,5,0
 105 DATA 0,2,0,6,2,2,7,0
 106 DATA 0,1,0,1,1,1,5,2
 107 DATA 0,4,5,6,6,5,5,0
 108 DATA 0,4,4,4,4,4,3,0
 109 DATA 0,0,5,7,7,7,5,0
 110 DATA 0,0,6,5,5,5,5,0
 111 DATA 0,0,2,5,5,5,2,0
 112 DATA 0,0,6,5,5,6,4,4
 113 DATA 0,0,3,5,5,3,1,1
 114 DATA 0,0,3,4,4,4,4,0
 115 DATA 0,0,3,4,2,1,6,0
 116 DATA 0,2,7,2,2,2,1,0
 117 DATA 0,0,5,5,5,5,7,0
 118 DATA 0,0,5,5,5,5,2,0
 119 DATA 0,0,5,7,7,7,2,0
 120 DATA 0,0,5,5,2,5,5,0
 121 DATA 0,0,5,5,5,3,1,6
 122 DATA 0,0,3,1,2,4,7,0
 123 DATA 0,5,0,6,1,7,7,0
 124 DATA 0,5,0,2,5,5,2,0
 125 DATA 0,5,0,0,5,5,7,0
 126 DATA 0,6,5,6,5,5,6,4
 127 DATA 6,8,11,10,11,8,9,6
 128 DATA 15,9,9,9,9,9,9,15
 129 DATA 3,3,3,3,0,0,0,0
 130 DATA 12,12,12,12,0,0,0,0
 131 DATA 15,15,15,15,0,0,0,0
 132 DATA 0,0,0,0,3,3,3,3
 133 DATA 3,3,3,3,3,3,3,3

```

134 DATA 12,12,12,12,3,3,3,3
135 DATA 15,15,15,15,3,3,3,3
136 DATA 0,0,0,0,12,12,12,12
137 DATA 3,3,3,3,12,12,12,12
138 DATA 12,12,12,12,12,12,12,12
139 DATA 15,15,15,15,12,12,12,12
140 DATA 0,0,0,0,15,15,15,15
141 DATA 3,3,3,3,15,15,15,15
142 DATA 12,12,12,12,15,15,15,15
143 DATA 15,15,15,15,15,15,15,15

```

```

200 DATA 9999
250 LET n=63487
300 READ x
310 IF x>500 THEN GO TO 400
320 LET n=n+1
330 POKE n,x
390 GO TO 300

```

En este programa podemos definir también el segundo juego de caracteres.; Pero sin miedo! No hace falta introducir otra vez tantos datos. Podemos calcular estos códigos de caracteres de los datos ya existentes. En el segundo código siempre se utiliza la parte izquierda de la matriz, sólo hay que multiplicar cada número por 16, y ya hemos terminado. O sea:

```

340 POKE n+1024,16*x

```

y así sucesivamente

```

400 SAVE "64MC" CODE 63488,1921

```

O el comando correspondiente, para grabar en Microdrive.

Ahora introduzca Vd. RUN para este programa y grabe ambos juegos. Entonces puede introducir el programa antes escrito y habrá terminado.

Las posibilidades de aplicación y de modificación de la rutina 'pr64' son muy amplias. Por ejemplo, Vd. puede efectuar cada entrada con este tipo de letra. Deje Ud. volar su fantasía.

Mastermind

Ahora vamos a programar el primer pequeño juego. Se llama Mastermind. Consiste en que el Spectrum "piensa" una combinación de cinco cifras del 1 al 9, en un orden determinado. El jugador debe intentar, con un mínimo de intentos, acertar estas cinco cifras así como su orden. Después de cada intento, el Spectrum contesta por una parte el número de cifras que se han acertado, y por otra, el número de cifras que están en la posición correcta.

Antes de empezar con la programación vamos a hacer un esquema del programa, algo así como un diagrama de flujo. Los pasos a realizar son los siguientes:

INICIALIZAR, o sea, borrar pantalla, preparar memoria, etc.

"PENSAR" LOS NUMEROS

COMPONER PANTALLA

ENTRADA DEL PRONOSTICO

COMPROBACION DE LA ENTRADA

VISUALIZACION DEL PRONOSTICO

COMPROBAR POSICIONES ACERTADAS

COMPROBAR CIFRAS ACERTADAS

VISUALIZACION DE LOS ACIERTOS

COMPROBAR SI SE HA ENCONTRADO LA SOLUCION

Si se ha acertado, visualizar comentario y FIN.

Si no se ha acertado, comprobar si son demasiados
Intentos
demasiados intentos: volver a ENTRADA DE
PRONOSTICOS
pocos intentos: visualizar comentario y fin

COMENTARIOS

FIN

Empecemos con el primer punto, la INICIALIZACION.

```
4000 REM Mastermind  
4010 BORDER 6: PAPER 6: INK 0: CLS
```

Como memoria necesitamos dos campos con 5 elementos cada uno. Al primero lo llamamos 'a(5)' y aquí memorizamos las cifras elegidas por el Spectrum. Al segundo lo llamamos b(5), y contiene el pronóstico introducido. O sea

```
4020 DIM a(5): DIM b(5)
```

Además introducimos un contador, para que el juego se finalice después de máximo 15 intentos.

```
4030 LET n=0
```

Con esto se ha acabado el primer punto, y podemos dejar al Spectrum que piense 5 números.

```
4040 FOR i=1 TO 5  
4050 LET a(i)=1+INT (9*RND)
```

Con el primer número no hace falta que comprobemos si este ya ha sido introducido anteriormente, o sea

```
4060 IF i=1 THEN GO TO 4100  
4070 FOR k=1 TO i-1  
4080 IF a(i)=a(i-k) THEN GO TO 4050  
4090 NEXT k  
4100 NEXT i
```

Ahora se encuentran en el campo a(5) cinco números diferentes, como deseábamos. Ahora podemos pasar a componer la pantalla.

```
4200 PRINT "                MASTERMIND (5 de 9)"
4210 PRINT AT nn+3,0;"N        INTENTO      Pos OK      NUM OK"
```

Y ahora la entrada del pronóstico:

```
4220 INPUT"(pronóstico (por ejemplo 12345 ENTER))"; LINE b$
```

Ahora comprobamos si esta entrada es correcta.

```
4230 IF LEN b$<>5 THEN GO TO 4240
4232 LET falta=0
4234 FOR i=1 TO 5
4236 IF CODE b$(i)<49 OR CODE b$(i)>57 THEN LET falta=1
4238 NEXT i
4239 IF falta=1 THEN GO TO 4220
```

Con esto se puede visualizar el pronóstico. En realidad no es necesario entrar en el campo el contenido del String b\$. Pero en este programa seguro que no habrá problemas de capacidad de memoria y podemos hacerlo tranquilamente.

```
4240 PRINT AT nn+4,0;nn+1
4250 FOR i=1 TO 5
4260 LET b(i)=VAL b$(i)
4270 PRINT AT nn+4,2*i+2;b(i)
4280 NEXT i
4290 LET nn=nn+1
```

Ahora comprobamos si se ha acertado la solución, o cuantos números y posiciones han sido acertadas. Primero colocamos los dos contadores a cero.

```
4300 LET r1=0:LET r2=0
```

y formamos un bucle de 1 hasta 5

```
4310 FOR i=1 TO 5
4320 IF a(i)=b(i) THEN LET r1=r1+1
```

Se cuentan los números en la posición correcta. Ahora se comprueban los números acertados, sea la posición que fuere.

```
4330 FOR k=1 TO 5
4340 IF a(i)=b(k) THEN LET r2=r2+1
4350 NEXT k
4360 NEXT i
```

De esta forma los valores buscados se encuentran en r1 y r2 y pueden ser visualizados.

```
4370 PRINT AT nn+3,18;r1
4380 PRINT AT nn+3,25;r2
```

Si todos los números están en la posición correcta, hay que editar un comentario.

```
4400 IF r1=5 THEN GO TO 4500
```

Si se han realizado 15 intentos, se interrumpe el juego.

```
4410 IF nn>15 THEN GO TO 4600
```

Si no es así, vuelta a la introducción de pronóstico.

```
4420 GO TO 4220
```

Ahora veamos los comentarios:

```
4500 PRINT: PRINT "Acertado en"; nn;"intentos"
4510 IF nn<=4 THEN PRINT "Muy bien"
4520 IF nn=5 THEN PRINT "Casi un buen jugador"
4530 IF nn=6 THEN PRINT "Bastante bien"
4540 IF nn=7 THEN PRINT "Casi bien, sigue practicando"
```

```
4550 IF nn>7 AND nn<=10 THEN PRINT "Mediocre"  
4560 IF nn>10 THEN PRINT "Esto podría estar mejor!"  
4570 GO TO 4900
```

```
4600 PRINT :PRINT "Hoy no esta en forma !"
```

Y al final la pregunta, si se desea otro partida.

```
4900 PRINT FLASH 1;"¿Otra partida? S/N"  
4910 PAUSE 0  
4920 IF INKEY$="s" THEN GO TO 4000  
4930 CLS : STOP
```

Con esto hemos terminado. Ahora asegure el programa con 'LINE 1' e introduzca RUN. ¡Disfrute acertando!

El programa no se ha acertado lo máximo posible para que sea claro. Le invitamos a Vd. para mejorarlo, así como todos los programas de este libro. Se quieren publicar programas listos para funcionar, pero esto no debe impedir actuar su propia creatividad.

Para finalizar este capítulo, vamos a exponer tres listados de programas que no vamos a explicar . Son programas relativamente cortos, y con un poco de voluntad pueden ser captados rápidamente. El primero representa un reloj tipo telediarario sobre la pantalla. El programa está descrito en sus rasgos básicos en el manual de manejo del Spectrum. Aquí se ha completado. El programa se puede aplicar, naturalmente, como subrutina en otros programas.

Lo mismo sucede con el programa 'Calendario', que sigue a continuación. El programa le permite visualizar cualquier mes con sus días, sobre la pantalla.

El tercer programa listado es el de biorritmos, tan estimado por tantas personas. Naturalmente el biorritmo no tiene que ver nada con la realidad, pero nunca se sabe...

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****
8 BORDER 0: PAPER 0: INK 6: CLS
10 LET u$="1111"
20 GO TO 2000
50 INPUT "HDRA (p.ej. 0845)"; LINE u$
55 IF LEN u$ <> 4 THEN GO TO 50
60 LET errores=0
70 FOR i=1 TO 4
75 IF CODE u$(i) < 48 OR CODE u$(i) > 57 THEN LET errores=1
80 NEXT i
85 IF errores=1 THEN GO TO 50
90 IF VAL u$ > 2400 THEN GO TO 50
95 RETURN
2000 CLS : REM reloj
2100 LET e=50*(3600*VAL u$(1 TO 2)+60*VAL u$(3 TO 4))
2110 LET ss=65536: LET pp=23672
2120 LET a1=INT (e/ss)
2130 LET a2=INT ((e-a1*ss)/256)
2140 LET a3=e-a1*ss-a2*256
2150 POKE pp+2,a1: POKE pp+1,a2: POKE pp,a3
2200 CLS
2210 FOR n=1 TO 12: PRINT AT 10-10*COS (n/6*PI),16+10*SIN (i/6*PI);n: NEXT n
2220 DEF FN t():=INT ((ss*PEEK (pp+2)+256*PEEK (pp+1)+PEEK pp)/50)
2230 LET t1=FN t(): LET ts=t1
2240 LET tm=INT (t1/60): LET th=tm/60
2250 REM LET am=tm/30*PI: LET ah=th*6/PI
2251 LET am=tm/30*PI: LET ah=th/6*PI
2260 LET mx=72*SIN am: LET my=72*COS am
2270 LET hx=40*SIN ah: LET hy=40*COS ah
2300 PLOT 132,90: DRAW OVER 1;hx,hy
2310 PLOT 130,90: DRAW OVER 1;mx,my
2320 LET a=t1/30*PI
2330 LET sx=56*SIN a: LET sy=56*COS a
2340 PLOT 131,91: DRAW OVER 1;sx,sy
2350 LET t=FN t()
2360 IF t<=t1 THEN GO TO 2350
2370 PLOT 131,91: DRAW OVER 1;sx,sy
2380 IF INKEY$="u" THEN GO SUB 50: GO TO 2000
2400 IF t/60<>INT (t/60) THEN GO TO 2450
2410 PLOT 130,90: DRAW OVER 1;mx,my
2420 PLOT 132,90: DRAW OVER 1;hx,hy
2430 LET tm=tm+1: LET th=th+1/60
2440 LET ts=t: LET t1=t
2445 GO TO 2250
2450 LET t1=t
2500 GO TO 2320

```

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****
8 BORDER 0: PAPER 0: INK 6: CLS
9 RESTORE : FOR k=0 TO 7: READ d: POKE USR "n"+k,d: NEXT
k: DATA 112,14,0,104,68,68,68,0: REM genera la N en mo
de GRAPHICS
10 GO TO 100
20 LET y=0
22 FOR j=1 TO LEN m$: IF m$(j)="$" THEN LET y=y+1
23 IF y=m THEN GO TO 25
24 NEXT j
25 FOR y=1 TO 9: IF m$(j+y)<>"$" THEN NEXT y
28 LET t$=m$(j+1 TO j+y-1)
29 RETURN
30 INPUT "DIA ="; LINE a$: IF a$="" THEN GO TO 30
32 IF VAL a$<10 THEN LET a$=" "+a$
35 LET dia=VAL a$
40 RETURN
100 LET m$="$ENERO$FEBRERO$MARZO$ABRIL$MAYO$JUNIO$JULIO$AGO
STO$SETIEMBRE$OCTUBRE$NOVIEMBRE$DICIEMBRE$"
110 LET N$=" 1 2 3 4 5 6 7 8 910111213141516171819202122232
425262728293031"
120 LET d$="LunMarMieJueVieSabDom"
130 LET c$="30323232323"
140 LET m=0: LET n=0
200 INPUT "A o ? (ENTER=1985)";a$
202 IF a$="" THEN LET ty=1985: LET a$="1985"
203 IF VAL a$<1760 THEN GO TO 200
204 IF VAL a$>9999 THEN GO TO 200
208 IF a$<>" " THEN LET ty=VAL a$
210 INPUT "Mes ? (Enero=1, etc) ";a$: IF a$="" THEN GO TO
210
215 LET m=VAL a$
220 GO SUB 20
230 GO SUB 30
400 CLS : LET ly=0
402 IF ty/4<>INT (ty/4) THEN GO TO 420

```

```

404 LET ly=1
410 IF ty/100<>INT (ty/100) AND ty/400<>INT (ty/400) THEN
LET ly=0
420 LET n=CODE c$(m)-20
422 IF m=2 AND ly THEN LET n=n-1
424 LET y=0
426 FOR j=1 TO m: LET y=y+CODE c$(j)-20: NEXT j
430 IF ly AND m>1 THEN LET y=y-1
435 LET y=y+1-n+(ty-1)*365+INT ((ty-1)/4)+INT ((ty-1)/100)+
INT ((ty-1)/400)
440 LET y=y-7*INT (y/7)
450 CLS : PRINT INK 3;ty;TAB (31-LEN t$)/2;t$;TAB 28;ty
460 FOR j=1 TO LEN d$ STEP 3: PRINT AT j/3*2+2,0;d$(j TO j+
2);TAB 29;d$(j TO j+2): NEXT j
500 LET p$=n$( TO n*2)
502 FOR j=1 TO y: LET p$=" "+p$: NEXT j
520 IF date=0: LET x=5: LET y=7
530 FOR j=1 TO LEN p$ STEP 2: LET y=y+2: IF y<16 THEN GO T
O 560
540 LET y=3: LET x=x+4
560 IF p$(j TO j+1)=a$ THEN FLASH 1
565 PRINT AT y,x: INK 4;p$(j TO j+1)
566 FLASH 0
570 NEXT j
580 PRINT AT 19,0;"Proximo mes ?      Anterior ? "
581 PRINT AT 20,0;"Fecha"
590 PRINT AT 20,0;"Fecha (introducir iniciales)"
600 PAUSE 0: LET f$=INKEY$
610 IF f$="f" THEN GO TO 200
650 IF f$<>"p" THEN GO TO 670
660 LET m=m+1: LET a$="99": IF m=13 THEN LET ty=ty+1: LET
m=1
661 IF ty>9999 THEN LET ty=9999
665 GO SUB 20
669 GO TO 400
670 IF f$<>"a" THEN GO TO 680
672 LET m=m-1: IF m=0 THEN LET m=12: LET ty=ty-1: IF ty<17
60 THEN LET ty=1760
673 LET a$="99"
674 GO SUB 20
676 GO TO 400
680 GO TO 600

```

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****

10 RESTORE : FOR k=0 TO 7: READ d: POKE USR "n"+k,d: NEXT
k
13 DATA 112,14,0,104,68,68,68,0
15 REM Estas lineas generan el caracter , lo cual evita
a el engorro de no poder escribir a o como corresponde.
***** ( = GRAPHICS N) *****
17 BORDER 6: PAPER 6: INK 0: CLS
19 LET m$="303232332323"
20 PRINT TAB 7; INVERSE 1;"B I O R R I T M O"
30 INPUT "nombre= ";n$
40 PRINT AT 2,0;"de",n$
50 INPUT "a o de nacimiento : ";j$
55 IF VAL j$<=1900 THEN GO TO 50
60 PRINT "fecha de nacimiento : ";j$;"-";
70 INPUT "mes de nacimiento (1-12) ";x$
80 PRINT x$;"-";
90 INPUT "dia de nacimiento = ";t$
100 PRINT t$
110 LET q=0
120 FOR i=1 TO VAL x$-1
130 LET q=q+CODE m$(i)-20
140 NEXT i
145 LET y=VAL j$
150 IF q>59 AND y/4=INT (y/4) THEN LET q=q+1
160 LET q=q+VAL t$+INT (365.25*(y-1))
200 INPUT "fecha del biorritmo:a o =";k$
205 IF k$<j$ THEN GO TO 200
210 PRINT "fecha del biorritmo : ";k$;"-";
220 INPUT "mes= ";l$
230 PRINT l$;"-";
240 INPUT "dia";y$
250 PRINT y$
300 LET r=0
310 FOR i=1 TO VAL l$-1

```


CAPITULO 5:

UN JUEGO PARA MENTES DESPIERTAS: ADIVINAR SIMBOLOS

Explicación del juego

Después de tratar cosas más bien básicas y todo tipo de consejos, trucos así como rutinas y programas pequeños, vamos a emplearnos a fondo y redactar un programa de juego más complejo.

Como dice el título de este capítulo, no se trata de un juego bélico. Tampoco importa mucho la rapidez de reacción. Lo que se pretende es entrenar el pensamiento lógico. ¡Pero sin miedo, es muy divertido!

El problema consiste en acertar un juego de ecuaciones relacionadas entre sí, o dicho de otra forma, solucionar un sistema de ecuaciones. ¿Qué significa esto? Vamos a escribir sencillamente una ecuación:

$$5 + 6 = 11$$

Y ahora escribamos una segunda ecuación:

$$14 + 7 = 21$$

Hagamos una raya por debajo de las dos ecuaciones y sumemos los números, de lo cual resulta:

$$5 + 6 = 11$$

$$14 + 7 = 21$$

$$19 + 13 = 32$$

¿Qué es lo interesante en este asunto? Bien, cuando el programa esté listo, el Spectrum "Idea" un sistema de ecuaciones de este tipo (o más complicado), pero no nos mostrará los números, sino que representará cada cifra del 0 al 9 por un símbolo determinado. El jugador debe ahora encontrar la cifra correspondiente a cada símbolo.

Ya que para la composición de las ecuaciones y su asignación de los símbolos utilizamos el generador aleatorio del Spectrum, prácticamente ningún juego será idéntico a otro. Al principio se deseará un poco de apoyo por parte del ordenador, pero este irá disminuyendo, a medida que se va adquiriendo práctica.

Desarrollo del programa

Para el programa son necesarias las siguientes rutinas:

1. Portada con explicación del juego
2. Reservar memoria para todos los campos (vectores) posibles.
3. Componer sistema de ecuaciones
4. Asignación de símbolos a los números
5. Composición de la pantalla
6. Introducción de pronósticos, modificaciones, y edición de ayudas del ordenador
7. Comprobación, si se ha encontrado la solución
8. Final de juego

Comenzemos con el primer punto, la composición de la portada.

```
100 PAPER 6: BORDER 3: INK 0
200 CLS: PRINT TAB 8; "ADIVINAR SIMBOLO"
210 PRINT" Se trata de solucionar un problema
matematico y adivinar"
220 PRINT" que simbolo corresponde a que numero
(de 0 a 9)"
230 PRINT" Simbolos diferentes corresponden a diferentes
numeros."
```

Ponga Vd. atención a los espacios libres al introducir este texto, para que la portada presente un aspecto adecuado.

Ahora vamos a determinar el funcionamiento del juego, o sea como se han de efectuar las entradas e indicarlo en la portada.

Para que el problema sea más claro, vamos a representar por debajo de los símbolos del sistema de ecuaciones, todos los símbolos en una línea horizontal. Por debajo de esta línea se podrá mover un pequeño arterisco intermitente de izquierda a derecha que podrá situarse por debajo de cada símbolo. Si se introduce un número, éste aparecerá por encima de cada símbolo correspondiente en las ecuaciones.

Añadimos a la portada:

```
240 PRINT "Para introducir un numero para un simbolo, hay
    que posicionar"
250 PRINT "el arterisco (*) intermitente por debajo del
    simbolo correspondiente"
260 PRINT "e introducir el numero."
270 PRINT "L= * hacia la izquierda, R= * hacia la derecha"
280 PRINT "***  otras posibilidades  ***"
290 PRINT "D = borra el anterior pronostico"
300 PRINT "V = admite el valor para el simbolo por encima
    del arterisco"
310 PRINT "S = solucion del problema"
320 PRINT "N = nueva partida"
330 PRINT "E = final del juego"
340 PRINT " "; FLASH 1; "por favor pulsar tecla"
350 IF INKEY$=" " THEN GO TO 350
360 BEEP.03,7
370 CLS : PRINT FLASH 1; AT 10,9;"esperar por favor";AT 9,9;
    " ";AT 11,9;"
```

Si Vd. ha introducido todo esto bien, la pantalla no debería estar llena después del comando RUN, o sea no debe surgir la pregunta ¿SCROLL? Si este fuera el caso, corríjalo para que quepa totalmente en una página.

Ahora tenemos que solucionar nuestro primer pequeño problema: tenemos que redactar el sistema de ecuaciones. En total surgen 9 números, de los cuales sólo podemos elegir algunos libremente, o sea a través del generador aleatorio del Spectrum. Vamos a memorizar estos números en el campo a(i). Vamos a dejar esto claro:

$$\begin{array}{rccccccc} a(1) & + & a(2) & = & a(3) & & \\ + & & + & & + & & \\ a(4) & + & a(5) & = & a(6) & & \\ = & & = & & = & & \\ a(7) & + & a(8) & = & a(9) & & \end{array}$$

El mayor número posible será el 9999, para que tengamos suficiente espacio en la pantalla. De estos números, a(1), a(2), a(4) y a(5) son libremente elegibles, pero a(2), a(4) y a(5) no pueden ser muy elevados, para que en las sumas no resulte ningún número de cinco cifras.

Elijamos el siguiente esquema:

```
1010 LET a(1) = 1+INT(5000*RND)
1020 LET a(2) = 1+INT(1000*RND)
1030 LET a(4) = 1+INT(1000*RND)
1040 LET a(3) = a(1)+a(2)
1050 LET a(5) = 1+INT((8000-a(1))*RND)
```

Y el resto está claro:

```
1060 LET a(6) = a(4) +a(5)
1070 LET a(7) = a(1) +a(4)
1080 LET a(8) = a(2) +a(5)
1090 LET a(9) = a(3) +a(6)
```

Podemos añadir una condición de seguridad:

```
1100 IFA(9)>9999 THEN GO TO 1010
```

En las magnitudes a(1) hasta a(9) se encuentran ahora los números deseados. Esto ha sido más fácil de lo que se suponía. Ahora podemos dificultar el problema permitiendo no sólo adiciones , sino también sustracciones. Por ejemplo, podemos componer el siguiente sistema:

$$a(1) + a(2) = a(3)$$

$$+ \quad - \quad +$$

$$a(4) - a(5) = a(6)$$

$$= \quad = \quad =$$

$$a(7) + a(8) = a(9)$$

Esto se soluciona de forma similar al primer ejemplo, pero sin duda trae consigo alguna problemática. Dejemos que el generador aleatorio decida la cuestión a plantear, y redactemos este segundo problema.

Primero introducimos:

```
1000 LET s=0: IF RND<.5 THEN GO TO 1200
```

De esta forma, cada posibilidad tiene la misma probabilidad. Sigamos con la línea 1200

```
1130 GO TO 1300
```

```
1200 LET s=1
```

Cuando s=0, tenemos el primero, cuando s=1, el segundo problema. Ahora procedamos como en el primer problema, pero poniendo atención para que ningún número calculado sea menor de 0. Esto se consigue así:

```
1210 LET a(1) = 1+INT (5000*RND)
```

```
1220 LET a(2) = 100+INT (900*RND)
```

```
1230 LET a(4) = 100+INT (900*RND)
```

```
1235 LET a(3) = a(1)+a(2)
```

```
1240 LET a(7) = a(1)+a(4)
```

```
1245 LET a(5) = 1+INT (1000*RND)
```

```
1250 IF a(5)>=a(2) OR a(5)>=a(4) THEN GO TO 1245
```

```

1260 LET a(8) = a(2)-a(5)
1265 LET a(6) = a(4)-a(5)
1270 LET a(9) = a(3)-a(6)

```

Si ahora introducimos

```
50 DIM a(9)
```

hemos terminado esta parte del programa.

Ahora tenemos que proporcionarnos los símbolos. En el Spectrum disponemos previamente de ocho símbolos gráficos, pero nos faltan dos para poder disponer de uno para cada cifra del 0 al 9. Aquí nos ayudan los UDGs, o sea los símbolos USER DEFINED GRAPHICS. Como está descrito en el manual, podemos definir otros símbolos. Nosotros elegimos triángulos. Si elegimos los triángulos espectacularmente simétricos, tenemos menos trabajo:

```

20 DATA 128,192,224,240,248,252,254,255
30 FOR i=0 TO 7: READ xn: POKE USR "A" + i,xn:
    POKE USR "B" + 7-i,xn:NEXT i

```

Ahora almacenamos todos los símbolos en el campo a\$ con

```
80 LET a$ = "▲ ■ □ ▣ ▤ ▥ ▦ ▧ ▨ ▩"
```

Cada uno de los 9 números a(i) debe ser representado ahora por símbolos. Para ello generamos los números del 0 al 9 en un orden 'casual':

```

1300 FOR i=1 to 10
1310 LET z(i) = INT (10*RND)

```

El primer número puede ser uno cualquiera, en todos los demás hay que comprobar si el número siguiente ya existe.

```
1320 IFT i=1 THEN GO TO 1360
```

```

1330 FOR k=1 TO i-1
1340 IF ABS (z(k)-z(i))<=0.5 THEN GO TO 1310
1350 NEXT k
1360 LET z$(i)=a$(z(i+1))
1370 NEXT i

```

En el campo z(i) se encuentran ahora los números del 0 al 9 en un orden aleatorio y en la cadena z\$(i) se encuentran los símbolos gráficos. z\$(i) es un campo como a\$, sólo que en a\$ se encuentran los números siempre en el mismo orden, mientras que en z\$ están en un orden aleatorio.

El siguiente problema es colocar cada uno de estos números en su lugar adecuado para que cada cifra de los mismos a(i) sea sustituida por un símbolo. Tenemos que fijarnos en que los símbolos en las cadenas que representan los números, empiecen por el margen derecho. Esto parece complicado, pero simplemente significa que las 'unidades' estén bajo las 'unidades', las 'decenas' bajo las 'decenas', etc.

Para conseguir esto transformamos los números a(i) en cadenas. Estas cadenas se copian luego a cadenas w\$ alineadas al margen derecho, y hemos conseguido lo que queríamos.

```

1380 FOR i=1 TO 9
1390 LET q$(i) = STR$ a(i)
1400 LET l=LEN (STR$ (a(i)))
1410 FOR k=1 TO l
1420 LET w$(i,5-k)=z$(VAL (q$(i,l+1-k))+1)
1430 NEXT k
1440 NEXT i

```

Tenemos que dimensionar aún los campos utilizados:

```

60 DIM z(10): DIM z$(10)
70 DIM q$(9,5): DIM w$(9,4)

```

Después de estos trabajos previos, podemos visualizar el problema en la pantalla:

```
2000 CLS : PRINT"-----PROBLEMA-----"
2010 PRINT AT 17,0;"-----"
2020 FOR i=1 TO 10: PRINT AT 19,3*i-1;z$(z(i)+1): NEXT i
2030 PRINT AT 13,0;"--"
2040 PRINT AT 21,0;"--"
2050 FOR i=0 TO 13: PRINT AT i,0;"--"; AT i,31;"--: NEXT i
2060 FOR i"17 TO 21: PRINT AT i,0;"--"; AT i,31; "--: NEXT i
2070 FOR i=1 TO 9
2080 LET k=0: LET n=0
2090 IF i=2 OR i=5 OR i=8 THEN LET k=1
2100 IF i=3 OR i=6 OR i=9 THEN LET k=2
2110 IF a(i)>9 THEN LET n=1
2120 IF a(i)>99 THEN LET n=2
2130 IF a(i)>999 THEN LET n=3
2140 LET y(i) = 6-2*n+10*k
2150 LET f(i)=10*k
2160 LET LET x(i)=2+4*INT ((i-1)/3)
2170 NEST i
2180 FOR i=1 TO 9
2190 FOR k=1 TO 4
2200 PRINT AT x(i)+1,f(i)+2*k;w$(i,k)
2210 NEXT k
2220 NEXT i
```

Con esto hemos colocado todos los símbolos en su lugar correcto, y sólo nos queda introducir los símbolos de restar y sumar.

```
2230 PRINT AT 3,10; "+"; AT 3,20;"="
2240 PRINT AT 7,10; "+"; AT 7,20;"="
2250 PRINT AT 11,10; "+"; AT 11,20;"="
2260 PRINT AT 5,6; "+";AT 5,16; "+"; AT 5,26; "+"
2270 PRINT AT 9,6; "=";AT 9,16; "=" AT 9,26; "="
```

Para el caso de que también se encuentren sustracciones, hay que añadir:

```
2280 IS s>=1 THEN PRINT AT 7,10;"-";AT 5,16;"-"
```

La pantalla se ha compuesto de tal manera que quede aún espacio para explicaciones.

```
3000 REM Cursor
3010 LET x=21
3020 LET y=2
3100 PRINT FLASH 1;AT x,y; ""
3200 PRINT AT 14,0;"D=* derecha      I = * izquierda"
3210 PRINT AT 15,0;"B=borra,          V = Dar valor"
3220 PRINT AT 16,0;"S=solucion,0=otra partida,F=fin"
3300 PAUSE 0: LET m$ = INKEY$
3310 IF m$ = "i" THEN BEEP . 1,1: GO TO 3700
3320 IF m$ = "d" THEN BEEP . 1,1: GO TO 3800
3330 IF m$ = "f" THEN GO TO 6000
3340 IF m$ = "o" THEN GO TO 1
3350 IF m$ = "v" THEN BEEP . 3,5: GO TO 3800
3360 IF m$ = "s" THEN GO TO 4ppp
3370 IF m$ = "b" THEN BEEP . 3,5: GO TO 5000
3380 IF CODE m$<58 AND CODE m$>47 THEN GO TO 5000
3390 BEEP .05,5: BEEP . 1,1
3400 GO TO 3300
```

Con esto hemos fijado las restantes rutinas. A partir de la línea 3700, el cursor se mueve hacia la derecha, y a partir de la línea 3800 hacia la izquierda. A partir de la línea 6000 se finaliza el juego, a partir de la línea 7000 el ordenador da un valor para apoyar al jugador, a partir de la línea 4000 se indica la solución del problema, y a partir de la línea 5000 se efectúa la introducción o el borrado de un pronóstico.

```
3700 IF y>27 THEN GO TO 3300
3710 LET y=y+3
3720 PRINT AT x,y-3; ""
3730 PRINT FLASH 1;AT x,y; ""
3740 FOR I=1 TO 50: NEXT I
3750 GO TO 3300
```

```
3800 IF y<3 THEN GO TO 3300
3810 LET y=y-3
3820 PRINT AT x,y+3; ""
```

```

3830 PRINT FLASH 1;AT x,y;"*"
3840 FOR i=1 TO 50: NEXT i
3850 GO TO 3300

```

El final del programa es bastante sencillo:

```

6000 PRINT AT 21,6;"Otra partida? S/N"
6010 PAUSE 0: LET m$=INKEY$
6020 IF m$="s" THEN GO TO 1
6030 CLS : STOP

```

La solución del problema también se puede indicar bastante fácilmente:

```

4000 REM solución
4010 FOR i=1 TO 20: BEEP .05,i: BEEP .05,-4: NEXT i

```

Esta línea no es necesaria pero queda muy bien.

```

4020 FOR i=1 TO 9
4030 FOR k=1 TO 4
4040 IF q$(i,k)= " " THEN GO TO 4060
4050 PRIN AT x(i),y(i)+2*k;q$(i,k)
4060 NEXT k
4070 NEXT i
4080 FOR i=1 TO 10
4090 PRINT AT 18,3*i-1;z(i)
4100 LET g(i)=z(i)
4110 NEXT i
4120 PRINT AT 14,0; PAPER 2,,,,,
4130 PRINT FLASH 1; AT 15,6;"Esta es la solución"
4140 GO TO 6000

```

La línea 4100 es algo extraña. Queremos utilizar el campo g(i) para contabilizar los símbolos acertados, de forma que en cada pronóstico se pueda comprobar si ya se ha encontrado la solución. Para ello escribimos al principio en todos los g(i) un número mayor que 9, por ejemplo 22.

También puede Vd. escribir el 432 si lo prefiere.

```
100 FOR I=1 TO 10: LET g(i) = 22: NEXT I
```

Se salta a la línea 5000 cuando se va a hacer una introducción o a borrar la misma.

```
5000 REM introducir pronóstico
5010 LET j$=m$
5020 IF m$=d"THEN LET j$=" ": LET g(INT (y/3)+1)=22:
      GO TO 5200
5030 LET G(INT (y/3)+1)=VAL j$
5040 BEEP .04,1: BEEP .04,8
5200 FOR i=1 TO 9
5210 FOR k=1 TO 4
5300 IF w$(i,k)=z$(z(y/3)+1)THEN LET c$(i,k)=j$
5310 NEXT k
5320 NEXT i
```

Esto es algo complicado, pero funciona. En c\$ se encuentra todo lo que ha sido acertado. Ahora sólo tenemos que ponerlo al día.

```
5500 PRINT AT 18,y;j$
5600 FOR i=1 TO 9
5610 FOR k=1 TO 4
5620 PRINT AT x(i),f(i)+2*k;c$(i,k)
5630 NEXT k
5640 NEXT i
```

Ahora se comprueba si se ha encontrado la solución:

```
5800 FOR i=1 TO 10
5810 IF g(i) <> z(i) THEN GO TO 3300
5820 NEXT i
5830 PRINT AT 14,0; PAPER 2,,,,,,
5840 PRINT FLASH 1; AT 15,8, "H.A..G.A.N.A.D.O."
```

En este caso se sigue automáticamente con la línea 6000 con la pregunta ''Otra partida?'. Lo que aún nos falta, es la rutina la cual indica el número correspondiente a un símbolo para ayudar al jugador, si así lo desea.

```
7000 REM indicar un número
7010 PRINT AT 18,y;z(INT (y/3+1))
7020 LET g(INT ((y/3)+1))=z(INT ((y/3)+1))
7030 FOR i=1 TO 9
7040 LET j=0: IF a(i) <1000 THEN LET j=1
7050 IF a(i) <100 THEN LET j=2
7060 IF a(i) <10 THEN LET j=3
7070 FOR k=1 TO 4
7080 IF w$(i,k)=z$(z(y/3)+1) THEN LET c$(i,k)=q$(i,k-j)
7090 PRINT AT x(i),y(i)+2*(k-j);c$(i,k)
7100 NEXT k
7110 NEXT i
7120 GO TO 3000
```

Las líneas del principio, en las cuales se dimensionan los campos, han de ser ampliadas de la siguiente forma:

```
50 DIM a(9): DIM x(9): DIM y(9): DIM f(9): DIM g(10)
60 DIM z$(10): DIM z(10)
70 DIM q$(9,5): DIM w$(9,4): DIM c4(9,4)
```

Al final añadimos:

```
10 CLEAR: RESTORE: FOR i=1 TO 20: BEEP .5/i,i: NEXT i
```

y hemos terminado. Al final del capítulo veremos el listado total del programa sin interrupciones.

Y ahora ¡que lo pase bien adivinando!

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****

10 CLEAR : RESTORE : FOR i=1 TO 20: BEEP .5/i,i: NEXT i
20 DATA 128,192,224,240,248,252,254,255
30 FOR i=0 TO 7: READ xn: POKE USR "a"+i,xn: POKE USR "b"+
7-i,xn: NEXT i
40 PAPER 6: BORDER 3: INK 0
50 DIM a(9): DIM x(9): DIM y(9): DIM f(9): DIM g(10)
60 DIM z$(10): DIM z(10)
70 DIM q$(9,5): DIM w$(9,4): DIM c$(9,4)
80 LET a$=" "
81 REM a$(1),a$(2),...,a$(10) son, respectivamente CHR$ de
142, 141, 139, 137, 134, 135, 143, 144, 140 y 145
90 LET s=0
100 FOR i=1 TO 10: LET g(i)=22: NEXT i
200 CLS : PRINT " A D I V I N A R S I M B O L O "
210 PRINT " Se trata de solucionar un problema matemati-
co, y adivinar"
220 PRINT "que simbolo corresponde a una cifra (de 0 a 9)
"
230 PRINT "Simbolos diferentes corresponden a diferentes cif-
ras."
240 PRINT " Para introducir una cifra para un simbolo hay
que posicionar"
250 PRINT "el asterisco intermitente debajo del simbolo cor-
respondiente e"
260 PRINT "introducir la cifra."
270 PRINT "I=# a la izq. D=# a la der."
280 PRINT "**** Otras posibilidades ****"
290 PRINT "B = Borra pronostico anterior"
300 PRINT "V = Asigna el valor al simbolo situado sobr-
e el asterisco"
310 PRINT "S = Solucion del problema"
320 PRINT "N = Nueva partida"
330 PRINT "F = Final del juego"
340 PRINT " "; FLASH 1;"POR FAVOR PULSE UNA TECLA"

```

```

350 IF INKEY$="" THEN GO TO 350
360 BEEP .03,7
370 CLS : PRINT FLASH 1;AT 10,6;"ESPERE POR FAVOR";AT 9,6;
"      ";AT 11,6;"      "
1000 IF RND<.5 THEN GO TO 1200
1010 LET a(1)=1+INT (5000*RND)
1020 LET a(2)=1+INT (1000*RND)
1030 LET a(4)=1+INT (1000*RND)
1040 LET a(3)=a(1)+a(2)
1050 LET a(5)=1+INT ((8000-a(1))*RND)
1060 LET a(6)=a(4)+a(5)
1070 LET a(7)=a(1)+a(4)
1080 LET a(8)=a(2)+a(5)
1090 LET a(9)=a(3)+a(6)
1100 IF a(9)>9999 THEN GO TO 1010
1130 GO TO 1300
1200 LET s=1
1210 LET a(1)=1+INT (5000*RND)
1220 LET a(2)=100+INT (900*RND)
1230 LET a(4)=100+INT (900*RND)
1235 LET a(3)=a(1)+a(2)
1240 LET a(7)=a(1)+a(4)
1245 LET a(5)=1+INT (1000*RND)
1250 IF a(5)>a(2) OR a(5)>a(4) THEN GO TO 1245
1260 LET a(8)=a(2)-a(5)
1265 LET a(6)=a(4)-a(5)
1270 LET a(9)=a(3)+a(6)
1300 FOR i=1 TO 10
1310 LET z(i)=INT (10*RND)
1320 IF i=1 THEN GO TO 1360
1330 FOR k=1 TO i-1
1340 IF ABS (z(k)-z(i))<=0.5 THEN GO TO 1310
1350 NEXT k
1360 LET z$(i)=a$(z(i)+1)
1370 NEXT i
1380 FOR i=1 TO 9
1390 LET q$(i)=STR$ a(i)
1400 LET l=LEN (STR$ (a(i)))
1410 FOR k=1 TO l
1420 LET w$(i,5-k)=z$(VAL (q$(i,l+1-k))+1)
1430 NEXT k
1440 NEXT i

```

```

2000 CLS : PRINT "          P R O B L E M A          "

2010 PRINT AT 17,0;"          "
2020 FOR i=1 TO 10: PRINT AT 19,3*i-1;z*(z(i)+1): NEXT i
2030 PRINT AT 13,0;"          "
2040 PRINT AT 21,0;"          "
2050 FOR i=0 TO 13: PRINT AT i,0;" ";AT i,31;" ": NEXT i
2060 FOR i=17 TO 21: PRINT AT i,0;" ";AT i,31;" ": NEXT i
2070 FOR i=1 TO 9
2080 LET k=0: LET n=0
2090 IF i=2 OR i=5 OR i=8 THEN LET k=1
2100 IF i=3 OR i=6 OR i=9 THEN LET k=2
2110 IF a(i)>9 THEN LET n=1
2120 IF a(i)>99 THEN LET n=2
2130 IF a(i)>999 THEN LET n=3
2140 LET y(i)=6-2*n+10*k
2150 LET f(i)=10*k
2160 LET x(i)=2+4*INT ((i-1)/3)
2170 NEXT i
2180 FOR i=1 TO 9
2190 FOR k=1 TO 4
2200 PRINT AT x(i)+1,f(i)+2*k;w$(i,k)
2210 NEXT k
2220 NEXT i
2230 PRINT AT 3,10;"+";AT 3,20;"="
2240 PRINT AT 7,10;"+";AT 7,20;"="
2250 PRINT AT 11,10;"+";AT 11,20;"="
2260 PRINT AT 5,6;"+";AT 5,16;"+";AT 5,26;"+"
2270 PRINT AT 9,6;"=";AT 9,16;"=";AT 9,26;"="
2280 IF s>=1 THEN PRINT AT 7,10;"-";AT 5,16;"-"
3000 REM cursor
3010 LET x=21
3020 LET y=2
3100 PRINT FLASH 1;AT x,y;"*"
3200 PRINT AT 14,0;"I = * izquierda      D = * derecha"
3210 PRINT AT 15,0;"B = borrar          V = dar valor"
3220 PRINT AT 16,0;"S=solucion  N=nueva part.  F=fin"
3300 PAUSE 0: LET m$=INKEY$
3310 IF m$="d" THEN BEEP .1,1: GO TO 3700
3320 IF m$="i" THEN BEEP .1,1: GO TO 3800
3330 IF m$="f" THEN GO TO 6000
3340 IF m$="n" THEN GO TO 1

```

```

3350 IF m$="v" THEN BEEP .3,5: GO TO 7000
3360 IF m$="s" THEN GO TO 4000
3370 IF m$="b" THEN BEEP .3,5: GO TO 5000
3380 IF CODE m$<58 AND CODE m$>47 THEN GO TO 5000
3390 BEEP .05,5: BEEP .1,1
3400 GO TO 3300
3700 IF y>27 THEN GO TO 3300
3710 LET y=y+3
3720 PRINT AT x,y-3;" "
3730 PRINT FLASH 1;AT x,y;"*"
3740 FOR i=1 TO 50: NEXT i
3750 GO TO 3300
3800 IF y<3 THEN GO TO 3300
3810 LET y=y-3
3820 PRINT AT x,y+3;" "
3830 PRINT FLASH 1;AT x,y;"*"
3840 FOR i=1 TO 50: NEXT i
3850 GO TO 3300
4000 REM solucion
4010 FOR i=1 TO 20: BEEP .05,1: BEEP .05,1: BEEP .05,-4: NEX
T i
4020 FOR i=1 TO 9
4030 FOR k=1 TO 4
4040 IF q$(i,k)=" " THEN GO TO 4060
4050 PRINT AT x(i),y(i)+2*k;q$(i,k)
4060 NEXT k
4070 NEXT i
4080 FOR i=1 TO 10
4090 PRINT AT 18,3*i-1;z(i)
4100 LET g(i)=z(i)
4110 NEXT i
4120 PRINT AT 14,0; PAPER 2,,,,,,
4130 PRINT FLASH 1;AT 15,6;"ESTA ES LA SOLUCION:"
4140 GO TO 6000
5000 REM introducir pronostico
5010 LET j$=m$
5020 IF m$="b" THEN LET j$=" ": LET g(INT (y/3)+1)=22: GO T
O 5200
5030 LET g(INT (y/3)+1)=VAL j$
5040 BEEP .04,1: BEEP .04,8
5200 FOR i=1 TO 9
5210 FOR k=1 TO 4

```

```

5300 IF w$(i,k)=z$(z(y/3)+1) THEN LET c$(i,k)=j$
5310 NEXT k
5320 NEXT i
5500 PRINT AT 18,y;j$
5600 FOR i=1 TO 9
5610 FOR k=1 TO 4
5620 PRINT AT x(i),f(i)+2*k;c$(i,k)
5630 NEXT k
5640 NEXT i
5650 PRINT AT 13,6;" "
5800 FOR i=1 TO 10
5810 IF g(i)<>z(i) THEN GO TO 3300
5820 NEXT i
5830 PRINT AT 14,0; PAPER 2,,,,,
5840 PRINT FLASH 1;AT 15,8;"H A G A N A D O"
6000 PRINT AT 21,6;"Otra partida ? (s/n)"
6010 PAUSE 0: LET m$=INKEY$
6020 IF m$="s" THEN GO TO 1
6030 CLS : STOP
7000 REM introducir una cifra
7010 PRINT AT 18,y;z(INT (y/3+1))
7020 LET g(INT ((y/3)+1))=z(INT ((y/3)+1))
7030 FOR i=1 TO 9
7040 LET j=0: IF a(i)<1000 THEN LET j=1
7050 IF a(i)<100 THEN LET j=2
7060 IF a(i)<10 THEN LET j=3
7070 FOR k=1 TO 4
7080 IF w$(i,k)=z$(z(y/3)+1) THEN LET c$(i,k)=q$(i,k-j)
7090 PRINT AT x(i),y(i)+2*(k-j);c$(i,k)
7100 NEXT k
7110 NEXT i
7120 GO TO 3300

```

-----P R O B L E M A-----

$$\begin{array}{r}
 \begin{array}{cccc} 2 & 3 & 3 & 3 \end{array} + \begin{array}{ccc} 5 & 3 & 9 \end{array} = \begin{array}{cccc} 3 & 1 & 7 & 2 \end{array} \\
 \begin{array}{ccc} & 5 & 5 \end{array} + \begin{array}{ccc} 1 & 4 & 1 \end{array} = \begin{array}{ccc} 5 & 2 & 4 \end{array} \\
 \begin{array}{cccc} 3 & 2 & 9 & 8 \end{array} + \begin{array}{ccc} 3 & 9 & 8 \end{array} = \begin{array}{cccc} 3 & 6 & 9 & 6 \end{array}
 \end{array}$$

Esta es la solucion

$$\begin{array}{cccccccccc}
 5 & 7 & 0 & 8 & 2 & 9 & 4 & 3 & 6 & 1
 \end{array}$$

-----Otra partida? (s/n)-----

ADIVINAR SIMBOLO

Se trata de solucionar un problema matematico, y adivinar que simbolo corresponde a una cifra (de 0 a 9).

Simbolos diferentes corresponden a diferentes cifras.

Para Introducir una cifra para un simbolo hay que posicionar el asterisco Intermitente debajo del simbolo correspondiente e Introducir la cifra.

I =  a la Izq. D =  a la der.

**** Otras posibilidades ****

B = Borra pronostico anterior

V = Asigna el valor al simbolo situado sobre el asterisco

S = Solucion del problema

N = Nueva partida

F = Final del juego

POR FAVOR PULSE UNA TECLA

CAPITULO 6:
SPECTRUM ESPECTACULAR:
DIA Y NOCHE EN LA TIERRA

Desarrollo del programa

En este capítulo se va a elaborar un programa que en realidad es superfluo y no es de ninguna utilidad. Además es difícil introducirlo para que funcione ordenadamente. ¿Por qué todo este trabajo?. Bien; cuando esté terminado demuestra las cualidades gráficas del Spectrum y nos proporciona además un mapa mundi sobre la pantalla.

Para que el programa sea un poco interesante y divertido, vamos a introducir un reloj, e indicar las zonas de día y noche en la tierra. Estas zonas deben de estar sincronizadas con el reloj y adaptarse automáticamente a la duración del día durante el invierno y el verano.

Cuando hayamos conseguido esto, el programa no resultará tan aburrido. Además, el mapa mundi puede memorizarse con 'SAVE name SCREEN\$' y utilizarse como portada.

De manera práctica representaremos los mares en color azul, las zonas de día en amarillo y las de noche en negro. Para ello tenemos que solucionar un pequeño problema. El Spectrum dispone de una resolución gráfica bastante elevada, pero en cada posición solo pueden representarse dos colores: uno para el fondo y otro para el primer plano. En nuestro caso, en las 'costas' en la transición de día y noche pueden interferirse tres colores: el mar azul, tierra negra y amarilla.

Però si actualizamos estas zonas cada hora, y nos lo arreglamos de forma que una hora corresponda a una posición en la pantalla, o sea 8 Pixels, podemos solucionar el problema. En este caso los tres colores se juntan justamente en el límite de dos posiciones.

Los datos para el mapa mundi los introduciremos por medio de comandos DATA y los leeremos por medio de READ en el programa. Para ello, dibujamos sobre un mapa mundi adecuado un pautado de 102 por 192 cuadrados. Después buscamos en cada fila de cuadrados, por ejemplo en dirección Sur-Norte, aquéllos en los cuales el mar límite con la tierra y los introducimos por medio de DATA. Cuando una fila está terminada, introducimos dos ceros, de manera que el programa sabe que hay que pasar a la siguiente fila.

Naturalmente, Vd no tiene que pasar por esta tortura. Únicamente está explicada para que Vd. sepa como se han elaborado los datos en el listado de DATA.

Después de haber introducido los datos (¡diviértase tecleando!), podemos dedicarnos al programa propiamente dicho. Primero nos saltamos los DATAs, que comienzan en la línea 132, en introducimos:

```
10 GO TO 8000
```

Para que el título sobresalga, podemos utilizar la letra grande - rutina del capítulo 4 - que introducimos a partir de la línea 20. Entonces seguimos:

```
8000 CLS : LET y$='Día y noche en la tierra':LET xpos=0:  
      LET ypos=2: GO SUB 20  
8010 PRINT AT 9,0;'Mientras el programa trabaja, significan  
      v adelantar reloj      h parar el reloj      n normal  
      u corrección del reloj e final"
```

```

8020 PRINT "El programa debe ser iniciado el primero de mes"
8030 PRINT "En caso de final, introduzca por favor e y
ENTER"
8040 PRINT AT 5,0;FLASH 1; "mes =(por ejemplo enero=1,etc.)"
8050 INPUT LINE m$: IF m$= "e" THEN CLS : STOP
8055 FOR i=1 TO 12: IF m$=STR$ i THEN GO TO 8065
8060 NEXT i: GO TO 8050
8065 LET mon=VAL m$
8070 PRINT AT 5,0,,AT 5,5; FLASH 1;"tiempo = ? (por ejemplo
1445)"
8075 INPUT LINE m$: IF m$="e" THEN CLS :STOP
8080 IF LEN m$<>4 AND LEN m$<>3 THEN GO TO 8075
8090 FOR i=1 TO LEN m$
8100 IF m$>"9" OR m$< "0" THEN GO TO 8075
8110 NEXT i
8200 LET hora=VAL m$: LET hr=INT(hora/100)
8210 IF hr<0 OR hr>24 THEN GO TO 8075
8220 LET min=hora-100*hr: IF min>59 THEN GO TO 8075
8230 PRINT AT 5,0,,: LET hr=100*hr

```

De esta forma hemos preguntado, de manera más o menos segura por el mes, y expuesto las posibilidades de mover el reloj. Ahora podemos dedicarnos a dibujar el mapa mundi, la indicación de la hora, de la traslación del tiempo y dibujar el sol. Si, cierto, vamos a dibujar un sol de color rojo que brille en aquel punto del mapa mundi donde sean las 12.00.

```

7200 FOR i=2 TO 4: PRINT AT i,0,,: NEXT i: REM sol
7210 LET yy=150
7220 LET xx=224-4*hora/50
7230 CIRCLE xx+8,yy,8
7240 INK 2
7250 FOR i=-8 TO 8
7260 FOR k=-8 TO 8
7270 IF i*i+k*k<>64 THEN GO TO 7290
7280 PLOT xx+8+k,yy+i
7290 NEXT k
7300 NEXT i

```

Si ahora introducimos una pequeña rutina para editar la hora, podemos visualizar todo en el mapa mundi.

```
7000 INK 6: PRINT AT 5,0,,,: REM hora
7010 PRINT AT 5,9; FLASH 1; hr;" reloj ";min;" horario
    europeo"
7020 RETURN
```

Ahora sigue:

```
8300 BORDER 1: PAPER 1: INK 6: CLS
8310 GO SUB 7000: GO SUB 7200: INK 6
8320 LET y$= "'Día y noche en la tierra': LET xpos=0: LET
    ypos=3: INK 7: GO SUB 20: INK 6
8380 PRINT AT 7,3;" -12 -8 -4 0 +4 +8 +12"
8340 GO TO 9600
```

Esto está muy bien. A partir de la línea 9600 se dibuja el mapa mundi, y si es necesario, el reloj, el sol y las zonas nocturnas se actualizan:

```
9600 LET seg=3000: REM transcurso
9610 GO SUB 9000
```

En la rutina a partir de línea 9000 fijamos para cada mes las zonas de día y noche y las calculamos para la hora correspondiente, después leemos los datos del mapa y lo dibujamos:

```
9000 LET anf=32; RESTORE : REM paint map
9010 DIM f(12) : DIM s(12)
9020 LET f(1)=3: LET f(2)=4: LET f(3)=5: LET f(4)=6:
    LET f(5)=7: LET f(6)=8
9030 LET f(7)=8: LET f(8)=7: LET f(9)=6: LET f(10)=5:
    LET f(11)=4: LET f(12)=3
9040 LET s(1)=5: LET s(2)=6: LET s(3)=7: LET s(4)=7:
    LET s(5)=8: LET s(6)=9
9050 LET s(7)=9: LET s(8)=8: LET s(9)=7: LET s(10)=6:
    LET s(11)=5: LET s(12)=5
```

```
9060 LET a1=xx-8*f(mon): LET a2=xx-8*s(mon)
```

```
9100 READ x1,x2
```

```
9110 IF x1>8000 THEN GO TO 9280
```

```
9120 IF x1=0 THEN LET anf=anf+1: GO TO 9270
```

```
9200 INK 6
```

```
9210 IF anf<a1 THEN INK 0
```

```
9220 IF anf<a2-192 THEN INK 6
```

```
9230 IF anf >=a2 THEN INK 0
```

```
9240 IF anf>=192+a1 THEN INK 6
```

```
9250 PLOT anf,x1
```

```
9260 DRAW 0,x2-x1
```

```
9270 GO TO 9100
```

```
9280 RETURN
```

El mapa mundi está dibujado, y ahora sólo tenemos que conseguir un transcurso adecuado, quiere decirse, la posibilidad de introducir la posibilidad de corregir el reloj, y dibujar cada minuto el reloj y cada hora la pantalla totalmente nueva.

```
9620 LET min=min+1
```

```
9630 LET dif=0
```

```
9640 IF min=60 THEN LET dif=1
```

```
9650 PAUSE 2325
```

```
9660 GO TO 9710
```

```
9700 PAUSE seg
```

```
9710 IF INKEY$="e" THEN CLS : STOP
```

```
9720 IF INKEY$="v" THEN LET seg=50
```

```
9730 IF INKEY$="h" THEN PAUSE 0
```

```
9740 IF INKEY$="n" THEN LET seg=3000
```

```
9750 IF INKEY$="u" THEN GO TO 8070
```

El adelantado y funcionamiento normal del reloj están asegurados ya que la pausa 'seg' cambia su valor. Ahora la marcha del reloj. Tenemos que tener en cuenta que el dibujar el mapa mundi lleva algún tiempo, y tenemos que aumentar el número de minutos en 1, pues el próximo minuto ha comenzado cuando la pantalla estaba lista.

```
9760 LET min=min+1
9770 IF min>60 THEN LET hr=hr+1
9780 LET reloj= 100*hr
9790 IF min>60 THEN LET min=0
9800 IF hr>24 THEN LET hr=0
9810 GO SUB 700
9820 IF min<>0 THEN GO TO 9700
9830 GO SUB 7200
9840 GO SUB 9000
9850 LET min=min+2+dif
9860 LET dif=0
9870 PAUSE 2325
9880 GO TO 9700
```

Ahora sólo faltan los DATA. Estos están contenidos en el listado total al final del capítulo.

Cuando Vd. tenga el mapa mundi en su ordenador, puede ampliar las posibilidades del programa. Si quiere, puede, por ejemplo, dibujar determinados continentes a una mayor escala.

```

1 REM *****
2 REM *****
3 REM
4 REM  F E R R E - M O R E T
5 REM
6 REM *****

10 GO TO B000
20 FOR i=1 TO LEN y$
21 LET x=B*CODE y$(i)+15360
22 FOR z=0 TO 3
23 LET a=PEEK (x+z)
24 LET b=PEEK (x+z+4)
25 POKE USR "a"+2*z,a: POKE USR "a"+2*z+1,a
26 POKE USR "b"+2*z,b: POKE USR "b"+2*z+1,b
27 NEXT z
28 PRINT AT xpos,ypos+i-1;" ";AT xpos+1,ypos+i-1;" ": REM
con A y B en modo grafico
29 NEXT i
30 RETURN
132 DATA 0,0
133 DATA 0,0
134 DATA 0,0
135 DATA 0,0
136 DATA 54,54,72,72,78,80,84,84,88,88,0,0
137 DATA 53,53,72,73,76,82,84,84,86,90,0,0
138 DATA 52,52,73,74,75,92,0,0
139 DATA 50,52,74,75,78,93,0,0
140 DATA 75,94,0,0
141 DATA 77,95,0,0
142 DATA 79,94,0,0
143 DATA 83,93,0,0
144 DATA 80,93,0,0
145 DATA 80,93,0,0
146 DATA 80,93,0,0
147 DATA 80,93,0,0
148 DATA 79,92,0,0
149 DATA 79,92,0,0
150 DATA 78,92,0,0
151 DATA 76,91,0,0
152 DATA 76,92,0,0
153 DATA 75,92,0,0

```

154 DATA 74,92,0,0
 155 DATA 72,93,0,0
 156 DATA 70,94,0,0
 157 DATA 69,93,96,96,0,0
 158 DATA 62,93,95,102,0,0
 159 DATA 60,92,94,102,0,0
 160 DATA 58,92,98,102,0,0
 161 DATA 57,91,93,94,96,99,100,102,0,0
 162 DATA 56,91,93,94,96,99,100,101,0,0
 163 DATA 55,90,92,93,95,100,0,0
 164 DATA 54,55,57,90,93,98,0,0
 165 DATA 53,54,56,90,93,98,0,0
 166 DATA 52,53,55,92,93,98,0,0
 167 DATA 53,92,93,98,0,0
 168 DATA 51,90,93,94,0,0
 169 DATA 49,90,93,93,97,100,0,0
 170 DATA 48,90,96,101,0,0
 171 DATA 48,90,92,92,96,101,0,0
 172 DATA 47,93,96,101,0,0
 173 DATA 46,50,56,90,94,94,96,96,0,0
 174 DATA 47,50,56,80,83,91,93,96,0,0
 175 DATA 46,50,56,68,69,76,85,93,97,100,0,0
 176 DATA 46,52,57,68,70,75,86,91,95,102,0,0
 177 DATA 45,52,57,64,68,68,69,75,84,88,89,102,0,0
 178 DATA 45,48,57,64,68,68,69,75,84,88,89,102,0,0
 179 DATA 44,48,58,68,69,76,84,87,89,102,0,0
 180 DATA 44,46,51,51,56,64,65,67,69,74,86,88,90,97,0,0
 181 DATA 37,37,44,45,51,51,53,65,66,68,69,72,94,97,0,0
 182 DATA 35,39,44,45,51,51,53,65,66,68,69,71,94,97,0,0
 183 DATA 34,40,43,45,50,51,59,73,75,83,94,97,0,0
 184 DATA 32,44,48,50,61,83,88,97,0,0
 185 DATA 6,10,12,18,30,45,52,52,64,81,84,97,0,0
 186 DATA 4,24,28,46,49,51,65,81,83,96,0,0
 187 DATA 2,46,49,49,66,80,83,95,0,0
 188 DATA 4,44,50,50,66,80,84,90,0,0
 189 DATA 4,4,12,44,49,49,65,69,70,82,84,90,0,0
 190 DATA 4,4,12,45,49,49,65,66,72,80,87,89,0,0
 191 DATA 14,46,48,48,66,66,72,79,88,88,0,0
 192 DATA 14,45,47,47,69,76,0,0
 193 DATA 15,44,67,73,0,0
 194 DATA 16,43,66,72,100,102,0,0
 195 DATA 19,42,68,70,96,102,0,0

86

196 DATA 20,41,69,69,88,90,94,96,98,102,0,0
 197 DATA 21,41,85,102,0,0
 198 DATA 22,39,82,102,0,0
 199 DATA 24,39,81,102,0,0
 200 DATA 26,38,80,102,0,0
 201 DATA 26,37,84,102,0,0
 202 DATA 27,37,85,102,0,0
 203 DATA 28,38,86,102,0,0
 204 DATA 32,37,87,102,0,0
 205 DATA 88,102,0,0
 206 DATA 88,102,0,0
 207 DATA 89,102,0,0
 208 DATA 90,102,0,0
 209 DATA 91,102,0,0
 210 DATA 92,95,97,102,0,0
 211 DATA 93,95,97,102,0,0
 212 DATA 93,95,96,102,0,0
 213 DATA 84,87,0,0
 214 DATA 84,88,0,0
 215 DATA 83,88,0,0
 216 DATA 47,54,85,88,0,0
 217 DATA 45,56,0,0
 218 DATA 43,58,61,65,72,75,0,0
 219 DATA 42,60,61,65,72,75,0,0
 220 DATA 42,59,61,65,74,76,79,79,0,0
 221 DATA 43,59,61,65,70,70,72,80,0,0
 222 DATA 43,60,62,65,68,71,72,76,0,0
 223 DATA 43,60,62,71,73,73,0,0
 224 DATA 44,61,64,72,0,0
 225 DATA 43,61,66,73,82,82,0,0
 226 DATA 42,61,67,73,80,83,0,0
 227 DATA 41,61,67,73,80,85,0,0
 228 DATA 41,61,67,74,76,76,80,85,0,0
 229 DATA 40,59,62,62,64,64,66,78,80,86,0,0
 230 DATA 37,58,65,74,78,90,0,0
 231 DATA 36,58,62,62,64,65,67,74,76,92,0,0
 232 DATA 28,30,32,58,62,66,68,74,77,94,0,0
 233 DATA 26,57,65,65,67,74,84,94,0,0
 234 DATA 22,57,64,74,82,83,85,95,0,0
 235 DATA 19,58,62,78,81,83,85,96,0,0
 236 DATA 18,58,61,78,81,84,86,96,0,0
 237 DATA 19,57,59,59,62,86,88,97,0,0

238 DATA 20,58,64,80,82,97,0,0
239 DATA 21,58,60,64,66,94,0,0
240 DATA 23,57,60,64,66,94,0,0
241 DATA 25,52,54,58,60,86,88,94,0,0
242 DATA 25,51,54,64,67,85,88,93,0,0
243 DATA 28,50,53,64,67,87,89,91,0,0
244 DATA 29,48,53,63,66,84,88,90,0,0
245 DATA 30,34,38,47,52,63,65,88,0,0
246 DATA 30,33,39,46,51,88,0,0
247 DATA 39,45,47,91,0,0
248 DATA 25,31,40,45,47,88,0,0
249 DATA 25,32,44,46,48,62,68,90,0,0
250 DATA 48,51,54,60,67,89,93,101,0,0
251 DATA 48,52,56,89,96,100,0,0
252 DATA 49,53,56,90,96,96,100,100,0,0
253 DATA 50,52,56,89,91,92,0,0
254 DATA 50,51,55,90,0,0
255 DATA 51,51,55,90,0,0
256 DATA 55,90,0,0
257 DATA 55,90,0,0
258 DATA 54,92,94,96,0,0
259 DATA 53,97,0,0
260 DATA 52,99,0,0
261 DATA 51,87,89,100,0,0
262 DATA 50,87,89,100,0,0
263 DATA 50,88,0,0
264 DATA 48,88,0,0
265 DATA 46,90,92,99,0,0
266 DATA 44,90,92,97,0,0
267 DATA 43,44,46,99,0,0
268 DATA 49,100,0,0
269 DATA 50,100,0,0
270 DATA 51,100,0,0
271 DATA 52,100,0,0
272 DATA 53,100,0,0
273 DATA 42,42,52,100,0,0
274 DATA 41,42,51,100,0,0
275 DATA 40,42,51,100,0,0
276 DATA 39,41,43,100,0,0
277 DATA 38,40,42,43,45,100,0,0
278 DATA 37,39,40,42,44,100,0,0
279 DATA 36,37,39,41,43,100,0,0

280 DATA 35,35,37,38,42,100,0,0
 281 DATA 34,35,41,100,0,0
 282 DATA 34,35,37,39,44,44,48,100,0,0
 283 DATA 24,25,34,35,37,40,49,100,0,0
 284 DATA 23,26,34,35,37,41,49,100,0,0
 285 DATA 19,26,34,34,38,42,50,99,0,0
 286 DATA 19,27,41,41,43,43,46,46,53,99,0,0
 287 DATA 20,28,34,34,36,40,44,44,54,55,57,57,60,60,64,98,0,
 0
 288 DATA 21,30,36,40,44,45,48,48,63,99,0,0
 289 DATA 21,31,44,45,64,99,0,0
 290 DATA 22,32,61,100,0,0
 291 DATA 22,32,61,100,0,0
 292 DATA 22,33,59,59,62,62,64,97,0,0
 293 DATA 22,33,58,60,66,97,0,0
 294 DATA 21,33,39,40,59,60,66,96,0,0
 295 DATA 19,32,37,38,60,61,66,96,0,0
 296 DATA 18,31,34,38,61,63,66,74,76,98,0,0
 297 DATA 17,31,34,38,62,64,66,66,80,98,0,0
 298 DATA 16,33,35,38,65,74,80,98,0,0
 299 DATA 16,33,36,38,65,67,72,72,80,97,0,0
 300 DATA 13,15,17,31,35,36,66,66,80,97,0,0
 301 DATA 13,15,17,30,35,36,80,97,0,0
 302 DATA 19,28,35,35,80,96,0,0
 303 DATA 23,25,80,94,0,0
 304 DATA 81,94,0,0
 305 DATA 73,79,82,94,0,0
 306 DATA 71,80,82,94,0,0
 307 DATA 72,80,82,94,0,0
 308 DATA 74,80,82,93,0,0
 309 DATA 74,81,82,91,0,0
 310 DATA 76,81,83,91,0,0
 311 DATA 78,91,0,0
 312 DATA 13,13,80,94,0,0
 313 DATA 12,14,80,92,0,0
 314 DATA 13,15,81,94,0,0
 315 DATA 14,15,17,19,82,94,0,0
 316 DATA 15,18,83,94,0,0
 317 DATA 15,17,84,94,0,0
 318 DATA 16,17,84,93,0,0
 319 DATA 16,16,83,83,86,92,0,0
 320 DATA 88,92,0,0

```

321 DATA 87,91,0,0
322 DATA 86,90,0,0
323 DATA 90,90,0,0
500 DATA 9999,9999
7000 INK 6: LET t=6: PRINT AT 5,0,,: REM hora
7010 PRINT AT 5,3; FLASH 1;st;"h.";min;"min. horario europeo
"
7020 RETURN
7200 FOR i=2 TO 4: PRINT AT i,0,,: NEXT i: REM sol
7210 LET yy=150
7220 LET xx=224-4*reloj/50
7230 CIRCLE xx+8,yy,8
7240 INK 2
7250 FOR i=-8 TO 8
7260 FOR k=-8 TO 8
7270 IF i*i+k*k>64 THEN GO TO 7290
7280 PLOT xx+8+k,yy+i
7290 NEXT k
7300 NEXT i
7310 RETURN
8000 CLS : LET y$="Dia y noche en la Tierra ? pcb": LET xpo
s=0: LET ypos=0: GO SUB 20
8010 PRINT AT 9,0;"Significado durante la ejecucionv adelan
tar h parar el relojn normal u corregir reloj
e fin"
8020 PRINT " El programa ha de iniciarse a primero de mes.
"
8030 PRINT "Para finalizar pulsar e y ENTER"
8040 PRINT AT 5,0; FLASH 1;"mes = ? (p.ej. Enero <=> 1 )"
8050 INPUT LINE m$: IF m$="e" THEN CLS : STOP
8055 FOR i=1 TO 12: IF m$=STR$ i THEN GO TO 8065
8060 NEXT i: GO TO 8050
8065 LET mes=VAL m$
8070 PRINT AT 5,0,,AT 5,5; FLASH 1;"hora = ? (p.ej. 1445)"
8075 INPUT LINE m$: IF m$="e" THEN CLS : STOP
8080 IF LEN m$<>4 AND LEN m$<>3 THEN GO TO 8075
8090 FOR i=1 TO LEN m$
8100 IF m$(i)>"9" OR m$(i)<"0" THEN GO TO 8075
8110 NEXT i
8200 LET reloj=VAL m$: LET st=INT (reloj/100)
8210 IF st<0 OR st>24 THEN GO TO 8075
8220 LET min=reloj-100*st: IF min>59 OR min<0 THEN GO TO 80

```

```

75
8230 PRINT AT 5,0,, : LET reloj=100*st
8300 BORDER 1: PAPER 1: INK 6: CLS
8310 GO SUB 7000: GO SUB 7200: INK 6
8320 LET y$="dia y noche en la Tierra": LET xpos=0: LET ypos
=3: INK 7: GO SUB 20: INK 6
8330 PRINT AT 7,3,"-12 -8 -4 0 +4 +8 +12"
8340 GO TO 9600
9000 LET anf=32: RESTORE : REM pintar mapa
9010 DIM f(12): DIM s(12)
9020 LET f(1)=3: LET f(2)=4: LET f(3)=5: LET f(4)=6: LET f(5)
)=7: LET f(6)=8
9030 LET f(7)=8: LET f(8)=7: LET f(9)=6: LET f(10)=5: LET f(
11)=4: LET f(12)=3
9040 LET s(1)=5: LET s(2)=6: LET s(3)=7: LET s(4)=7: LET s(5)
)=8: LET s(6)=9
9050 LET s(7)=9: LET s(8)=8: LET s(9)=7: LET s(10)=6: LET s(
11)=5: LET s(12)=5
9060 LET a1=xx-8*f(mes): LET a2=xx+8*s(mes)
9100 READ x1,x2
9110 IF x1>8000 THEN GO TO 9280
9120 IF x1=0 THEN LET anf=anf+1: GO TO 9270
9200 INK 6
9210 IF anf<a1 THEN INK 0
9220 IF anf<a2-192 THEN INK 6
9230 IF anf>=a2 THEN INK 0
9240 IF anf>=192+a1 THEN INK 6
9250 PLOT anf,x1
9260 DRAW 0,x2-x1
9270 GO TO 9100
9280 RETURN
9600 LET seg=3000: REM transcurso
9619 GO SUB 9000
9620 LET min=min+1
9630 LET dif=0
9640 IF min=60 THEN LET dif=1
9650 PAUSE 2325
9660 GO TO 9710
9700 PAUSE seg
9710 IF INKEY$="e" THEN CLS : STOP
9720 IF INKEY$="v" THEN LET seg=50
9730 IF INKEY$="h" THEN PAUSE 0 Q{
9740 IF INKEY$="n" THEN LET seg=3000
9750 IF INKEY$="u" THEN GO TO 8070
9760 LET min=min+1
9770 IF min>=60 THEN LET st=st+1
9780 LET reloj=100*st
9790 IF min>=60 THEN LET min=0
9800 IF st>=24 THEN LET st=0
9810 GO SUB 7000
9820 IF min<>0 THEN GO TO 9700
9830 GO SUB 7200
9840 GO SUB 9000
9850 LET min=min+2+dif
9860 LET dif=0
9870 PAUSE 2325
9880 GO TO 9700

```

CAPITULO 7: APARATOS ADICIONALES Y AMPLIACIONES

Introducción

Tarde o temprano, a todos los propietarios de ordenadores personales les sucede lo mismo: buscan accesorios para mejorar sus sistema, ampliar o, simplemente, calmar sus ansias de juego.

La propia empresa Sinclair comercializa una serie de aparatos adicionales, como son el Interfaced 1 y los Microdrives que amplían seriamente las posibilidades del ordenador. Gracias a la gran divulgación del Spectrum, un gran número de empresas independientes se han especializado en la producción de periféricos para ordenadores Sinclair. De esta forma existe para el Spectrum, como para ningún otro ordenador, una gran variedad para elegir que sólo está limitada por el propio presupuesto financiero.

El primer paso es, casi siempre, una impresora, ya que el editar programas largos es un trabajo laborioso, si sólo se dispone de la pantalla. La oferta más económica es la miniimpresora de Sinclair. Se puede conectar directamente al Spectrum y escribe sobre papel metálico. Desde luego, este papel no es muy barato. Tampoco la calidad de impresión es imponente. Entretanto, la producción de este aparato ha cesado. Pero otras empresas fabrican igualmente pequeñas impresoras, que escriben sobre papel térmico o sobre papel normal dejando una impresión marcada.

Si la impresora sólo se utiliza de vez en cuando para listados, es suficiente la Impresora Alphacon. Usuarios más exigentes pueden adquirir con la Seikosha GP 50S una verdadera impresora de matriz. Con interface paralelo de Centronics es posible conectar muchos tipos de impresoras de 80 y más caracteres por línea, fabricadas en sí para ordenadores mucho más grandes. Estas disponen generalmente de un tipo de letra muy agradable, y es posible utilizar el Spectrum como pieza básica para un sistema de procesado de texto.

Si de verdad se quiere escribir mucho texto con el Spectrum, llegará el momento en el que las teclas de goma no satisfarán. (Esto no vale, naturalmente, para propietarios del Spectrum Plus). Muchas empresas se han dado cuenta de este hueco en el mercado, y así existen sobradas posibilidades. Sinclair ha dado máximas facilidades a estos fabricantes de teclados adicionales. El teclado del Spectrum es en sí una parte separada totalmente del ordenador. Si se examina, se verá que del teclado parten dos tiras de cables hacia la tarjeta, los cuales están introducidos simplemente en un conector. Cualquier teclado se puede conectar sacando estos dos cables e introduciéndolos en el nuevo teclado.

Si ha decidido invertir una determinada cantidad en un teclado nuevo y abierto el Spectrum, Vd. se verá recompensado. Todo este libro se ha redactado con tal configuración.

El Interface 1

El aparato adicional más versátil es sin duda el Interface 1 de Sinclair Research. Por un precio muy económico ofrece la posibilidad de acceso a memorias de gran capacidad: los Microdrives, así como una conexión RS 232 con la cual es posible la conexión con otros ordenadores, impresoras y también Modems. Además permite la formación de una red local de hasta 64 Spectrums (y Sinclair QLs). En ello se conserva la posibilidad de conexión a un magnetófono y la impresora ZX.

El Interface 1 dispone de una ROM adicional de 8 KByte, el cual se suma a los 16 KByte ROM del Spectrum. Esta ROM adicional amplía el juego de comandos del Spectrum:

- CAT
- CLOSE
- ERASE
- FORMAT
- MOVE
- OPEN

así como amplía las posibilidades de los siguientes comandos:

- CLS
- CLEAR
- INPUT
- INKEY\$
- PRINT
- LPRINT
- LIST
- LLIST
- SAVE
- LOAD
- MERGE
- VERIFY

El significado de estos comandos se expone en el apéndice. Aquí sólo vamos a tratar algunas peculiaridades.

El comando 'CAT' se utiliza para obtener un índice de una cassette de Microdrive. Por ejemplo el comando 'CAT 1' suministra el nombre de la cassette en el Microdrive 1 y los nombres de los primeros 50 ficheros no protegidos contenidos en la misma. Un fichero protegido es aquel cuyo nombre comienza con CHR\$ 0. Estos ficheros son ignorados por 'CAT'. Además se indica la memoria utilizable en KBytes.

Esta lista es visualizada en la pantalla con la forma anterior de comando. Si se quiere imprimir esta lista, hay que abrir la corriente correspondiente e introducir la fórmula:

CAT #n;1

"n" es el número de la corriente correspondiente.

El comando FORMAT se utiliza para fijar la densidad de transmisión a través de la conexión RS 232. Se pueden introducir los valores 50,110,300,600,1200,2400,4800,9600 y 19200. Si no se introduce ningún valor, la densidad Baud será de 9600. Si se introduce un valor no previsto, se aplicará siempre el próximo valor inferior 'permitido'. El mínimo es siempre 50.

Con el Interface esta permitido igualmente CLS#. A diferencia del comando CLS simple, no sólo se borra el fichero de visualización en pantalla sino que, además, se fija:

INK	0, ó sea negro
PAPER	7, ó sea blanco
BORDER	igualmente 7
INVERSE	se pone en 0

BRIGHT	se pone a 0
OVER	se pone a 0
FLASH	se pone a 0

El comando CLEAR# borra datos posibles, fijados a través de corrientes. Todos las corrientes se conectan en cortocircuito y los datos que se encuentren en cualquier memoria intermedia se pierden, cosa que no ocurre con el comando CLOSE#.

Se pueden conectar y comandar hasta ocho Microdrives paralelas al Interface 1. Con estos Microdrives se pueden escribir y leer sin fin, cassettes especiales, y utilizarlas así como memoria.

Cada una de estas cassettes contiene una cinta de unos 5 m de longitud, que tarda unos 7 segundos en pasar totalmente. En ermino medio, el tiempo de acceso a datos es de 3,5 segundos. Cada cassette puede memorizar datos en 90 Kbyte aproximadamente. Es aconsejable efectuar varias veces el formateo, ya que la memoria disponible puede variar en tales intentos.

Una particularidad del Interface 1 es la red local LAN, la cual permite la conexión directa de hasta 64 Spectrums y/6 S Fair QLS. Esta conexión se realiza con una cable simple de dos hilos. Los datos se intercambian con una velocidad de 5 KByte por segundo.

Con el comando FORMAT "n", 7 se le asigna a un Spectrum por ejemplo el número de estación 7. Este número de estación debe estar comprendido entre 1 y 64. El mismo efecto se consigue con el comando POKE 23749, 7.

El intercambio de datos es bastante simple. Se utilizan los comandos LOAD, SAVE, MERGE, etc, sustituyendo únicamente la "m" de Microdrives por una "n". Las posibilidades de aplicación son múltiples. Como explica el manual del Interface 1, algunos propietarios del Spectrum pueden compartir un aparato periférico caro o, por ejemplo, una impresora.

Se pueden elaborar juegos totalmente nuevos, si se conectan Spectrum entre sí. También se pueden redactar programas en los cuales cada ordenador se encarga de trabajar con una parte del programa. Los resultados de cálculos pueden ser intercambiados y representados en las dos o en varias pantallas (o más según el número de ordenadores). Tanto por parte de la Hardware como de la Software se abre un campo de actividades amplio.

Impresoras en serie

Los propietarios del Interface 1 y con ello de la conexión RS 232, tienen la posibilidad de utilizar, aparte de impresoras puras, máquinas de escribir con una conexión adecuada como Impresoras. Entre algunos modelos destacaremos el EP44 de la empresa Brother. El motivo es la versatilidad de este aparato.

Esta máquina de escribir lo hace sin cinta sobre papel térmico (el cual no es barato), o con cinta (lo cual tampoco es barato) sobre papel convencional. La impresión es muy buena gracias a una matriz de puntos de 18 x 24. La aplicación como Impresora en conexión con el Spectrum no puede ser más fácil. Se conecta el cable RS 232 en el enchufe e introduce en forma de parámetros los siguientes valores para la transmisión de datos:

BAUD RATE	a voluntad, pero siempre de acuerdo con los valores del Spectrum
BIT LENGTH	8
PARITY	N
NEX LINE	CR + LF
CODE	7 BIT
ER	Y

Entonces se coloca el conmutador en la posición 'TERMINAL' y se introduce el siguiente comando en el Spectrum:

```
OPEN# 3,"+": FORMAT "+"; baud rate
```

donde en lugar de 'baud rate' se encuentra el valor que se introdujo en la máquina EP 44.

A partir de ahora los comandos LPRINT y LLIST actúan en la EP 44. Esto es porque LPRINT y LLIST no significan otra cosa que PRINT#3 y LIST#3. COPY no funciona porque desgraciadamente, la EP 44 no tiene cualidades gráficas.

La máquina de escribir dispone de una serie de posibilidades de memorización de texto y transmisión a distancia de datos. Incluso dispone de un editor de texto. Lo que es realmente interesante para propietarios de Spectrum sin teclado adicional, es la posibilidad de utilizar la EP 44 como aparato de entrada para el Spectrum.

Una rutina sencilla para lograr esto es la siguiente:

```
10 OPEN#3,"+":FORMAT "+";300
20 BORDER 0
100 LET m$=INKEY#3
110 IF CODE m$>31 THEN PRINT m$;
150 GO TO 100
```

Todo lo que ahora se introduzca en la EP 44 se visualiza en la pantalla. Esta rutina indica el método para realizar la entrada en sus programas a través de la máquina de escribir. Es un buen negocio utilizar la EP 44 como impresora aunque de forma limitada como teclado, lo cual significa para muchos propietarios del Spectrum un ahorro de 11.000 Ptas.

Si se quisiera escribir desde el Spectrum directamente en la máquina de escribir, y quizás mantener un sistema de diálogo entre ambos, sólo hay que insertar las siguientes dos líneas en la rutina anterior:

```
120 LET n$=INKEY$
```

```
130 IF CODE n$>31 THEN PRINT#3;n$
```

Naturalmente, de esta manera, es factible visualizar en la pantalla el texto que estuviera memorizado en la EP 44. Las posibilidades aquí son numerosas.

Lápiz óptico

Un aparato adicional muy atractivo para el Spectrum (como para cualquier ordenador) es un lápiz óptico (conocido generalmente por Ligthpen). Con este lápiz se pueden hacer cosas que para un inexperto parecen de magia. Se pueden 'dibujar' líneas, círculos y otras figuras en la pantalla. La utilización del lápiz al elegir en un menú es muy espectacular.

Pero ciertamente, no se pueden esperar verdaderos milagros del lápiz. El que, por ejemplo quiera dibujar la catedral de Aquisgrán en la pantalla, y quiera conseguir líneas más o menos rectas, tiene que practicar bastante tiempo.

¿Cómo funciona un lápiz óptico? En principio es un receptor de luz, que la transforma en una señal eléctrica inteligible para el ordenador. La imagen de televisión se escribe 25 veces por segundo con un rayo de electrones, para lo cual este rayo pasa por 625 líneas horizontales. Si se coloca el lápiz sobre la pantalla, éste 've' luz siempre que el rayo de electrones pase por su situación. En este momento es cuando el lápiz manda una señal al ordenador.

Este sólo tiene que calcular cuánto tiempo ha transcurrido desde que el rayo de electrones comenzó con la imagen, o sea, cuándo se encontraba en la esquina izquierda superior. Con esta diferencia de tiempo, el ordenador sabe exactamente la posición en que se encontraba el lápiz óptico.

Aunque es casi imposible realizar dibujos complicados a mano alzada, se puede conseguir mucho con software adecuada. El lápiz óptico es también un aparato de introducción agradable para inexpertos en ordenadores, ya que sólo tienen que señalar los puntos que quiere realizar.

Si con el lápiz sólo se quiere efectuar la elección en un menú, esto es bastante fácil. Pero hay que tener cuidado de que el lápiz no reciba luz antes de llegar a la pantalla, o mientras el usuario se mueve sobre la pantalla para llegar al punto deseado.

Un método sencillo para eliminar tales 'errores', consiste en efectuar dos confirmaciones y dejar entre ambas un corto lapso de tiempo. Si ambas confirmaciones tienen el mismo resultado, significa que el lápiz ha sido mantenido quieto en una misma posición para seleccionar.

En un programa esto puede escribirse como sigue:

```
100 LET lÍnea1=USR 63109 ( o donde haya sido colocada
    la rutina por el fabricante)
110 PAUSE 10
120 LET lÍnea 2=USR 63109
130 IF lÍnea1=lÍnea2 THEN GO TO .....
140 GO TO 100
```

La pausa en la lÍnea 110 puede ser prolongada a voluntad hasta que se consiga un resultado óptimo.

Procesado de texto

El programa de procesado de texto TASWORD II para el Spectrum, es casi con seguridad el más divulgado. Esto tiene su motivo, ya que el programa es sencillamente muy bueno. Aparte de ello es muy agradable para el usuario trabajar con él. Vamos a dar algunas observaciones sobre el programa mismo.

El programa se suministra con una descripción muy detallada. Pero ésta no se utiliza mucho tiempo, ya que el manejo es muy fácil. Además es posible, a través de dos 'páginas de ayuda', visualizar las características más importantes en la pantalla, incluso durante el trabajo con el programa.

El programa ofrece todas las posibilidades que se puedan esperar de un procesado de texto. El cursor se puede dirigir hacia las cuatro direcciones. En todo momento es posible corregir, insertar o borrar en cualquier punto. El programa visualiza en el caso normal con 64 caracteres por línea. A quien este tipo de letra le parezca muy pequeña, puede editar el texto con la escritura normal del Spectrum.

El texto es formateado automáticamente al ser introducido. Esto significa que en cada línea se introducen tantas separaciones como sean necesarias para conseguir un borde uniforme en el lado derecho. Incluso después de borrar o insertar texto se puede conseguir en todo momento este formateado.

Se pueden centrar palabras y correrlas hacia la izquierda o derecha. Es posible buscar palabras en el texto para ser corregidas o sustituidas, así como es posible mover y copiar bloques enteros de texto. También es posible 'mezclar' (MERGE) texto memorizado con otro documento.

Lo agradable para el usuario consiste en que el programa puede ser copiado para el propio uso. Incluso es más: Se ha previsto una posición en el menú que permite realizar esto automáticamente. Como consecuencia, es posible adaptar el programa a la propia impresora según las indicaciones pertinentes, y esta versión adaptada es la que se puede memorizar y aplicar.

Otra ventaja consiste en que la parte de mando del programa esta escrita en BASIC, y es muy fácil transcribirlo para su utilización con Interface 1 y Microdrives. ¡Quién no se enfada cuando tiene un programa fantástico de ajedrez y quiere traspasarlo después de adquirir Microdrives sin conseguirlo, y debe esperar, en consecuencia, muchos minutos para cargarlo!

Si se transcribe TASWORD II a Microdrives, hay que ser ahorrativo con la memoria, ya que para la parte en BASIC sólo se dispone de las direcciones hasta 31999. Por ello es aconsejable efectuar medidas de ahorro de memoria. A continuación se expone un ejemplo de cómo funciona el programa utilizando dos Microdrives y cassette para la memorización, así como una impresora conectada a través de la conexión RS 232. También se han cambiado los colores de la pantalla, lo cual va a gusto del consumidor.

```

5 BORDER 7: PAPER 7: INK 0: LET q$="pcb"
10 CLS : LET a=USR 64330: GO TO 20
15 BRIGHT 1: INK 0: PAPER 7: BORDER 7: POKE 23609,2: CLEAR 31999: LET q$="pcb"
: GO SUB 4000: LOAD *"m";1;"taswordmc"CODE : CLS : LET a=USR 59081: GO TO 10
20 BORDER NOT PI: PAPER NOT PI: INK VAL "6": CLS : LET a=64*INT (a/64+0.99): I
F a=NOT PI THEN GO TO 3000
25 LET a$=" text file"
26 GO SUB 4000: LET v=VAL "31": PRINT AT VAL "4",NOT PI;"print";a$;TAB v;"p"
28 PRINT "save";a$;TAB v;"s"
30 PRINT "load";a$;TAB v;"j"
35 PRINT "merge";a$;TAB v;"m"
40 PRINT "return to";a$;TAB v;"y"
45 PRINT "define graphics/printer";TAB v;"g"
50 PRINT "save tasword";TAB v;"t"
55 PRINT "into Basic";TAB v;"b"
70 PRINT AT 20,VAL "7";"press key"
80 LET a$=INKEY$: IF a$="" THEN GO TO 80
90 LET i=0: LET b=CODE a$: IF b<97 THEN LET b=b+32
110 IF b=115 THEN LET i=VAL "6"
120 IF b=106 THEN LET i=VAL "8"
125 IF b=116 THEN LET i=16
130 IF b=112 THEN LET i=VAL "4"
140 IF b=121 THEN LET i=12
150 IF b=109 THEN LET i=10
160 IF b=103 THEN LET i=14
170 IF b=98 THEN LET i=18
180 IF i>VAL "0" THEN PRINT AT i,31; FLASH VAL "1";CHR$ b;; GO TO 500
190 GO TO 80
200 CLS : GO SUB 4000: PRINT AT VAL "4",VAL "8";"PRINT OPTIONS": PRINT " just
press ENTER for default values given in brackets"
210 LET i=VAL "8": LET j0=23: PRINT AT i,NOT PI;"Line spacing? (1)": GO SUB 600
0: IF a$="" THEN LET a$="1"
215 POKE 62235,VAL a$
220 LET i=10: PRINT AT i,NOT PI;"Start at line? (1)": GO SUB 6000: IF a$="" THE
N LET a$="1"
230 LET c=64*(INT VAL a$-1): LET b=c+FN p(62216): LET x=60045: GO SUB 950
240 LET i=12: PRINT AT i,NOT PI;"Finish at line? (last)": GO SUB 6000: IF a$=""
THEN LET b=a-c: GO TO 250
245 LET b=64*INT VAL a$-c
250 RANDOMIZE USR 59806: RANDOMIZE USR (FN p(62472))
260 CLS : PRINT AT 20,NOT PI;"Press the q key to quit printing"
265 FORMAT "t";300: OPEN 3;"t"

```

```

270 LET x=60049: GO SUB 950
275 LET c=PEEK 62470: IF c<>0 THEN LPRINT CHR$ c
280 RANDOMIZE USR 60038
285 LET c=PEEK 62471: IF c<>0 THEN LPRINT CHR$ c
290 RANDOMIZE USR 59806: CLOSE 3: GO TO 5
300 CLS : GO SUB 4000: PRINT "Printer control graphics chars:"
305 LET b=VAL "4": FOR i=NOT PI TO 15: LET b=ABS (b-4): PRINT AT i+4,b;i+128;CH
R$ (i+128)
307 FOR j=NOT PI TO VAL "3": LET c=PEEK (60860+i*4+j): LET a$=STR$ c: IF c=255
THEN LET a$=""
308 PRINT AT i+4,10+4*j;a$: NEXT j: NEXT i
320 INPUT "Type graphics character code 128-143 (ENTER if finished)";a$
325 IF a$="" THEN GO TO 400
340 LET b=VAL a$: IF b<128 OR b>143 THEN GO TO 320
350 PRINT AT 21,VAL "3"; FLASH VAL "1";b; FLASH NOT PI;" ";CHR$ b
355 FOR j=NOT PI TO VAL "3": POKE (60348+b*4+j),255: NEXT j
360 FOR j=NOT PI TO VAL "3": INPUT "Code? (ENTER if finished)";a$: IF a$="" THE
N GO TO 300
370 POKE (60348+b*4+j),VAL a$: PRINT AT 21,10+4*j;VAL a$: NEXT j: GO TO 300
400 CLS : GO SUB 4000: PRINT AT VAL "3",NOT PI;"Reset interface/printer codes?"
: LET i=VAL "5": GO SUB 920: CLS : IF i=NOT PI THEN GO TO 25
401 GO SUB 4000: PRINT AT 3,0;"just ENTER to keep values given:"
403 LET j0=27: LET i=8: LET j=0: LET a$="Interface control code1=": LET x=60924
: GO SUB 850
404 LET i=9: LET j=18: LET a$="code2=": LET x=62470: GO SUB 850
405 LET i=10: LET j=18: LET a$="code3=": LET x=62471: GO SUB 850
406 LET i=11: PRINT AT i,18;"code4=";FN p(62472): LET i=12: LET j0=24: GO SUB 6
000: LET j0=27: IF a$<>"" THEN LET b=VAL a$: LET x=62472: GO SUB 950
410 LET i=14: LET j=0: LET a$="Printer carriage return=": LET x=60925: GO SUB 8
50
420 LET i=16: LET j=0: LET a$="Printer linefeed=": LET x=60926: GO SUB 850
430 LET i=18: LET j=0: LET a$="Left margin on printing=": LET x=60927: GO SUB 8
50
490 GO TO 20
500 PRINT AT 20,0;" press the "; FLASH 1;"ENTER"; FLASH 0;" key to proceed";AT
21,0;" press "; FLASH 1;"c"; FLASH 0;" to change the choice "
510 LET a$=INKEY$: IF a$="c" THEN GO TO 20
520 IF CODE a$=13 THEN GO TO 600
530 GO TO 510
600 IF b=116 THEN GO TO 700
610 IF b=121 THEN CLS : GO TO 5
620 IF b=115 THEN CLS : GO TO 1000

```

```

630 IF b=109 THEN GO TO 2000
640 IF b=106 THEN LET a=USR 59081: LET a=0: GO TO 2000
650 IF b=112 THEN GO TO 200
660 IF b=103 THEN GO TO 300
699 CLS : STOP
700 CLS : LET i=8: LET a$="tasword": SAVE *"m";1;a$ LINE 15
710 SAVE *"m";1;a$+"mc"CODE 54784,10751
770 PRINT AT 19,0;
780 VERIFY *"m";1;a$: PRINT AT 21,0;"tasword: basic O.K.";AT 19,0;
790 VERIFY *"m";1;a$+"mc"CODE 54784,10751: PRINT AT 21,20;" m/code O.K.": GO TO
25
850 PRINT AT i,j;a$;PEEK x: GO SUB 6000: IF a$<>" THEN POKE x,VAL a$
860 RETURN
920 PRINT AT i,4;"press y for yes";AT i+2,11;"n for no"
930 IF INKEY$="n" OR INKEY$="N" THEN LET i=0: RETURN
940 IF INKEY$="y" OR INKEY$="Y" THEN LET i=1: RETURN
945 GO TO 930
950 POKE x,b-256*INT (b/256): POKE (x+1),INT (b/256): RETURN
1000 LET b=FN p(62216): CLS
1005 PRINT AT VAL "8",NOT PI;"Name of text file for saving?": LET i=10: LET j0=N
OT PI: GO SUB 6000
1010 IF a$="" OR LEN a$>10 THEN GO TO 1005
1020 INPUT "MD (1 or 2) or c for Cassette"; LINE m$: IF m$<>"1" AND m$<>"2" AND
m$<>"c" THEN GO TO 1020
1022 IF m$="c" THEN GO TO 1035
1025 ERASE "m";VAL m$;a$
1030 LET i=12: SAVE *"m";VAL m$;a$CODE b,a: LET i=14: VERIFY *"m";VAL m$;a$CODE
b,a: GO TO 1040
1035 LET i=12: SAVE a$CODE b,a: LET i=14: PRINT "rewind to verify and press key"
: PAUSE 0: VERIFY a$CODE b,a
1040 CLS : PRINT AT 8,NOT PI;"text file ";a$;" saved:";AT 10,NOT PI;a;" bytes","
a/PEEK 62237;" lines"
1100 PRINT AT 21,6;"text file verified": PAUSE 150: GO TO 25
2000 CLS : BORDER 0: PRINT AT 8,0;"type the name of the text file";AT 10,0;"and
press ENTER"
2020 LET j0=0: LET i=16: GO SUB 6000
2025 INPUT "MD (1 or 2) or c for cassette "; LINE m$: IF m$<>"1" AND m$<>"2" AND
m$<>"c" THEN GO TO 2025
2030 LET b=FN p(62216): IF m$<>"c" THEN LOAD *"m";VAL m$;a$CODE (a+b),((FN p(62
221)+22)*64-a): GO TO 5
2040 LET b=FN p(62216): IF m$="c" THEN LOAD a$CODE (a+b),((FN p(62221)+22)*64-a
): GO TO 5

```

```

3000 FOR i=23296 TO 23361: POKE i,32: NEXT i
3003 POKE 23362,NOT PI
3007 PRINT AT 3,4;"to use old word just ENTER"AT 5,(LEN q$(27))*(26-LEN q$)/2;"o
w="";q$;" "
3010 PRINT AT 8,0;" or type a new word to be replaced / found"
3012 LET j0=NOT PI: LET i=10: GO SUB 6000: IF a$="" THEN LET a$=q$
3015 LET q$=a$
3020 LET j=NOT PI: FOR i=VAL "1" TO LEN a$: POKE 23297+i,CODE a$(i): IF a$(i)="
" THEN LET j=j+1
3021 NEXT i
3022 IF j<>NOT PI THEN CLS : PRINT AT 12,0;"just a word - no spaces allowed": G
O TO 3000
3025 POKE 23297,LEN a$
3030 PRINT AT 12,NOT PI;"with (just ENTER for find only)": LET i=14: GO SUB 6000
3040 IF a$="" THEN POKE 23362,1: GO TO 3060
3050 FOR i=VAL "1" TO LEN a$: POKE 23329+i,CODE a$(i): NEXT i
3060 LET a=USR 64955: LET a=USR 64333: GO TO 20
4000 PRINT AT NOT PI,VAL "9";"Tasword Two";AT VAL "1",VAL "4";"? Tasman Software
1983";AT VAL "2",VAL "6";"MICRODRIVE VERSION": RETURN
6000 LET u$=" ": LET hl=NOT PI: LET a$="": PRINT AT i,j0: FLASH 1;" "
6010 LET j=j0: IF INKEY$<>" " THEN GO TO 6010
6020 PRINT FLASH VAL "1";AT i,j,u$: LET b$=INKEY$
6030 IF b$="" THEN GO TO 6020
6040 IF CODE b$=13 THEN PRINT AT i,j;" ": RETURN
6045 IF CODE b$=14 AND hl=NOT PI THEN GO TO 6300
6047 IF hl=VAL "1" THEN GO TO 6400
6050 IF CODE b$<>12 THEN GO TO 6170
6060 IF j=j0 THEN GO TO 6200
6070 LET j=j-1: PRINT AT i,j: FLASH 1;u$: FLASH 0;" ": LET a$=a$( TO j-j0): GO T
O 6200
6170 IF CODE b$<32 OR CODE b$>127 THEN GO TO 6200
6180 BEEP .005,5: PRINT AT i,j;b$: LET j=j+1: LET a$=a$+b$
6190 IF j=32 THEN PRINT AT i+1,NOT PI;" ": RETURN
6200 IF INKEY$<>" " THEN GO TO 6200
6210 GO TO 6020
6300 LET hl=VAL "1": LET u$="E": BEEP .005,5: GO TO 6200
6400 IF CODE b$=14 THEN LET hl=NOT PI: LET u$=" ": BEEP .005,5: GO TO 6200
6405 LET h$="": GO SUB 6500+CODE b$
6410 IF h$="" THEN GO TO 6200
6415 LET b$=h$: LET hl=NOT PI: LET u$=" ": GO TO 6180
6578 RETURN
6579 LET h$="O": RETURN

```

```
6583 RETURN
6584 LET h$="?": RETURN
6585 LET h$="ü": RETURN
6588 RETURN
6589 LET h$="A": RETURN
6610 RETURN
6611 LET h$="ö": RETURN
6615 RETURN
6616 LET h$="ß": RETURN
6617 LET h$="ü": RETURN
6620 RETURN
6621 LET h$="ä": RETURN
6800 RETURN
7000 DEF FN p(x)=PEEK x+256*PEEK (x+1)
```

Compilador

Los programas redactados en BASIC tienen el inconveniente de que amplias operaciones matemáticas se realizan de forma relativamente lenta. Es mucho más rápido si se ahorra al ordenador el trabajo de traducir del BASIC al lenguaje de máquina. Pero esto es un proceso bastante complejo y no muy sencillo.

De todas formas, también aquí el ordenador puede prestarnos su ayuda si existe un programa adecuado. Un programa de esta índole se denomina 'Compilador' y realiza exactamente lo que antes exponíamos: traduce del BASIC al lenguaje de máquina. El mismo Compilador está escrito, también en lenguaje de máquina.

Primero, el Compilador es cargado en el Spectrum. Después se puede comenzar con la programación en BASIC o cargar un programa ya confeccionado, e inmediatamente después a través de un comando, comenzar la traducción.

Según el tipo de Compilador, el programa traducido se realiza de 5 a 50 veces más rápido que el redactado en BASIC. Generalmente se distingue entre el Compilador que sólo puede traducir números enteros y los que dominan números de coma flotante. Quizás Vd. se pregunte el sentido que tienen los primeros, si existen Compiladores que dominan toda la gama de números.

La solución está en la rapidez. Justamente la representación de números en notación de coma flotante es muy laboriosa. Pero ¿por qué vamos a molestarnos, si esto no es necesario y podemos escribir programas diez veces más rápido?

Si por ejemplo queremos traducir programas de juegos , en los cuales lo que importa es la capacidad de reacción y la representación gráfica, es recomendable un Compilador 'sencillo'. En este caso los puntos ganados se pueden editar en números enteros.

Si por el contrario, queremos traducir un programa de negocios, se aplicará el Compilador más general. ¿Qué negociante quiere prescindir de los céntimos?

Por regla general, los Compiladores no dominan todos los comandos de los que dispone el Spectrum. Por ejemplo, el número de dimensión de un campo está casi siempre limitado a 1. Aunque esto no representa ningún problema, pues con un poco de trabajo se puede representar cualquier campo multidimensional como campo unidimensional, memorizando las dimensiones en serie. El Spectrum realiza esto internamente.

Igualmente, el Compilador no puede ejecutar comandos VAL y saltos calculados. Pero también aquí se puede solucionar el problema a través de CODE CHR\$ y pinitos similares.

Un aspecto agradable al utilizar el Compilador es la posibilidad de traducir varias subrutinas en serie, aunque cada subrutina comience en una dirección diferente. Estas rutinas pueden ser activadas y utilizadas en cualquier programa a través de RAND USR....

CAPITULO 8: REPRESENTACIONES GRAFICAS

Introducción

El Spectrum dispone de muy buenas cualidades gráficas, que son aplicadas en muchos programas de juego. No obstante, en este capítulo no queremos dedicarnos a dibujar cuadros, sino a la representación gráfica de tablas.

Tales programas de negocios son aplicaciones muy extendidas en ordenadores personales, pero también nuestro Spectrum ofrece bastante, y sin gran esfuerzo. Este va a ser el primer programa que tendría aplicación en una empresa pequeña.

Las representaciones de tablas de valores gráficos más agradables son los diagramas de columnas, de círculos o tipo pastel. Elaboremos, pues, un programa con el que representaremos datos en la pantalla.

El programa terminado debe estar capacitado para elegir la escala, memorizar los datos de varios diagramas y efectuar modificaciones.

Desarrollo del programa Busygraph

El esqueleto del programa es en sí bastante sencillo. Son necesarias las siguientes rutinas:

1. Iniciar los diagramas, o sea reservar capacidad de memoria
2. Un menú que utilizaremos paralelamente como índice de contenido para los diferentes diagramas
3. Comando de la representación en la pantalla
4. Realización de los diagramas con columnas
5. Realización de los diagramas con círculos
6. Introducciones y modificaciones
7. Memorización del programa y datos

Comenzemos con la reserva de capacidad de memoria. Nos limitaremos a 15 diagramas, que pueden ser activados paralelamente en el programa. Esto debería ser suficiente. Quien quiera memorizar más diagramas, sólo tiene que memorizar el programa varias veces. O sea:

```
20 DIM t$(15,20)
```

quiere decir, cada título puede tener una longitud de 20 caracteres. Como los títulos aún no están definidos, sigue:

```
30 FOR i=1 TO 15:LET t$(i) = "libre": NEXT i
```

El número de valores que se pueden introducir en un diagrama será 24. Esta magnitud la denominamos dímax.

113

Además definimos la magnitud `dim`, la cual indica la cantidad real de valores en un diagrama, y se memoriza para cada diagrama en la magnitud `a(dim)`. Los valores estarán en `x(dim,15)`. Además necesitamos un vector adicional, en el cual existan los valores a una escala aceptable.

```
40 LET dimax=24: LET dim=dimax: DIM x(dim,15)
50 DIM y(dim): DIM a(dim)
```

Esta parte del programa sólo puede transcurrir una vez, ya que de lo contrario se borrarían todos los datos. Introduzcamos:

```
10 BORDER 0: PAPER 0: INK 6: LET m=9000: GO TO m
```

Esto tiene la ventaja de que en caso de ruptura, el programa puede ser reiniciado con `GO TO 1`, sin perder datos.

Ahora veamos el menú que iniciaremos en la línea 9000.

```
9000 CLS: PRINT TAB 2; INK 3; "B U S Y G R A P H Menu"
9010 FOR i=1 TO 15: PRINT i; TAB 5; +4(i): NEXT i
9020 PRINT AT 20,0; INK 6; "E.....Final"
9030 INPUT INK 4; "¡Por favor introducir un número!"
      ; LINE m$
```

Comprobaremos una cuestión:

```
9110 IF m$="e" THEN GO TO 9900
9120 FOR i=1 TO 15: IF m$=STR$ i THEN GO TO 5000
9130 NEXT i
```

Si este bucle se recorre hasta el final, se ha realizado una falsa introducción y saltamos hacia 9100:

```
9140 GO TO 9100
```

En la línea 9900 se pregunta si los datos han de ser asegurados, y memorizados en cassette o Microdrive como se expuso en el capítulo 4.

En la línea 5000 se elige la forma de representación, o sea, diagrama de columnas o de círculos, y después de dibujarlos se esperan los siguientes comandos. Al elegir la forma de representación, podemos elegir aquella que fue visualizada la última vez. Para ello utilizamos, por motivos de sencillez, una magnitud x0, que es la coordenada de origen del diagrama. En el diagrama de columnas x0 = 40, en el diagrama de círculos 90.

```
5000 LET dím=a(VAL m$)
5010 IF x0<50 THEN GO TO 1000
5020 IF x0>50 THEN GO TO 3000
5100 PAUSE 0: LET n$=INKEY$
5200 IF n$= "m" THEN GO TO m
5300 GO TO 5100
```

Por cuestiones de seguridad deberíamos escribir:

```
60 LET x0 = 40: LET y0=20
```

Para que estas magnitudes estén definidas en el primer inicio.

En la línea 1000 se elabora el diagrama de barras, a partir de la línea 3000 el de tarta. En ambos casos queremos introducir una línea de comando, para recordar las opciones abiertas a este nivel. Como necesitamos esta línea dos veces, la escribimos en una subrutina en la línea 100:

```
100 PRINT INK 4: "'M Columns Círculos Next Last
    Modificar";VAL m$
110 RETURN
```

La M significa naturalmente menú. Next y Last deben dar la opción de ver el primer o último diagrama, sin tener que pasar por el menú e introducir el número correspondiente. La D juntamente con VALm\$ muestra en la segunda línea el número del diagrama actual

Para obtener esta línea de comando escribimos:

```
1000 CLS: GO SUB 100
```

y

```
1010 PRINT AT 2,3; INK 3; +$(VAL m$)
```

Ahora hay que dibujar el diagrama. Primero los margenes:

```
1100 LET x0=40: LET y0=20:
```

```
1110 FOR I=0 TO 1
```

```
1120 PLOT x0+1,y0+1
```

```
1130 DRAW 200,0: DRAW 0,120
```

```
1140 DRAW -200,0: DRAW 0,-120
```

```
1150 NEXT I
```

y una rotulación de la coordenada x:

```
1160 FOR I=INT (1+dím/6) TO dím STEP INT (1+dím/6)
```

```
1170 PRINT AT 20,(x0-8)/8+I/*(192/dím);I: NEXT I
```

Los valores deben estar en las primeras posiciones de la matriz $x(dímax, VAL m\$)$. Los valores ocupados se memorizaron en $a(VAL m\$)$ y en la línea 5010 se traspasan a la magnitud $dím$. Ahora tenemos que encontrar el mayor de estos valores para poder dibujar a una escala adecuada.

```
1180 LET max=0
```

```
1190 FOR I=1 TO dím: IF x(I, VAL m$)>max THEN LET max=
      x(I, VAL m$)
```

```
1200 NEXT I
```

```
1210 LET top=max
```

En caso de que sea $max=0$, no se ha introducido aún ningún valor en este diagrama, y se debería saltar rápidamente a la rutina para modificar o introducir datos. Esta deberá comenzar en la línea 4000.

```
1220 IF max=0 THEN PRINT AT 20,0,, : GO TO 4000
```

Pero si ya existen datos, tenemos que normalizar estos de manera razonable:

```
1230 LET max=100/max
```

```
1240 FOR i=1 TO dim: LET y(i)=max*x(i,VAL m$): NEXT i
```

Aquí es cuando se necesita el vector auxiliar y. Para conseguir un rotulado razonable de la coordenada y, haremos un pequeño truco que funciona bastante bien:

```
1250 LET sc=1: IF top<30 THEN LET sc=10
```

```
1260 FOR i=0 TO 4
```

```
1270 PRINT AT 9+i*3-y0/8,0; OVER 1; INT  
      (sc*(top-I/4*top))/sc
```

```
1280 NEXT i
```

Ahora sólo tenemos que trasladar las barras en su lugar correcto, y esta parte se da por terminada.

```
1300 LET xa=x0+50/dim
```

```
1310 FOR i=1 TO dim
```

```
1320 FOR j=0 TO 125/dim STEP 2
```

```
1330 PLOT xa+j,y0
```

```
1340 DRAW 0,y(i)
```

```
1350 NEXT j
```

```
1360 PLOT xa,y(i)+y0
```

```
1370 DRAW 125/dim,0
```

```
1380 DRAW 0,-y(i)
```

```
1390 LET xa=xa+200/dim
```

```
1400 NEXT i
```

```
1900 GO TO 5100
```

Esto ya está correcto, sólo que aún no podemos comprobarlo, ya que el programa siempre vuelve a saltar a la línea 4000 mientras no hayamos introducido datos.

Antes de que procedamos a la representación de los diagramas de círculos, arreglemos este punto. Primero necesitamos una nueva línea de comando:

```
4000 PRINT AT 0,0; INK 5; "EING:";
4010 PRINT INK 3;M Título borrar valores OK "
4020 PAUSE o: LET n$=INKEY$: IF n$="o" THEN GO TO 5000
4030 IF n$="t" THEN GO TO 4200
4040 IF n$="w" THEN GO TO 4300
4050 IF n$="m" THEN GO TO m
4060 IF n$="l" THEN GO TO 4100
4070 GO TO 4000
```

Ya que no queremos precipitarnos al borrar datos, introducimos en la línea 4100 una cuestión de seguridad:

```
4100 INPUT '¿borrar? S/N';LINE n$
4110 IF n$<>"S" THEN GO TO 4000
4120 FOR i=1 TO dimax: LET x(i,VAL m$)=0; NEXT i
4130 LET a(VAL m$)=1
4140 LET t$(VAL m$)= " libre "
4150 GO TO 5000
```

Ahora tenemos que ocuparnos de la introducción o de la modificación del título, cantidad de valores y de los mismos valores. Comencemos con el título:

```
4200 INPUT 'Introducción: ' LINE n$<>' ' THEN LET
      t$(VAL m$)=n$
4210 PRINT AT 2,0,,AT 2,3; INK 3; t$(VAL m$)
4220 GO TO 4000
```

Después la cantidad de valores:

```
4300 INPUT 'Cantidad de valores? (ENTER='';
      (a(VAL m$));') ' ; LINE n$: IF n$= ""
      THEN GO TO 4340
4310 FOR i=1 TO dimax: IF n$=STR$ i THEN GO TO 4330
4315 NEXT i
```

```

4320 GO TO 4000
4330 LET a(VAL m$)=VAL n$
4340 LET dím=a(VAL m$)
4350 INPUT "'Alterar valores? S/N'"; LINE n$: IF n$<>
    "S" THEN GO TO 4000
4360 FOR i=1 TO a(VAL m$)
4370 INPUT "valores";(i);"ENTER="; (x(i,VAL m$));
    "E=final"; LINE n$: IF n$=" " THEN GO TO 4400
4380 IF n$="e" THEN GO TO 5000
4390 LET x(i,VAL m$)=VAL n$
4400 NEXT i

```

Aquí habría que prever si el valor ha sido introducido correctamente. Pero esto lo dejamos a deseo del lector.

Ahora vamos a elaborar los diagramas de tarta. Si aún no han sido introducidos valores, no puede ser dibujado naturalmente ningún diagrama.

```

3000 LET sum=0
3010 FOR i=1 TO dím
3020 LET sum=sum+x(i,VAL m$)
3030 NEXT i
3040 IF sum=0 THEN GO TO 1000
3050 CLS: GO SUB 100
3060 PRINT AT 2,3; INK 3; +$(VAL m$)

```

Dibujemos ahora un círculo:

```

3100 LET x0=90: LET y0=72
3110 CIRCLE x0,y0,y0

```

Ahora normalizamos los valores en el vector auxiliar y(dím):

```

3120 FOR i=1 TO dím: LET y(i)=x(i,VAL m$)/sum: NEXT i

```

Para dibujar los sectores de circunferencia, deberemos recordar la trigonometría de las clases de matemáticas:

```

3200 LET win=0
3210 LET xp=22-y0/8: LET yp=x0/8
3220 LET u=2*PI*y0

```

Para posicionar correctamente las denominaciones de los segmentos, introducimos el valor scal:

```

3230 LET scal=.9

```

Ahora vamos a hacer un bucle con todos los valores y dibujar cada vez el centro del círculo y una línea con un determinado ángulo respecto a la línea anterior:

```

3400 FOR i=1 TO dim
3410 PLOT x0,y0
3420 LET win=win+y(i)
3430 LET x1=y0*COS(2*PI*win)
3440 LET x2=scal*y0COS (2*PI*(win-y(i)/2))
3450 LET y1=y0*SIN (2*PI*win)
3460 LET y2=scal*y0SIN (2*PI*(win-y(i)/2))
3470 DRAW x1,y1
3480 PRINT AT 21-y0/8-y2/8,x0/8-1+x2/8; INVERSE 1;1
3490 NEXT i

```

Ya que al dibujar el diagrama circular aún nos queda espacio en la parte izquierda de la pantalla, visualizamos la repartición porcentual de los valores sobre la suma total de todos los valores, así como el valor de la media aritmética:

```

3500 PRINT AT 3,23; INK 5; "Valores(%)"
3510 FOR i=1 TO dim STEP 2: PRINT AT (i+6)/2,21;
    INK 5;1;
3520 PRINT TAB 23; INT (1000*y(i))/10;
3530 IF (i+1)<=dim THEN PRINT TAB 28;
    INT (1000*y(i+1))/10
3540 NEXT i
3550 PRINT AT 17,20; "Suma"; AT 18,20;
    INT (100*sum)/100
3560 PRINT AT 20,20; "'valor media aritmética'";
    AT 21,20; INT(100*sum/dim)/100
3900 GO TO 5100

```

Ya habríamos terminado, si no hubiésemos olvidado un pequeño detalle.

Después de haber dibujado un diagrama, el programa salta a la línea 5100. Aquí hemos previsto el retorno al menú, pero no todavía para todas las opciones. Realicemos este punto rápidamente:

```
5210 IF n$="b" THEN GO TO 1000
5220 IF n$="k" THEN GO TO 3000
5230 IF n$="n" AND VAL m$<15 THEN LET m$=
      STR$ (VAL m$+1)
5240 IF n$="l" AND VAL m$>1 THEN LET m$=
      STR$ (VAL m$-1)
5250 IF n$="n" OR n$="l" THEN GO TO 5000
5260 IF n$="a" THEN GO TO 4000
5900 GO TO 5100
```

Para iniciar el programa la primera vez, o para más tarde borrar todos los valores, introducimos GO TO 20. Para sucesivas veces introducimos GO TO 1.

Trabajar con el programa

Ahora vamos a tratar de conocer el programa un poco mejor (y mejorarlo). Introduzcamos, después de asegurarlo, GO TO 20. Si todo funciona, debería surgir en la pantalla:

B U S Y G R A P H Menu

```
1     libre
2     libre
3     libre
4     libre
5     libre
6     libre
7     libre
8     libre
9     libre
10    libre
11    libre
12    libre
13    libre
14    libre
15    libre
```

E.....final

Por favor introducir un número!

Tecleemos ahora '1 ENTER'. Primero surge la línea de comando

M columnas círculo Next Last Modificar

y

D1

libre

y los márgenes laterales para el diagrama. Pero como todos los valores están a 0, no es posible dibujar ningún diagrama, y el programa sabe que ahora queremos componer un nuevo diagrama. Por ello indica rápidamente la línea de comando nueva:

Entradas: Menu Borrar Título Valores OK

Si no queremos introducir datos, volvemos a través de m al menú.

Si por lo contrario introducimos 'T', podemos introducir un título, por ejemplo Test. Esto sustituye rápidamente el 'libre' que existía anteriormente. Ahora tecleamos 'W', y surge la pregunta por la cantidad de valores. Si ésta debe quedarse igual, no sólo tenemos que introducir ENTER, sino un número entre 1 y 24. Introduzcamos un '10'.

Ahora podemos indicar si queremos cambiar valores y contestamos con 'S'. Introduzcamos los valores 1, 2, 3... hasta 10 y esperemos el resultado. Después de la última introducción, el diagrama es automáticamente dibujado.

Ahora podemos observar estos valores en forma de diagrama circular pulsando la letra 'K'. Con esta opción se indican además los repartos porcentuales de cada valor, así como la suma y el valor de la media aritmética.

Si queremos modificar o añadir un valor, lo efectuaremos pulsando 'A'. Dejemos la cantidad de valores sin modificar, pero cambiemos el primer número por 20. A continuación tecleemos y el nuevo diagrama está terminado.

Lo que molesta algo en el programa, es que no podemos visualizar todos los valores simultáneamente, y un valor falso sólo puede ser corregido cuando la introducción se ha terminado por medio de e, o se ha alcanzado el último valor.

Pero esto es fácilmente reparable.

Interrumpamos el programa con BREAK e introduzcamos:

```
200 FOR i=5 TO 17: PRINT AT i,6;"
    NEXT i: RETURN
```

Esto origina un campo de 13 líneas por 23 columnas.

Ahora cambiamos la línea 3450:

```
4350 INK 0: PAPER 7: GO SUB 200: PRINT AT 5,6;
    "TABLA DE VALORES"
```

y entonces:

```
4352 FOR i=1 TO dim: IF i<13 THEN PRINT AT i+5,6;i;
    TAB 9;x(i,VAL m$)
4353 IF i>12 THEN PRINT AT i-7,18;i;TAB 22;x(i,VAL m$)
4354 NEXT i
4356 NEXT 6

4380 IF n$="e" THEN PAPER 0: INK 6: GO TO 5000
4382 IF n$="l" AND i>1 THEN LET i=i-1: GO TO 4370
4384 IF CODE n$>57 THEN GO TO 4370

4392 IF i<13 THEN PRINT AT i+5,9 PAPER 7;"    ";
    AT i+5,9;INK 0;x(i,VAL m$)
4394 IF i>12 THEN PRINT AT i-7,22;PAPER 7;"    ";
    AT i-7,22;INK 0;x(i,VAL m$)
4396 PAPER 0
```

Por favor, asegure Vd. esta versión. Si después quiere realizar otra modificación o introducción de valores, surge una tabla de valores que respresenta todo con bastante claridad. Si la alteración se realiza sobre un diagrama circular (o de tarta), se asemeja a la técnica de ventana.

A través de L podemos volver al último valor introducido. Al observar los diagramas, 'L' o 'N' originan el paso al último o siguiente diagrama respectivamente, sin tener que volver al menú.

El programa tiene mucho espacio para ampliaciones. Por ejemplo, Vd. puede introducir la posibilidad de representación por medio de diagramas de líneas. Si lo necesita, puede representar también números negativos. Si es muy ambicioso, introduzca la posibilidad de representar dos diagramas a la vez, o sumar dos diagramas y memorizar el diagrama resultante.

Al final de este capítulo veamos el listado completo del programa.

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****

10 BORDER 0: PAPER 0: INK 6: LET m=9000: GO TO m
20 DIM t$(15,20)
30 FOR i=1 TO 15: LET t$(i)="libre": NEXT i
40 LET dimax=24: LET dim=dimax: DIM x(dim,15)
50 DIM y(dim): DIM a(dim)
60 LET x0=40: LET y0=20
90 GO TO 1
100 PRINT INK 4;"M Barras Tarta Prox. Ant. Modif.":VAL m
110 RETURN
200 FOR i=5 TO 17: PRINT AT i,6;"
NEXT i: RETURN
1000 CLS : GO SUB 100
1010 PRINT AT 2,3: INK 3;t$(VAL m$)
1100 LET x0=40: LET y0=20
1110 FOR i=0 TO 1
1120 PLOT x0+i,y0+i
1130 DRAW 200,0: DRAW 0,120
1140 DRAW -200,0: DRAW 0,-120
1150 NEXT i
1160 FOR i=INT (1+dim/6) TO dim STEP INT (1+dim/6)
1170 PRINT AT 20,(x0-8)/8+i/8*(192/dim);i: NEXT i
1180 LET max=0
1190 FOR i=1 TO dim: IF x(i,VAL m$)>max THEN LET max=x(i,
L m$)
1200 NEXT i
1210 LET top=max
1220 IF max=0 THEN PRINT AT 20,0,,: GO TO 4000
1230 LET max=100/max
1240 FOR i=1 TO dim: LET y(i)=max*x(i,VAL m$): NEXT i
1250 LET sc=1: IF top<30 THEN LET sc=10
1260 FOR i=0 TO 4
1270 PRINT AT 9+i*3-y0/8,0: OVER 1;INT (sc*(top-i/4*top))/
1280 NEXT i
1300 LET xa=x0+50/dim

```

```

1310 FOR i=1 TO dim
1320 FOR j=0 TO 125/dim STEP 2
1330 PLOT xa+j,y0
1340 DRAW 0,y(i)
1350 NEXT j
1360 PLOT xa,y(i)+y0
1370 DRAW 125/dim,0
1380 DRAW 0,-y(i)
1390 LET xa=xa+200/dim
1400 NEXT i
1900 GO TO 5100
3000 CLS : GO SUB 100: REM diagrama de tarta
3005 LET sum=0
3010 FOR i=1 TO dim
3020 LET sum=sum+x(i,VAL m$)
3030 NEXT i
3040 IF sum=0 THEN GO TO 1000
3050 CLS : GO SUB 100
3060 PRINT AT 2,3: INK 3;t$(VAL m$)
3100 LET x0=90: LET y0=72
3110 CIRCLE x0,y0,y0
3120 FOR i=1 TO dim: LET y(i)=x(i,VAL m$)/sum: NEXT i
3200 LET win=0
3210 LET xp=22-y0/8: LET yp=x0/8
3220 LET u=2*PI*y0
3230 LET scal=0.9
3400 FOR i=1 TO dim
3410 PLOT x0,y0
3420 LET win=win+y(i)
3430 LET x1=y0*COS (2*PI*win)
3440 LET x2=scal*y0*COS (2*PI*(win-y(i)/2))
3450 LET y1=y0*SIN (2*PI*win)
3460 LET y2=scal*y0*SIN (2*PI*(win-y(i)/2))
3470 DRAW x1,y1
3480 PRINT AT 21-y0/8-y2/8,x0/8-1+x2/8: INVERSE 1;i
3490 NEXT i
3500 PRINT AT 3,21: INK 5;"VALORES (%)"
3510 FOR i=1 TO dim STEP 2: PRINT AT (i+6)/2,21: INK 5;i;
3520 PRINT TAB 23:INT (1000*y(i))/10;
3530 IF (i+1)<=dim THEN PRINT TAB 28:INT (1000*y(i+1))/10
3540 NEXT i
3550 PRINT AT 17,19;"SUMA":AT 18,19:INT (100*sum)/100

```

```

3560 PRINT AT 20,19;"MEDIA";AT 21,19;INT (100*(sum/dim))/100
3900 GO TO 5100
4000 PRINT AT 0,0; INK 5;"Intr:";
4010 PRINT INK 4;"M Titulo Valores Borrar OK"
4020 PAUSE 0: LET n$=INKEY$: IF n$="o" THEN GO TO 5000
4030 IF n$="t" THEN GO TO 4200
4040 IF n$="v" THEN GO TO 4300
4050 IF n$="m" THEN GO TO m
4060 IF n$="b" THEN GO TO 4100
4070 GO TO 4000
4100 INPUT "BORRAR ? (s/n) "; LINE n$
4110 IF n$<>"s" THEN GO TO 4000
4120 FOR i=1 TO dimax: LET x(i,VAL m$)=0: NEXT i
4130 LET a(VAL m$)=1
4140 LET t$(VAL m$)="libre"
4150 GO TO 5000
4200 INPUT "ENTRADA: "; LINE n$: IF n$<>" " THEN LET t$(VAL
m$)=n$
4210 PRINT AT 2,3,,;AT 2,3; INK 3;t$(VAL m$)
4220 GO TO 4000
4300 INPUT "Numero de valores ? (ENTER=";(a(VAL m$));")"; L
INE n$: IF n$="" THEN GO TO 4340
4310 FOR i=1 TO dimax: IF n$=STR$ i THEN GO TO 4330
4315 NEXT i
4320 GO TO 4000
4330 LET a(VAL m$)=VAL n$
4340 LET dim=a(VAL m$)
4350 INK 0: PAPER 7: GO SUB 200: PRINT AT 5,6;"TABLA DE VALO
RES"
4352 FOR i=1 TO dim: IF i<13 THEN PRINT AT i+5,6;i;TAB 9;x(
i,VAL m$)
4353 IF i>12 THEN PRINT AT i-7,18;i;TAB 22;x(i,VAL m$)
4354 NEXT i
4356 INK 6
4360 FOR i=1 TO a(VAL m$)
4370 INPUT "valor ";(i);" ENTER=";(x(i,VAL m$));" F=Fin "; L
INE n$: IF n$="" THEN GO TO 4400
4380 IF n$="f" THEN PAPER 0: INK 6: GO TO 5000
4382 IF n$="b" AND i>1 THEN LET i=i-1: GO TO 4370
4384 IF CODE n$>57 THEN GO TO 4370
4390 LET x(i,VAL m$)=VAL n$
4392 IF i<13 THEN PRINT AT i+5,9; PAPER 7;" ";AT i+5,

```

```

9; INK 0;x(i,VAL m$)
4393 IF i>12 THEN PRINT AT i-7,22; PAPER 7;"      ";AT i-7
,22; INK 0;x(i,VAL m$) ,
4394 REM
4395 REM
4396 PAPER 0
4400 NEXT i
5000 LET dim=a(VAL m$)
5010 IF x0<50 THEN GO TO 1000
5020 IF x0>50 THEN GO TO 3000
5100 PAUSE 0: LET n$=INKEY$
5200 IF n$="m" THEN GO TO m
5210 IF n$="b" THEN GO TO 1000
5220 IF n$="t" THEN GO TO 3000
5230 IF n$="p" AND VAL m$<15 THEN LET m$=STR$ (VAL m$+1)
5240 IF n$="a" AND VAL m$>1 THEN LET m$=STR$ (VAL m$-1)
5250 IF n$="p" OR n$="a" THEN GO TO 5000
5260 IF n$="v" THEN GO TO 4000
5300 GO TO 5100
9000 CLS : PRINT TAB 2; INK 3;"B U S Y G R P H  Menu""
9010 FOR i=1 TO 15: PRINT "  ";i;TAB 5;t$(i): NEXT i
9020 PRINT AT 20,2; INK 6;"F.....Fin"
9100 INPUT INK 4;"Introduzca un numero!  "; LINE m$
9110 IF m$="f" THEN GO TO 9990
9120 FOR i=1 TO 15: IF m$=STR$ i THEN GO TO 5000
9130 NEXT i
9140 GO TO 9100
9990 CLS : PRINT AT 9,7;"Asegurar datos ? (s/n)": PAUSE 0: L
ET m$=INKEY$: IF m$="s" THEN GO TO 9996
9995 CLS : STOP
9996 REM CLS : ERASE "m";1;"Busygrph"
9997 REM SAVE # "m";1;"Busygraph" LINE 1
9998 REM VERIFY # "m";1;"Busygraph"

```

CAPITULO 9:
EL TABLON DE ANUNCIOS:
PUBLICIDAD CON EL SPECTRUM

Introducción

Lo más probable es que Vd. haya echado un vistazo al cassette de demostración que acompañaba a su Spectrum. Contiene algunos programas escritos sobre todo en BASIC y que pueden ser listados fácilmente. Pero ahora no vamos a tratar sobre estos programas en BASIC, sino a dirigir nuestro interés a las grandes letras que surgen en la pantalla.

En el cuarto capítulo vimos ya una pequeña rutina, con la que pueden representarse caracteres a doble tamaño. Si utilizamos este sistema para letras aún más grandes, el programa será muy lento.

Es más rápido el lenguaje de máquina, ya que así evitamos al ordenador la labor de traducción del BASIC. Por ello, la rutina en el cassette para la representación de caracteres grandes, está escrita en código máquina. La rutina es, como todos los programas en cassette, de la empresa PSION. Pero no hay nada en contra de utilizar esta rutina, ya que está a nuestra disposición.

Podemos usarla y, junto con otros efectos, conseguir un programa agradable, con el cual se pueden representar comunicados en la pantalla. Esto por ejemplo puede emplearse para comunicar a la familia que no le esperen a Vd. para la cena. Quizás podría Vd. estrenar su último chiste en una fiesta ante sus invitados. Ciertamente se pueden hacer muchas tonterías con este programa.

Naturalmente, un propietario puede colocar un televisor en el escaparate, e informar a sus clientes sobre sus ofertas especiales. Si esto se hace con mucho movimiento y colorido, seguro que resultará efectivo.

El programa debería pues, permitir introducir y memorizar texto, y representarlo en la pantalla a tamaño, colores y tiempo según voluntad. Denominemos al programa AWIS.

Un punto importante en el programa es la rutina de caracteres grandes. Tomemos un programa de la PSION, por ejemplo 'WALL', que es el primero en la cara B del cassette. Tras Interrumpir el programa por medio de BREAK, vemos que después de un CLEAR 31999, el código 'c' ha sido cargado en las direcciones 32000 hasta 3339.

Podemos memorizar este código en cassette o en Microdrive. Esto debería realizarse con el comando SAVE "c" CODE 32000,340. Igualmente, si Vd. dispone de un Spectrum de 48 K, debe encontrarse el código en estas direcciones y al activar esta rutina no se ha de armar un caos. En la mencionada rutina se encuentran saltos hacia direcciones absolutas. Si en estas direcciones no se encuentra el código correcto, pasan cosas desagradables. Por regla general, hay que desconectar al buen Spectrum, ya que no quiere volver al funcionamiento en BASIC. Esto, naturalmente, puede evitarse si se cambian las direcciones correspondientes. Entonces se puede correr la rutina hacia atrás, y disponer de más espacio para el programa BASIC.

Nosotros vamos a dejar la versión tal como está y a componer el programa restante de la manera más económica, de forma que se pueda memorizar el máximo de texto. Esto tiene la ventaja de que los propietarios de 16 K también tendrán su diversión. La rutina funciona de la siguiente manera: El texto se introduce en la cadena p\$. Después se fija la magnitud yy, que indica el número de línea en la pantalla. Se determinan los valores para xs y ys. Estos son factores de escala para el tamaño de la escritura en las coordenadas x e y.

Escribamos el siguiente programa de pruebas:

```
10 CLEAR 31999
20 LOAD "c" CODE 32000,340
30 INK 0: BORDER 6: PAPER 6:CLS
40 LET p$="HOLA"
50 LET yy=10
60 LET xs=2:LET ys=4
70 GO SUB 3500
90 PAUSE 0:STOP
```

Ahora sólo tenemos que llegar a la subrutina en la línea 3500. Podemos tomarla del programa 'WALL':

```
3500 LET xx=(256-8*xs*LEN p$)/2
3510 LET I=23306: POKE I,xx: POKE I+1,yy: POKE I+2,xs
3520 POKE I+3,ys: POKE I+4,8
```

En la línea 3500 se calcula el número de columna de manera que el texto es centrado exactamente en la línea. Después, las magnitudes xx,yy,xs e ys son escritas en las direcciones 23306. Desde aquí pueden ser leídas y elaboradas por la rutina del código máquina. Las direcciones desde 23296 son la memoria intermedia para la Impresora, y pueden ser utilizadas aquí sin ningún peligro.

Por ello, también los códigos de los caracteres de la cadena p\$ (o sea el texto que queremos escribir) son introducidos mediante 'POKE' en esta memoria.

```

3530 LET i=i+4: LET w=LEN p$
3540 FOR n=1 TO w: POKE i+n, CODE p$(n): NEXT n
3550 POKE i+w+1,255
3560 LET w=USR 32000
3570 RETURN

```

La línea 3560 activa la rutina en código máquina. Si todo está correctamente, en la pantalla debe surgir, después de RUN, un saludo en caracteres grandes. La PAUSE 0 se ha introducido para que, después del trabajo realizado, no surja ningún comunicado en la pantalla.

Si hasta aquí todo va bien, Vd. debería experimentar un poco con los parámetros xs,ys, etc. Introduzca otros textos y fíjese qué es lo que pasa cuando el texto es demasiado ancho y no cabe en una línea. La rutina está muy bien elaborada. ¡No puede llegarse a Crash alguno!

El desarrollo del programa

Ahora podemos empezar a construir un programa completo basándonos en estos incios. Dejamos la subrutina a partir de la línea 3500; sólo las primeras líneas han de borrarse. Comencemos de forma habitual:

```
10   LET big=3500: LET m=9000: GO TO m
```

y

```
9000  INK 0: BORDER 6: PAPER 6: CLSI
9010  LET p$="AWIS-MENU": LET yy=10: LET xs=2: LET ys=4
9020  GO SUB big
9030  PRINT AT 9,3;''1 = Preparación de nueva
      publicidad''
9040  PRINT    '' '2 = Prueba y/o cambiar''
9050  PRINT    '' '3 = Dejar correr publicidad''
9060  PRINT    '' '4 = F I N A L ''
9070  PRINT AT 21,2; ''Por favor introducir opción
      deseada''
9100  PAUSE 0: LET m$=INKEY$
9110  IF m$= "1" THEN GO TO 1000
9120  IF m$= "2" THEN GO TO 5000
9130  IF m$= "3" THEN GO TO 2000
9140  IF m$= "4" THEN GO TO 9900
9150  GO TO 9100
```

Ahora vamos a fijar las posibilidades que debe ofrecer el programa. Vamos a elaborarlo de tal manera que podamos producir determinados dibujos con la introducción. Cada dibujo está compuesto por una determinada cantidad de líneas, cada línea compuesta por 30 caracteres. Cada línea puede representarse al tamaño deseado. También debe ser libremente elegible el color de la escritura, el fondo y los márgenes laterales. Entre la visualización de línea y línea se ha de poder hacer una pausa de duración indefinida. Cada línea debe ser intermitente si así se desea, y cuando una línea esté terminada, el margen deberá cambiar rápidamente de color.

Para llevar a cabo estas posibilidades fácilmente, y poderlas modificar en todo momento, vamos a asignar a cada línea de texto que esté memorizada en un campo a\$, otro campo q\$. Para cada línea de texto existirán entonces 10 magnitudes, que fijan el tipo de representación durante el transcurso del programa. Detallemos este campo q\$, para tener un resumen claro:

- q\$(1) - si con esta línea se ha de iniciar una nueva imagen 0 ó 1
- q\$(2) - Memoriza en forma de código la longitud de la línea de texto
- q\$(3) - Memoriza en forma de código el número de línea
- q\$(4) - Indica el tamaño de la escritura
- q\$(5) - Si la línea ha de ser parpadeante 0 ó 1
- q\$(6) - Indica la longitud de la pausa, antes de surgir la próxima línea
- q\$(7) - Indica el color de los márgenes laterales, durante la duración de esta línea
- q\$(8) - Indica el color del fondo
- q\$(9) - Indica el color de la escritura
- q\$(10) - Si el margen lateral debe visualizar todos los colores 0 ó 1

Además de estas posibilidades, queremos prever la posibilidad de poder introducir o borrar texto en cualquier parte. También debe ser posible la intervención durante el transcurso y la introducción de modificaciones. Cada una de estas opciones se dispone en una subrutina. Estas rutinas las denominamos, por cuestiones de facilidad, con la primera letra de la labor que han de realizar.

Introduzcamos:

```
80 LET nb=100: LET te=100: LET ze=120: LET gr=130:  
   LET bl=140
```

```
82 LET pa=150: LET fa=160: LET va=170
```

```
90 RETURN
```

y después

```
5 GO SUB 80
```

Con estas ayudas casi podemos programar las líneas, sin haber pensado aún en la estructura de las mismas. A partir de ahora vamos a intentar ocupar la memoria en lo mínimo posible, para que puedan ser memorizadas la máxima cantidad de líneas a la vez. Esto significa que en lugar del 0, aparecerá frecuentemente un NOT PI, o el valor 1 expresado a través de VAL "1". Por este mismo motivo no se va a cuestionar en cada introducción, si esta es legal o no. Quien quiera introducir tales cuestiones, puede hacerlo ya que generalmente, el programa es fácilmente modificable a voluntad.

Antes de comenzar, deberíamos introducir la línea:

```
70 LET I$=" "
```

I\$ representa una cadena con 30 espacios en blanco, y posibilita la cuestión, si en un línea a\$(i) ya se ha introducido texto. Ahora vamos a ver la parte del programa que manda la representación de las líneas:

```
2000 CLS : IF a$(1)=I$ THEN GO TO m
```

Si no se ha introducido texto, se vuelve directamente al menú.

```
2010 LET anf = VAL "1": LET end = d
```

d se actualiza después según el número de líneas.

```
2020 PRINT " " Todas las líneas? S/N "": PAUSE NOT  
PI: LET z$=INKEY$: CLS      2030 IF z$<> "N"  
THEN GO TO 2100
```

Esta pregunta permite visualizar determinadas líneas, desde anf hasta end.

```
2040 INPUT "Primera línea = "; LINE z$: IF z$= " " "  
THEN LET z$= "1"  
2050 LET anf = VAL z$: IF anf>d THEN GO TO 2040  
2060 INPUT " última línea = ? " ";LINE z$: IF z$=  
" " THEN LET z$=STR$ d  
2070 LET end= VAL z$: IF end>d THEN GO TO 2060  
2080 IF anf>end THEN GO TO 2000
```

Aquí comienza el bucle con todas las líneas:

```
2100 FOR z= anf TO end
```

que ha de finalizar en la línea 2500.

```
2500 NEXT z
```

Para poder intervenir en el transcurso del programa, que aquí tiene un bucle indeterminado, introducimos:

```
2110 IF INKEY$ = "m" THEN GO TO m
```

De esta manera, pulsando la tecla m volvemos al menú.

```
2070 LET end=VAL z$: IF end>d THEN GO TO 2060  
2080 IF anf>end THEN GO TO 2000
```

Ahora comienza en bucle con todas las líneas:

```
2100 FOR z=anf TO end
```

que ha de finalizar en la línea 2500

```
2500 NEXT z
```

Para poder intervenir en el transcurso del programa, que aquí tiene un bucle indeterminado, introducimos:

```
2110 IF INKEY$="m" THEN GO TO m
```

De esta manera, pulsando la tecla m volvemos al menú.

```
2120 IF a$(z) = 1$ THEN GO TO 2100
```

Cuando se ha llegado a la primera línea vacía, comienza de nuevo el juego.

Por fin podemos empezar con la representación de cada línea. Primero vamos a fijar los colores:

```
2130 BORDER VAL q$(7,z): PAPER VAL q$(8,z): INK VAL  
q$(9,z)
```

Después se determina la línea, en la cual se visualiza texto:

```
2200 LET yy=8*CODE q$(3,z)
```

y la escritura

```
2210 GO SUB 135
```

Salamos hacia la subrutina 'gr', ya que en la escritura se calculan dos parámetros: xz e ys. Ya que esto también se efectúa al introducir una línea, podemos evitarnos la introducción de estas líneas de programa.

Ahora debemos traer el texto al campo p\$, ya que nuestra rutina exige esto. La longitud del texto esta en q\$(2,z), o sea:

```
2220 LET p$=a$(z,1 TO CODE q$(2,z))
2230 IF q$(1,z)="1" THEN CLS
```

borrar pantalla, si se exige nueva imagen y

```
2240 FLASH VAL q$(5,z)
```

o sea, cese del parpadeo.

Ahora nos vamos a la subrutina 'big', para editar el texto

```
2300 GO SUB big
```

y evitamos un posible parpadeo de la pantalla:

```
2310 FLASH NOT PI
```

Ahora preguntamos si el margen lateral ha de cambiar a todos los colores

```
2320 IF q$(10,z)="1" THEN FOR k=1 TO 7: BORDER k:
      PAUSE 50: NEXT k
```

Ahora se fija la duración de la pausa. Esta viene dada a través de q\$(6,z). Si CODE q\$(6,z)=0, la línea se quedaría visualizada hasta que fuese pulsada una tecla. Esto queremos evitarlo. o sea:

```
2350 IF CODE q$(6,z)=0 THEN LET Q$(6,z)= CHR$ 1
2400 PAUSE 50* CODE q$(6,z)
```

Deberíamos introducir dos posibilidades en este punto, que permiten que el programa sea más confortable: Primero deberíamos poder interrumpir el programa, para poder observar una imagen el tiempo deseado y, por otra parte, facilitar modificaciones. Esto se realiza fácilmente a través de los comandos:

```

2410 IF INKEY$="p" THEN PRINT AT NOT PI,NOT PI;"PAUSE"
      NOT PI: PRINT AT NOT PI,NOT PI;"  ": GO TO 2410

```

Si queremos saber el número de la línea, podemos ver este si pulsamos la tecla z, introduciendo:

```

2420 IF INKEY$="z" THEN PRINT AT NOT PI, NOT PI;
      "Línea";z:PAUSE NOT PI: PRINT AT NOT PI,
      NOT PI; "      ": GO TO 2410

```

Ahora la posibilidad de cambiar una línea:

```

2430 IF INKEY$ ="k" THEN GO TO 5200

```

La línea '2500 NEXT z' ya la hemos introducido. Ahora introducimos:

```

2600 GO TO 2100

```

y el proceso sin fin esta asegurado.

Vamos a ocuparnos de la entrada de textos, para que por fin se vea algo en la pantalla. Primero nos aseguraremos de si hay que borrar algunas líneas, para que no ocurra por descuido:

```

1000 CLS : PRINT''' "¿Borrar todas las líneas? S/N"
      : PAUSE NOT PI:IF INKEY$ ="S" THEN GO TO 1100
1010 CLS : LET flag=1: GO TO 1200

```

Si la bandera = 1, no se debe sobrecribir línea alguna.

```

1100 CLEAR : GO SUB 20

```

A partir de línea 20 se colocan los valores iniciales para todas las líneas. Ahora iniciamos un bucle a través de todas las líneas posibles:

```
1200 FOR z=VAL "1" TO d
1210 IF flag=1 AND a$(z)<>1$ THEN GO TO 1800
1800 NEXT z
```

o sea, se busca la primera línea en blanco (en la línea 1800 esta escrito NEXT z).

Ahora se efectúan todas las preguntas, para lo cual cada vez se activa una subrutina. Gran parte de estas rutinas se han previsto con nombre en la línea 80.

```
1220 GO SUB nb
```

Esta es la pregunta para iniciar una nueva imagen o no. Ahora vamos a introducir el texto. Para ello escribiremos, en la parte izquierda de la pantalla, el número de líneas; de esta manera es más fácil la orientación. Esto se efectúa con una rutina a partir de la línea 200, que aún ha de ser escrita, pero que es muy sencilla:

```
1230 GO SUB 200: GO SUB te
```

Si se introduce una línea en blanco, significa que la introducción se ha finalizado:

```
1240 IF CODE q$(2,z)=0 THEN GO TO m
1250 LET a$(z,1 TO LEN t$)=t$
1260 GO SUB ze: GO SUB gr
```

Ahora escribimos esta línea. Para no interrumpir el transcurso se efectúa a través de:

```
1280 GO SUB 500
```

A partir de línea 500 se encuentra una rutina de escritura, que activa también la rutina big. Los parámetros más importantes se han introducido, y ahora se ha de preguntar si se han de cambiar algunas magnitudes. Para ello utilizamos una pequeña rutina, que se llamará 'pregunta':

```
1290 GO SUB pregunta
```

Todo esto se finaliza con

```
1800 NEXT z
1900 GO TO m
```

como ya se han mencionado. Aún faltan algunos detalles, pero los introduciremos más tarde.

Ahora vamos a completar la inicialización, o sea a reservar la memoria, o borrar, y asignar valores iniciales más o menos razonables a los parámetros.

```
20 LET d=25
30 DIM q$(10,d): DIM a$(d,30)
40 FOR k=1 TO d: LET q$(4,k)="1": LET q$(6,k)=
  CHR$ 5:LET q$(7,k)="6":LET q$(8,k)="6":
  LET q$(9,k)="0"
50 LET q$(1,k)="0": LET q$(5,k)="0": LET q$(10,k)="0"
60 LET big=3500: LET m=9000: LET flag=NOT PI
84 LET pregunta=300
```

De esta forma se han previsto los colores -negro para la escritura y amarillo para el fondo- y los márgenes laterales. Esto, como todas las demás magnitudes, se puede cambiar; sólo que hay que indicarlo. Además se ha previsto 'ninguna nueva imagen', 'ningún parpadeo', pausa de 5 segundos y ninguna variación de los márgenes laterales.

Ahora tenemos el deber de elaborar las diferentes subrutinas, que permitan introducir todos los parámetros.

Comencemos con la rutina 'pregunta':

```
300 PRINT AT 21,NOT PI; INK NOT PI; PAPER VAL "7";  
"T U H F Z B G P N E V Okay"  
310 PAUSE NOT PI: LET n$=INKEY$  
312 IF n$="u" OR n$= "h" OR n$= "í" THEN GO TO fa
```

todos los colores se tratan conjuntamente.

```
313 IF n$="z" THEN GO TO ze  
314 IF n$="b" THEN GO TO bl  
315 IF n$="p" THEN GO TO pa  
316 IF n$="n" THEN GO TO nb  
317 IF n$="e" THEN GO TO 800  
318 IF n$="v" THEN GO TO va  
319 IF n$="o" THEN RETURN  
320 IF n$="t" THEN GO TO te  
321 IF n$="g" THEN GO TO gr  
330 GO TO pregunta
```

Según la tecla que sea pulsada, saltamos hacia la subrutina correspondiente. Si todo está en orden, volvemos a la parte del programa 'Componer'. Así aseguramos poder pasar a elaborar la siguiente línea. Por ello introducimos

```
1320 IF n$="o" THEN GO TO 1800  
1330 GO TO 1280
```

Ahora tenemos que ocuparnos de la introducción de los diferentes parámetros. Comencemos con la rutina 'nb', o sea la pregunta, si con la línea se ha de iniciar una nueva imagen. Esta comienza con la línea 100.

```
100 LET q$(1,z)="0": PRINT AT 21, NOT PI: "¿Nueva  
Imagen S/N (0=Menú)"; PAUSE NOT PI: LET f$=INKEY$  
: IF f$="S" THEN LET q$(1,z="1": CLS  
102 IF z=VAL "1" THEN GO TO 1066  
104 IF f$="N" THEN let q$(6,z-1)="1"  
106 IF f$<> "S" AND f$<>"n" THEN GO TO m
```

```
108 RETURN
```

Ahora la Introducción del texto:

```
110 LET q$(2,z)=CHR$ 0:PRINT AT 21, NOT PI;"texto ="  
  ,,: INPUT LINE t$  
112 IF LEN t$> 80 THEN GO TO 110  
114 LET q$(2,z)=CHR$ (LEN t$)  
116 RETURN
```

Ahora la pregunta por el número de línea:

```
120 GO SUB 200: PRINT AT 21,NOT PI;"Línea (1-20)="'  
  ,:INPUT LINE n$
```

Esto escribe los números de línea en la parte izquierda de la pantalla, y luego, si Vd. desea, esta seguridad:

```
122 FOR s=VAL "1" TO 20: IF n$=SRT$ s THEN GO TO 128  
124 NEXT s  
126 GO TO 120
```

y

```
128 LET q$(3,z)= CHR$ (VAL,n$):RETURN
```

A partir de la línea 130 comienza la rutina, la cual determina el tamaño de la escritura:

```
130 PRINT AT 21,NOT PI;"Tamaño de la escritura(1-9)''  
  ,: LET f$=INKEY$: IF f$>"9" OR f$<"1" THEN GO TO 130  
132 LET q$(4,z)=CHR$ VAL f$  
135 LET ys=CODE q$(4,z): LET xs=1+INT (ys?2)  
137 IF CODE q$(4,z)=2 THEN LET xs=1  
139 RETURN
```

se trata la posibilidad de hacer parpadear líneas de manera muy sencilla:

```

140 LET q$(5,z)="0": PRINT AT 21, NOT PI;
    "Intermitencia? S/N",,: PAUSE NOT PI: IF INKEY$
    ="S" THEN LET q$(5,z)="1"
142 RETURN

```

También la duración de la pausa de fija rápidamente:

```

150 PRINT AT 21, NOT PI; "'¿Cuántos segundos?'," : INPUT
    LINE n$: IF LEN n$=0 THEN GO TO 150

152 IF VAL n$<1 OR NAL n$>255 THEN GO TO 150
154 LET q$(6,z)=CHR$ VAL n$
156 RETURN

```

Los colores para el fondo, márgenes laterales y escritura los arreglamos así:

```

160 PRINT AT 21,NOT PI; "Color (0-7)",,: PAUSE NOT PI:
    LET f4=INKEY$: IF CODE f$<48 OR CODE fs>55
    THEN GO TO 160
162 IF n$="f" THEN LET q$(9,z)=f$
164 IF n$="u" THEN LET q$(7,z)=f$
166 IF n$="h" THEN LET q$(8,z)=f$
168 PRINT AT 21,NOT PI,, : RETURN

```

Ahora la pregunta por el cambio de color de los márgenes laterales;

```

170 LET q$(10,z)="0": PRINT AT 21, NOT PI;
    "Variaciones de color? S/N",,: PAUSE NOT PI: IF
    INKEY$="S" THEN LET q$(10,z)="1"
172 RETURN

```

Con esto hemos terminado esta parte. Ahora sólo faltan tres rutinas que ya hemos incluido en el programa, y son : a partir de línea 200 (escribir número de línea en la parte izquierda de la pantalla), a partir de línea 500 (visualizar una línea) y a partir de línea 800 el insertar una línea de texto.

```

200 FOR i=11 TO 30: LET f$=STR$ i: PRINT INK NOT PI;
    PAPER VAL "7";AT i-10,NOT PI;f$(2): NEXT i: RETURN

```

```

500 LET p$=t$: GO SUB 135: LET yy=CODE q$(3,z)*8:
    BORDER VAL q$(8,z):PAPER VAL q$(7,z): INK VAL
    q$(9,z)
510 FLASH VAL q$(5,z): GO SUB big: FLASH NOT PI:
    RETURN

```

y

```

800 FOR k=d-1 TO z STEP -1
810 IF a$(k)=I$ THEN GO TO 840
820 LET a$(k+1)=a$(k)
830 FOR i=VAL "1" TO 10: LET q$(i,k+1)=q$(i,k):NEXT i
840 NEXT k
850 LET a$(z)=I$
860 LET q$(2,z)=CHR$ 0: LET q$(3,z)=CHR 0: LET q$(4,z)
    ="1"
870 LET q$(5,z)="0": LET q$(6,z)=CHR$ 5: LET q$(7,z)
    ="6": LET q$(8,z)="6"
880 LET q$(9,z)="0": LET q$(10,z)="0"
890 GO TO te

```

Con esto podemos introducir textos y dejarlos transcurrir en la pantalla de la forma deseada. Pero tenemos que asegurar la posibilidad de corregir posibles faltas o de efectuar las necesarias modificaciones.

Esto estaba previsto a partir de la línea 5000

```

5000 CLS
5010 LET anf=VAL "1":LET end=d

```

Primero la pregunta, si únicamente se han de añadir nuevas líneas:

```

5020 PRINT''¿Añadir texto?S/N':PAUSE NOT PI:IF
    INKEY$="S" THEN LET flag=1 GO TO 1200
5030 CLS :PRINT''¿Todas las líneas? S?N':PAUSE
    NOT PI: IF INKEY$<>"N"THEN GO TO 5100
5050 INPUT ''¿A partir de qué línea?':LINE z$
5060 LET anf=VAL z$

```

Para estar seguro, puede introducir aquí una pregunta respecto a la exactitud de la introducción. Después se continúa con la corrección. Para ello utilizaremos el código a partir de la línea 2130.

```
5100 FOR anf TO end
5110 IF a$(z)=1$ THEN GO TO m
5120 GO SUB 2130
```

y muy importante:

```
2380 IF m$="2" THEN RETURN
```

Ahora seguimos con:

```
5130 PRINT AT NOT PI,NOT PI; 'Línea',z
5200 PRINT AT 21,NOT PI;"Cambiar S/N
      (0=Menu,W=Publicidad)": PAUSE NOT PI:LET f$=INKEY$
      : IF f$="S" THEN GO TO 5600
5210 IF f$="o" THEN GO TO m
5220 IF f$="w" THEN LET m$="3": PRINT AT 21,NOT PI,,GO
      TO 2130
5300 NEXT z
5310 GO TO m
```

Si se ha de hacer algún cambio, activaremos la rutina 'preguntar'.

```
5600 GO SUB preguntar
5610 IF n$<>"y" AND n$<>"e" THEN GO TO 5120
5620 IF CODE q$(2,z)=0 THEN GO TO 5900
5630 LET a$(z)=1$
5640 LET a$(z,1 TO LEN t$)=t$
5700 GO TO 5120
5900 GO SUB 400
5910 CLS : GO TO 5120
```

Lo único que falta es asegurar el programa y, con él, las líneas, si se han de utilizar más tarde. Para conseguir la autoejecución se debería introducir la siguiente línea:

```
8000 POKE 23730,255: POKE 23731,125: LOAD "*"m";1;"c"
CODE
```

si se trabaja con cassette. Los POKES son necesarios, para que CLEAR no borre todas las variables y, con ello, todas las líneas memorizadas.

La rutina de memorización puede tener la siguiente forma:

```
9900 CLS : PRINT "Asegurar el programa/FINAL"
9910 PRINT "' 1= memorizar en Microdrive'"
      2= memorizar en cassette'"'3 = FINAL sin
      memorizar'"'4 = Menu"
9920 PAUSE NOT PI: LET m$=INKEY$: IF m$="3" THEN
      CLS : STOP
9925 IF m$="4" THEN GO TO m
9930 IF M$="2" THEN GO TO 9980
9940 IF m$<>"1" THEN GO TO 9920
9945 ERASE "m"; 1;"awis"
9950 SAVE "*"m";1;"awis" LINE 8000
9955 REM SAVE*"m";1;"c" CODE 32256,340
9960 GO TO 9900
9980 CLS: SAVE "awis" LINE 8000
9985 CLS: PRINT "'¡Por favor pulsar una tecla!'"
9990 SAVE "c" CODE 32256,340
9995 GO TO m
```

En la línea 9955 hay que borrar el REM, si la parte del código máquina no esta aún memorizada en el Microdrive c cassette correspondiente.

Bien, ya hemos terminado y el asunto puede funcionar. Quizás quiera Vd., después de algun tiempo, introducir otras posibilidades en el programa. Por ejemplo, el campo a\$ podría fijarse a menos de 30 caracteres, desconectar el centrado automático y conseguir diferentes tamaños de escritura en una línea. También puede Vd. introducir la posibilidad de dejar transcurrir ciertas líneas en la pantalla. Esto desde luego, debería hacerse en código máquina para que no sea muy lento. No hay límites para su fantasía.

Pero en este momento, el programa es aceptable. Cuando Vd. lo carga, se inicia automáticamente y muestra el menú. Tenga Vd. en cuenta, que cuando el programa trabaja, es posible interrumpirlo a través de las teclas 'm' para el menú, 'p' para pausa, "z" para número de línea y "k" para correcciones.

Al final del capítulo vemos todo el listado completo del programa.

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****

9 GO SUB 80
10 LET big=3500: LET m=9000: GO TO m
20 LET d=25
30 DIM q$(10,d): DIM a$(d,32)
40 FOR k=1 TO d: LET q$(4,k)="1": LET q$(6,k)=CHR$ 5: LET
q$(7,k)="6": LET q$(8,k)="6": LET q$(9,k)="0"
50 LET q$(1,k)="0": LET q$(5,k)="0": LET q$(10,k)="0": NEX
T k
60 LET big=3500: LET m=9000: LET flag=NOT PI
70 LET v$=" ": LET l$=" "
80 LET nb=100: LET te=110: LET ze=120: LET gr=130: LET bl=
140
82 LET pa=150: LET fa=160: LET va=170
84 LET pregunta=300
90 RETURN
100 LET q$(1,z)="0": PRINT AT 21,NOT PI;"NUEVA PANTALLA ? (
s/n) (0=MENU)",: PAUSE NOT PI: LET f$=INKEY$: IF f$="s" THEN
LET q$(1,z)="1": CLS
102 IF z=VAL "1" THEN GO TO 106
104 IF f$="n" THEN LET q$(6,z-1)=CHR$ 1
106 IF f$<>"s" AND f$<>"n" THEN GO TO m
108 RETURN
110 LET q$(2,z)=CHR$ 0: PRINT AT 21,NOT PI;"Texto=",,: INPU
T LINE t$
112 IF LEN t$>30 THEN GO TO 110
114 LET q$(2,z)=CHR$ (LEN t$)
116 RETURN
120 GO SUB 200: PRINT AT 21,NOT PI;"Linea(1-20) =",,: INPUT
LINE n$
122 FOR s=VAL "1" TO 20: IF n$=STR$ s THEN GO TO 128
124 NEXT s
126 GO TO 120
128 LET q$(3,z)=CHR$ (VAL n$): RETURN
130 PRINT AT 21,NOT PI;"Altura escritura (1-9)",: LET f$=IN

```

150

```

KEY$: IF f$>"9" OR f$<"1" THEN GO TO 130
132 LET q$(4,z)=f$
135 LET ys=VAL q$(4,z): LET xs=1+INT (ys/2)
137 IF q$(4,z)="2" THEN LET xs=1
139 RETURN
140 LET q$(5,z)="0": PRINT AT 21,NOT PI;"Parpadeo ? (s/n)",
, : PAUSE NOT PI: IF INKEY$="s" THEN LET q$(5,z)="1"
142 RETURN
150 PRINT AT 21,0;"Durante cuantos segundos ?": INPUT LIN
E n$: IF LEN n$=0 THEN GO TO 150
152 IF VAL n$>250 OR VAL n$<1 THEN GO TO 150
154 LET q$(6,z)=CHR$ VAL n$
156 RETURN
160 PRINT AT 21,NOT PI;"Color ? (0-7)",,: PAUSE NOT PI: LET
f$=INKEY$: IF CODE f$<48 OR CODE f$>55 THEN GO TO 160
162 IF n$="f" THEN LET q$(9,z)=f$
164 IF n$="u" THEN LET q$(7,z)=f$
166 IF n$="h" THEN LET q$(8,z)=f$
168 PRINT AT 21,NOT PI,, : RETURN
170 LET q$(10,z)="0": PRINT AT 21,NOT PI;"Variaciones de co
lor ? (s/n)", : PAUSE NOT PI: IF INKEY$="s" THEN LET q$(10,z
)="1"
172 RETURN
200 FOR i=11 TO 30: LET f$=STR$ i: PRINT AT i-10,NOT PI; PA
PER VAL "7"; INK NOT PI;f$(2): NEXT i: RETURN
300 PRINT AT 21,NOT PI; PAPER VAL "7"; INK NOT PI;"T U H F
Z B G P N E V Ok "
305 LET v$=" "
310 PAUSE NOT PI: LET n$=INKEY$
311 PRINT AT 21,0,,
312 IF n$="u" OR n$="h" OR n$="f" THEN GO TO fa
313 IF n$="z" THEN GO TO ze
314 IF n$="b" THEN GO TO bl
315 IF n$="p" THEN GO TO pa
316 IF n$="n" THEN GO TO nb
317 IF n$="e" THEN GO TO 800
318 IF n$="v" THEN GO TO va
319 IF n$="o" THEN RETURN
320 IF n$="t" THEN GO TO te
321 IF n$="g" THEN GO TO gr
330 GO TO pregunta
400 FOR k=z TO d-1

```

```

402 LET a$(k)=a$(k+1)
406 FOR r=VAL "1" TO 10: LET q$(r,k)=q$(r,k+1): NEXT r
420 NEXT k
425 LET a$(d)=1$
430 RETURN
500 LET p$=t$: GO SUB 135: LET yy=CODE q$(3,z)*8: BORDER VA
L q$(7,z): PAPER VAL q$(8,z): INK VAL q$(9,z)
510 FLASH VAL q$(5,z): GO SUB big: FLASH NOT P1: RETURN
800 FOR k=d-1 TO z STEP -1
801 IF a$(k)=1$ THEN NEXT k
810 IF a$(k)=1$ THEN GO TO 840
820 LET a$(k+1)=a$(k)
830 FOR i=VAL "1" TO 10: LET q$(i,k+1)=q$(i,k): NEXT i
840 NEXT k
850 LET a$(z)=1$
860 LET q$(2,z)=CHR$ 0: LET q$(3,z)=CHR$ 0: LET q$(4,z)="1"
870 LET q$(5,z)="0": LET q$(6,z)=CHR$ 5: LET q$(7,z)="6": L
ET q$(8,z)="6"
880 LET q$(9,z)="0": LET q$(10,z)="0"
890 GO TO te
1000 CLS : PRINT "Borrar todas las lineas ? (s/
n)": PAUSE NOT P1: IF INKEY$="s" THEN GO TO 1100
1010 CLS : LET flag=1: GO TO 1200
1100 CLEAR : GO SUB 20
1200 FOR z=VAL "1" TO d
1210 IF flag=VAL "1" AND a$(z)<>1$ THEN GO TO 1800
1220 GO SUB nb
1230 GO SUB 200: GO SUB te
1240 IF CODE q$(2,z)=0 THEN GO TO m
1250 LET a$(z,1 TO LEN t$)=t$
1260 GO SUB ze: GO SUB gr
1280 GO SUB 500
1290 GO SUB pregunta
1310 IF n$="t" THEN LET a$(z,1 TO LEN t$)=t$
1320 IF n$="o" THEN GO TO 1800
1330 GO TO 1280
1800 NEXT z
1900 GO TO m
2000 CLS : IF a$(1)=1$ THEN GO TO m
2010 LET anf=VAL "1": LET end=d
2020 PRINT "Todas las lineas ? (s/n)": PAUSE NOT P1: LET
z$=INKEY$: CLS

```

```

2030 IF z$(">n") THEN GO TO 2100
2040 INPUT "Primera linea = ?"; LINE z$: IF z$="" THEN LET
z$="1"
2050 LET anf=VAL z$: IF anf<d THEN GO TO 2040
2060 INPUT "Ultima linea = ?"; LINE z$: IF z$="" THEN LET z
$=STR$ d
2070 LET end=VAL z$: IF end>d THEN GO TO 2060
2080 IF anf>end THEN GO TO 2000
2100 FOR z=anf TO end
2110 IF INKEY$="m" THEN GO TO m
2120 IF a$(z)=1$ THEN GO TO 2100
2130 BORDER VAL q$(7,z): PAPER VAL q$(8,z): INK VAL q$(9,z)
2200 LET yy=8*CODE q$(3,z)
2210 GO SUB 135
2220 LET p$=a$(z,1 TO CODE q$(2,z))
2230 IF q$(1,z)="1" THEN CLS
2240 FLASH VAL q$(5,z)
2300 GO SUB big
2310 FLASH NOT PI
2320 IF q$(10,z)="1" THEN FOR k=0 TO 7: BORDER k: PAUSE 50:
NEXT k
2330 IF m$="2" THEN RETURN
2340 IF CODE q$(6,z)=NOT PI THEN LET q$(6,z)=CHR$ 1
2390 IF CODE q$(6,z)=0 THEN LET q$(6,z)=CHR$ 1
2400 PAUSE 50*CODE q$(6,z)
2410 IF INKEY$="p" THEN PRINT AT 0,0;"Pausa": PAUSE 0: PRIN
T AT 0,0;" ": GO TO 2410
2420 IF INKEY$="z" THEN PRINT AT 0,0;"Linea ";z: PAUSE 0: P
RINT AT 0,0;" ": GO TO 2410
2430 IF INKEY$="k" THEN GO TO 5200
2500 NEXT z
2600 GO TO 2100
3500 LET xx=(256-8*xs*LEN p$)/2
3509 IF yy=256 THEN LET yy=255
3510 LET i=23306: POKE i,xx: POKE i+1,yy: POKE i+2,xs: POKE
i+3,ys: POKE i+4,8: LET i=i+4: LET w=LEN p$: FOR n=1 TO w: P
OKE i+n,CODE p$(n): NEXT n: POKE i+w+1,255: LET w=USR 32256:
RETURN
5000 CLS
5010 LET anf=VAL "1": LET end=d
5020 PRINT "Adjuntar texto ? (s/n)": PAUSE 0: IF INKEY$="s
" THEN LET flag=1: GO TO 1200

```

```

5030 CLS : PRINT ""Todas las lineas ? (s/n)": PAUSE NOT PI:
    IF INKEY$<>"n" THEN CLS : GO TO 5100
5050 INPUT "Desde que linea ? "; LINE z$
5060 LET anf=VAL z$
5100 FOR z=anf TO end
5110 IF a$(z)=1$ THEN GO TO m
5120 GO SUB 2130
5130 PRINT AT NOT PI,NOT PI;"Linea ";z
5200 PRINT AT 21,NOT PI;"Cambiar? (s/n)<0=Menu W=Anuncio>":
    PAUSE NOT PI: LET f$=INKEY$: IF f$="s" THEN GO TO 5600
5210 IF f$="o" THEN GO TO m
5220 IF f$="w" THEN LET m$="3": PRINT AT 21,0,, : GO TO 2130
5300 NEXT z
5310 GO TO m
5600 GO SUB pregunta
5610 IF n$<>"t" AND n$<>"e" THEN GO TO 5120
5620 IF CODE q$(2,z)=0 THEN GO TO 5900
5630 LET a$(z)=1$
5640 LET a$(z,1 TO LEN t$)=t$
5710 GO TO 5120
5900 GO SUB 400
5910 CLS : GO TO 5120
8000 POKE 23730,255: POKE 23731,125: LOAD "c"CODE 32256: REM
    : LOAD $"m";1;"c"CODE
8015 LET flag=NOT PI
9000 INK 0: BORDER 6: PAPER 6: CLS
9010 LET p$="Menu AWIS": LET yy=10: LET xs=2: LET ys=4
9020 GO SUB big
9030 PRINT AT 9,3;"1 = Componer nuevo anuncio"
9040 PRINT ""      2 = Comprobar y/o modificar"
9050 PRINT ""      3 = Transcurrir publicidad"
9060 PRINT ""      4 = F I N"
9070 PRINT AT 21,2;"Introduzca la cifra deseada"
9100 PAUSE 0: LET m$=INKEY$
9110 IF m$="1" THEN GO TO 1000
9120 IF m$="2" THEN GO TO 5000
9130 IF m$="3" THEN GO TO 2000
9140 IF m$="4" THEN GO TO 9900
9150 GO TO 9100
9900 CLS : PRINT "Almacena programa/FIN"
9910 PRINT ""1 = almacenar en el microdrive""2= almacenar
    en cassette""3 = FIN sin almacenamiento""4 = Menu"
9920 PAUSE NOT PI: LET m$=INKEY$: IF m$="3" THEN CLS : STOP

9925 IF m$="4" THEN GO TO m
9930 IF m$="2" THEN GO TO 9980
9940 IF m$<>"1" THEN GO TO 9920
9945 REM ERASE "m";1;"awis"
9950 REM SAVE $"m";1;"awis" LINE 8000
9955 REM SAVE $"m";1;"c"CODE 32256,340
9960 GO TO 9900
9980 CLS : SAVE "AWIS" LINE 8000
9985 CLS : PRINT "Por favor pulse una tecla! "
9990 SAVE "c"CODE 32256,340
9995 GO TO m

```

CAPITULO 10: EL SPECTRUM EN LA PEQUEÑA EMPRESA

Aspectos generales

¿El ZX Spectrum en aplicaciones profesionales? Quizás esto suene un poco presuntuoso. Pero si observamos la situación de modo realista, nos daremos cuenta de que el Spectrum puede realizar una serie razonable de trabajos y, por consiguiente, puede significar un ahorro de tiempo para 'pequeños empresarios'.

Por encima de esto, ofrece la posibilidad económica de introducción en la elaboración de datos. Si se trata de un Spectrum Plus, o de un Spectrum con teclado adicional y dado que existen Microdrives e impresoras para su disposición, este pequeño empresario no ahorrará nada más, si posee los programas adecuados.

Los trabajos más frecuentes caen en el campo del procesado de textos, ficheros de clientes y administración de datos, o sea administración de almacén y volumen de ventas. Para el procesado de textos hemos mencionado en el capítulo 4 el programa Tasword 2. Con este programa se pueden redactar perfectamente cartas y pequeños documentos. También pueden escribirse facturas normalizadas, creando un formulario de factura, y memorizando luego el mismo. Para cada factura se carga y se 'rellena'.

A continuación queremos desarrollar un programa para fichero de clientes, así como una administración de almacén y volumen de ventas. Juntamente con el procesador de texto Tasword 2 y el programa gráfico descrito en el capítulo 8, se dispone de un paquete amplio de programas que pueden traer muchos beneficios.

Quien quiera utilizar su Spectrum de forma auténticamente profesional, se proporcionará los Microdrives como memoria, para evitar tiempos de espera al cargar y almacenar programas. Por ello, vamos a comenzar con un pequeño programa de mando, que simplifica mucho el trabajo rutinario, ya que los programas pueden ser seleccionados desde un menú. A parte de esto, este programa también se puede utilizar en plan particular.

El programa de mando

Si se conecta el Spectrum o ha introducido NEW y despues RUN y ENTER, el Spectrum comprueba si hay una cassette en el Microdrive 1. En este caso, el directorio se chequea y se busca un programa "run". Si este programa existe, se carga automáticamente. O sea, vamos a nombrar a nuestro programa de mando 'run', y vamos a asegurarlo con la opción 'LINE', de forma que el inicio automático esté garantizado.

El programa es muy sencillo y no necesita de ninguna aclaración.

```
100 CLEAR 65535: PAPER 0: INK 6: BORDER 0:CLS
110 PRINT INK 3: TAB 7; "MENU PRINCIPAL"
120 PRINT "*****"
130 PRINT "'' "1 = Administracion de almacen y
    volumen de ventas"
140 PRINT "'' "2 = Fichero de clientes"
150 PRINT "'' "3 = Representaciones graficas"
160 PRINT "'' "4 = Procesado de texto"
170 PRINT "'' "5 = FINAL"
200 PRINT AT 21,0,,AT 21,4; INK 4; "Por favor,
    introducir un numero"
300 PAUSE 0: LET m$=INKEY$
310 IF m$= "5" THEN CLS : PRINT AT 9,11; "FINAL"
    :STOP
320 IF m$= "1" THEN GO SUB 600: LOAD* "m";1;"Busyman"
330 IF m$= "2" THEN GO SUB 600: LOAD* "m";1;"Busydat"
340 IF m$= "3" THEN GO SUB 600: LOAD* "m";1;
    "Busygraph"
350 IF m$= "4" THEN GO SUB 600: LOAD* "m";1;"Tasword"
400 PRINT AT 21,0; FLASH 1;
    "***** Falsa introducción*****"
410 FOR I=1 TO 10: BEEP .05,I: NEXT I: GO TO 200
600 CLS: PRINT AT 2,9; "Por favor, esperar": RETURN
```

Este programa se almacena después de su introducción, con el comando directo:

SAVE *"m";1;"run" LINE 1

Ahora vamos a comenzar con el fichero de clientes y la administración de almacén y volumen de ventas. En estos dos programas vamos a proceder de la siguiente manera: Primero se dará una descripción de los listados, o sea una explicación de las variables y campos, así como la explicación de cada parte del programa. Después se adjunta el listado completo del programa, al cual se le añade un 'manual de manejo' del mismo.

Si Vd. quiere realizar algunos cambios en el programa, no será ningún problema orientarse en los listados, después de las aclaraciones dadas.

El fichero de clientes Busydat

Comenzemos con el fichero de clientes. Ya que no queremos construir un fichero convencional, es decir, que el programa no sólo memorice el nombre y dirección de un cliente, sino que también memorice un determinado comentario respecto a cada nombre, estos comentarios se dividirán en 5 categorías, que Vd. puede elegir libremente. Estas categorías pueden ser el volumen de compras, los principales intereses, o si así se desea, el color del cabello. Naturalmente debe ser posible seleccionar ciertos clientes del fichero que cumplan determinados criterios.

Otra pequeña peculiaridad del fichero: los clientes a fichar de una pequeña empresa, generalmente no están distribuidos por toda España o Europa. Lo más corriente es que no se memoricen más de 5 o 10 localidades. A causa de esto, se va a prever, para cada cliente, un código para su localidad, como máximo 10 localidades, que Vd. puede elegir según su criterio. Esto le ahorrará mucho espacio de memoria, pero en la mayoría de los casos será suficiente. Quien necesite más localidades, puede memorizar por ejemplo dos versiones del programa.

La estructura del programa se ha construido de tal forma, que todos los nombres y datos son cargados juntamente con el programa. Esto permite tiempos de acceso relativamente cortos, pero también tiene la ventaja de que quien no disponga de microdrives, pueda trabajar igualmente con el programa. Únicamente tiene la desventaja de no poder utilizar el programa de mando, y contar con tiempos de espera más largos.

Ahora al programa propiamente dicho. Contiene un menú que comienza en la línea 9000.

Aquí se dan las posibilidades de introducir datos, leer, corregir y borrar. Los siguientes puntos son la entrada de los datos básicos, o sea de la categoría de los comentarios, los propios comentarios y las localidades con número directivo postal.

La primera línea del programa fija los colores para el primer plano y el fondo. Estos se pueden cambiar en cualquier momento según sus deseos, pero entonces debería variar todos los demás comandos correspondientes en el programa.

A partir de la línea 20 se reservan los espacios de memoria para los diferentes campos. Las líneas 200 hasta 999 contienen diferentes subrutinas, que son aplicadas en determinados sectores del programa. A partir de la línea 1000 se maneja la introducción de un cliente. En ello se distingue entre Señor, Señora y Empresa. Respecto a cada cliente se puede memorizar el nombre, apellidos, el código para la localidad donde reside, así como un número de cliente. A continuación puede ser clasificado en cinco categorías, de las cuales 4 están previstas como cadena y la quinta como número. Este es el espacio en el cual se puede introducir, por ejemplo, el número de cuenta bancaria del cliente.

La lectura del fichero de clientes comienza a partir de la línea 2000 con un menú. Aquí Vd. puede decidirse si quiere leer todos los clientes, o únicamente una selección de ellos. Vd. puede buscar un nombre determinado, un número de cliente, clientes con determinados comentarios o a partir de un número sucesivo. También al leer dispone Vd. de la posibilidad de corregir cualquier entrada. Si así lo desea, el programa salta a las líneas superiores a 3000, en las cuales se efectúan las modificaciones propiamente dichas.

A partir de la línea 4000 se borra una entrada. Esto se realiza fácilmente, todas las demás entradas se corren un puesto hacia abajo.

k\$(a,4) Código para los comentarios de las primeras
cuatro categorías

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****

9 BORDER 0: PAPER 0: INK 6
10 LET m=9000: GO TO m
20 LET a=100
30 DIM q$(a): DIM n$(a,18)
40 DIM v$(a,22)
50 DIM s$(a,22)
60 DIM p$(10,22)
65 FOR i=1 TO 10: LET p$(i,5 TO 9)="libre": NEXT i
70 DIM k(a)
80 DIM t$(5,10): DIM k$(a,4): DIM z(a)
85 FOR l=1 TO 5: LET t$(l)=" libre": NEXT l
90 DIM c$(4,4,10)
100 GO TO m
200 LET f$="0": CLS : PRINT TAB 5;"S E L E C C I O N"
205 PRINT "1 = Buscar nombre""2 = Buscar numero de client
e""3 = Buscar comentario""4 = A partir de un numero dado
""5 = Menu"
210 PAUSE 0
215 IF INKEY$="5" THEN GO TO m
220 IF INKEY$="1" THEN GO TO 2130
225 IF INKEY$="2" THEN GO TO 330
230 IF INKEY$="4" THEN GO TO 300
235 IF INKEY$="3" THEN GO TO 8000
240 GO TO 210
260 CLS : LET g$="Sra. "
264 IF n$(i,1)="h" THEN LET g$="Sr. "
265 IF n$(i,1)="e" THEN LET g$="Empresa "
269 PRINT AT 0,5: INVERSE 1;i;AT 4,0: INVERSE 0;g$
270 LET k=i: GO SUB 500: PRINT AT 5,0;v$(i,1 TO 1);" ";
275 LET k=i: GO SUB 510: PRINT INVERSE 1;n$(1,2 TO 1)
280 PRINT AT 7,0;s$(i)
290 PRINT p$(VAL q$(i))
292 PRINT AT 1,0;"Numero de cliente ";k(i);AT 10,0;"Comenta
rio:"

```

```

293 GO SUB 560
295 RETURN
300 INPUT "Desde num. ? (1 - ";(a);")",m$
305 FOR l=1 TO LEN m$: IF CODE m$(l)<48 OR CODE m$(l)>57 TH
EN GO TO 300
310 NEXT l
315 IF VAL m$>a THEN GO TO 300
320 LET anf=VAL m$
325 GO TO 2220
330 INPUT "Que numero? ";m$
335 FOR i=1 TO a: IF STR$ k(i)=m$ THEN GO TO 350
340 NEXT i
345 CLS : PRINT AT 10,8; FLASH 1;"No existe": PAUSE 150: GO
TO 9000
350 GO SUB 260
360 GO TO 700
500 FOR l=LEN v$(k) TO 1 STEP -1: IF v$(k,l)<>" " THEN GO
TO 505
504 NEXT l
505 RETURN
510 FOR l=LEN n$(k) TO 1 STEP -1: IF n$(k,l)<>" " THEN GO
TO 515
514 NEXT l
515 RETURN
550 FOR l=1 TO 12: PRINT AT 9+l,0,, : NEXT l: RETURN
560 FOR l=1 TO 4: PRINT AT 10+l,0;1;TAB 3;t$(l);: IF k$(i,l
)>>" " THEN PRINT TAB 20;c$(1,VAL k$(i,l))
562 NEXT l
565 PRINT AT 15,0;"5";TAB 3;t$(5);TAB 20;z(i)
570 RETURN
600 FOR l=1 TO 10: PRINT AT 1+10,0;1;TAB 3;p$(l): NEXT l
602 INPUT "Codigo para poblacion ";m$
604 LET q$(k)=m$: GO SUB 550
606 PRINT AT 8,0;p$(VAL q$(k))
610 RETURN
620 REM correccion comentario
630 INPUT "Num. de la categoria? (1-5)";m$: IF CODE m$(1)>5
3 OR CODE m$(1)<49 THEN GO TO 632
635 FOR j=16 TO 21: PRINT AT j,0,, : NEXT j
636 IF m$="5" THEN INPUT "Numero = ";z(k): GO TO 3300
640 FOR j=1 TO 4: PRINT AT 17,0+j,0;j;TAB 3;c$(VAL m$,j): N
EXT j

```

```

650 INPUT "Comentario de codigo (0=ninguno)"; LINE x$: IF x
$="0" THEN LET k$(k,VAL m$)=" ": GO TO 3300
660 IF x$<"1" OR x$>"4" THEN GO TO 650
670 LET k$(k,VAL m$)=x$
690 GO TO 3300
700 PRINT AT 21,0; INK 4; "Menu.....Correccion.....Borrar"
710 PAUSE 0
715 IF INKEY$="b" THEN GO TO 4100
720 IF INKEY$="c" THEN GO TO 3100
730 IF INKEY$="m" THEN GO TO m
740 PRINT AT 21,0,,
745 PRINT AT 21,0;
750 GO TO 700
800 REM nueva localidad
810 FOR i=1 TO 10
820 IF p$(i,1)=" " THEN GO TO 840
825 NEXT i
830 CLS : PRINT FLASH 1; "No acepto ninguna localidad mas":
PAUSE 150
839 GO TO 600
840 LET c$(k)=STR$ i
845 LET p$(i)=m$
850 PRINT AT 8,0;p$(i)
855 FOR i=1 TO 11: PRINT AT 10+i,0,, : NEXT i
860 GO TO 610
1000 CLS : REM *****ENTRADA*****
1020 FOR k=1 TO a
1030 IF n$(k,1)=" " THEN GO TO 1100
1040 NEXT k
1050 PRINT AT 10,8; FLASH 1; "FICHERO COMPLETO": PAUSE 300
1070 GO TO 9000
1100 INPUT "Sr./Sra./Empresa H/S/E"; m$
1101 IF m$="h" THEN LET n$(k,1)="h": GO TO 1106
1102 IF m$="s" THEN LET n$(k,1)="s": GO TO 1106
1103 IF m$="e" THEN LET n$(k,1)="e": GO TO 1106
1104 GO TO 1100
1106 LET g$="Sra.": IF n$(k,1)="h" THEN LET g$="Sr."
1107 IF n$(k,1)="e" THEN LET g$="Empresa"
1108 PRINT AT 5,0;g$;
1109 INPUT "Nombre propio = ";v$(k): GO SUB 500
1110 PRINT v$(k,1 TO 1);" "
1112 INPUT "Apellido = ";n$(k,2 TO )

```

```

1113 IF CODE n$(k,2)>95 THEN LET n$(k,2)=CHR$ (CODE n$(k,2)
-32)
1115 PRINT n$(k,2 TO )
1140 INPUT "c/   = ";s$(k): PRINT AT 7,0;s$(k)
1150 GO SUB 600
1160 INPUT "Numero de cliente = ";k(k)
1165 PRINT AT 0,0;"Numero de cliente = ";k(k)
1170 PRINT AT 12,0;"Comentario:": FOR i=1 TO 4: PRINT i;TAB
3;t$(i): NEXT i
1175 FOR i=1 TO 4: IF k$(k,i)>"0" THEN PRINT AT 12+i,20;c$(
i,VAL k$(k,i))
1176 NEXT i
1179 PRINT AT 17,0;"5   ";t$(5);TAB 20;z(k)
1180 INPUT "num. de la categoria (0=fin)"; LINE m$: IF m$="0
" THEN GO TO 1300
1190 IF m$<"1" OR m$>"5" THEN GO TO 1180
1200 IF m$="5" THEN INPUT "Cifra ";z(k): PRINT AT 17,20;z(k
): GO TO 1180
1210 FOR i=1 TO 4: PRINT AT 17+i,3; PAPER 7; INK 0;i;" ";c$(
VAL m$,i): NEXT i
1220 INPUT "Codigo = (0=ninguno) ";x$: IF x$="0" THEN GO TO
1170
1230 IF x$<"1" OR x$>"4" THEN GO TO 1220
1240 LET k$(k,VAL m$)=x$
1245 FOR i=18 TO 21: PRINT AT i,0;"                               ": NEXT i
1250 GO TO 1170
1300 INPUT "Otra inscripcion? (s/n) "; LINE m$
1302 IF m$<>"s" THEN GO TO 9000
1305 CLS
1310 LET k=k+1
1320 IF k>a THEN GO TO 1050
1330 IF n$(k,1)<>" " THEN GO TO 1310
1400 CLS
1410 GO TO 1100
2000 CLS : REM *****LECTURA*****
2010 LET k=1
2100 PRINT AT 10,8;"Todos los nombres? (s/n) ": PAUSE 0
2120 IF INKEY$<>"n" THEN GO TO 2200
2125 GO TO 200
2130 CLS : INPUT "Nombre buscado = ";r$
2132 LET anf=1
2135 FOR i=anf TO a

```

```

2140 IF r$=n$(i,2 TO 1+LEN r$) THEN GO TO 2160
2145 NEXT i
2150 CLS : PRINT AT 10,5; FLASH 1;" El nombre no existe ": P
AUSE 200: GO TO m
2160 GO SUB 260
2165 LET anf=i+1
2170 PRINT AT 21,0;"Seguir buscando? (s/n) ": PAUSE 0: PRINT
AT 21,0,,
2172 IF INKEY$="s" THEN GO TO 2135
2175 IF m$="4" THEN GO TO 4100
2176 IF m$="3" THEN GO TO 3100
2180 PRINT AT 21,0; INK 4;"Menu.....Correccion.....Borrar"
2183 PAUSE 0
2185 IF INKEY$="b" THEN GO TO 4100
2186 IF INKEY$="c" THEN GO TO 3100
2188 IF INKEY$="m" THEN GO TO m
2190 PRINT AT 21,0,,
2195 GO TO 2400
2210 LET anf=1
2220 FOR i=anf TO a
2250 CLS : IF n$(i,1)=" " THEN GO TO m
2260 GO SUB 260: PAUSE 99
2360 IF INKEY$<>" " THEN PRINT AT 20,0;"ENTER= seguir o bie
n...": GO TO 2180
2400 NEXT i
2999 GO TO 9000
3000 CLS : REM ***CORRECCION***
3010 GO TO 2130
3100 PRINT AT 15,0; INK 2;"Que se ha de cambiar ? ": PRINT
INK 5;"A = apellido","N = nombre","C = calle, numero","P =
poblacion + codigo postal","S = Sr./Sra.",
3110 PRINT INK 5;"K = num cliente","R = comentario","M = me
nu"
3120 PRINT INK 2;"D = Todo en orden"
3130 PAPER 0: PAUSE 0
3140 IF INKEY$="m" THEN GO TO m
3150 IF INKEY$="a" THEN INPUT "Nuevo apellido = ";n$(i,2 TO
)
3160 IF INKEY$="n" THEN INPUT "Nuevo nombre = ";v$(i)
3170 IF INKEY$="n" THEN INPUT "Nuevo nombre = ";v$(i)
3180 IF INKEY$="c" THEN INPUT "Nueva calle (y num.) = ";s$(
i)
3190 IF INKEY$<>"s" THEN GO TO 3230
3200 INPUT "Sr./Sra./Empresa H/S/E"; LINE x$
3210 IF x$<>"h" AND x$<>"s" AND x$<>"e" THEN GO TO 3200
3220 LET n$(i,1)=x$
3230 IF INKEY$="o" THEN LET anf=i: GO TO 2200
3240 IF INKEY$="k" THEN INPUT "Nuevo numero de cliente = ";
k(i)
3250 IF INKEY$="r" THEN GO TO 620
3260 IF INKEY$="p" THEN GO SUB 550: GO SUB 600
3300 GO SUB 260: GO TO 3100

```

```

4000 CLS : REM ****BORRAR****
4010 GO TO 2130
4100 PRINT AT 20,0,,AT 21,0;"Borrar ?(s/n)",,,: PAUSE 0
4110 IF INKEY$="s" THEN GO TO 4200
4120 GO TO 2260
4210 LET n$(k-1)=n$(k)
4220 LET s$(k-1)=s$(k)
4230 LET k(k-1)=k(k)
4240 LET k$(k-1)=k$(k)
4250 LET v$(k-1)=v$(k)
4260 LET q$(k-1)=q$(k)
4270 IF n$(k,1)=" " THEN GO TO 4300
4280 NEXT k
4300 LET n$(a,1)=" "
4310 LET anf=1: GO TO 2260
5000 CLS : PRINT "Composicion de los datos basicos"
5002 PRINT ""1 $Localidades con numero postal"
5004 PRINT ""2 $Categorias para comentarios"
5006 PRINT ""3 $Comentarios"
5008 PRINT ""4 $Menu"
5010 PAUSE 0: LET m$=INKEY$: IF m$="4" THEN GO TO m
5020 IF m$<>"1" THEN GO TO 5040
5021 CLS : PRINT "Localidades y numero postal"
5022 FOR i=1 TO 10: PRINT AT 2*i,0;i;AT 2*i,3;p$(i): NEXT i
5025 INPUT "Localidad + num. postal (0=fin)"; LINE x$: IF x$
="0" THEN GO TO 5000
5026 PRINT AT 21,0; INVERSE 1;x$
5027 INPUT "Num. correspond. ?(0=ninguno)"; LINE m$: IF m$="
10" THEN GO TO 5030
5028 IF m$<"0" OR m$>"9" THEN GO TO 5027
5029 IF m$="0" THEN GO TO 5000
5030 LET p$(VAL m$)=x$
5039 GO TO 5021
5040 IF m$<>"2" THEN GO TO 5060
5041 CLS : PRINT "Categoria de los comentarios"
5042 FOR i=1 TO 5: PRINT AT 2*i,0;i;AT 2*i,3;t$(i): NEXT i
5045 INPUT "Nombre categoria (0=fin)"; LINE x$: IF x$="0" TH
EN GO TO 5000
5046 PRINT AT 21,0,,AT 21,0; INVERSE 1;x$
5047 INPUT "Num. correspond. ?(0=ninguno)"; m$: IF m$="0" THE
N GO TO 5041
5048 IF m$<"1" OR m$>"5" THEN GO TO 5047
5050 LET t$(VAL m$)=x$
5059 GO TO 5041
5060 IF m$<>"3" THEN GO TO 5000
5061 CLS : PRINT "Introducir comentarios": FOR i=1 TO 5: PRI
NT AT i,0;i;AT i,3;t$(i): NEXT i
5065 INPUT "Que categoria ? (0=fin)"; LINE m$: IF m$="0" THE
N GO TO 5000
5066 IF m$<"1" OR m$>"5" THEN GO TO 5065
5067 PRINT AT VAL m$,3; FLASH 1;t$(VAL m$)
5068 IF m$="5" THEN PRINT "" Aqui se introduce siempre
un numero al introducir un nombre, como por ejemplo, el volu
men de ventas, etc...": PAUSE 500: GO TO 5061
5070 FOR i=1 TO 4: PRINT AT 10+i,0;i;AT 10+i,3;c$(VAL m$,i):
NEXT i
5072 INPUT "Comentario = (0=FIN) "; LINE x$: IF x$="0" THEN
GO TO 5061

```

```

5074 PRINT AT 21,0;x$
5076 INPUT "Que numero ? (0=ninguno) ";o$: IF o$="0" THEN P
RINT AT 21,0,,,: GO TO 5072
5078 IF o$<"1" OR o$>"4" THEN GO TO 5076
5080 LET c$(VAL m$,VAL o$)=x$
5090 PRINT AT 21,0,,,: GO TO 5070
6000 CLS : REM *****FIN*****
6010 PRINT AT 10,0;"Almacenar datos ? (s/n)";: PAUSE 0
6020 IF INKEY$="n" THEN GO TO 6200
6030 CLS : PRINT " Almacenar datos""1 = Microdrive""2
= Casette""3 = M E N U""4 = F I N": LET m$=INKEY$
6040 IF m$="1" THEN GO TO 9996
6050 IF m$="2" THEN GO TO 6100
6055 IF m$="3" THEN GO TO m
6060 IF m$="4" THEN GO TO 6200
6070 GO TO 6030
6100 CLS : INPUT "Nombre para el fichero = ";x$: IF x$="" TH
EN GO TO 6100
6110 SAVE x$ LINE 1
6120 PRINT AT 15,0;"Rebobinar cinta para VERIFY ","despues p
ulse PLAY !": VERIFY x$: GO TO 6030
6200 REM LOAD $"m";1;"run"
8000 CLS : PRINT "Categorias": REM buscar
8010 INK 0: PAPER 5: FOR i=1 TO 5: PRINT AT 5*i-4,0;i;TAB 3;
t$(i): NEXT i: PAPER 4
8020 FOR i=1 TO 4
8030 FOR l=1 TO 4
8040 PRINT AT 5*i-5+1,18;l;" ";c$(i,l)
8050 NEXT l
8060 NEXT i
8065 PAPER 0: INK 6
8070 INPUT "Categoria (0=FIN)";x$: IF x$="0" THEN GO TO m
8080 IF x$<"1" OR x$>"5" THEN GO TO 8070
8100 INPUT "Comentario = (0=sin com.)"; LINE o$: IF o$="0" TH
EN GO TO 8070
8110 IF o$<"1" OR o$>"4" THEN GO TO m
8120 PRINT AT 5*VAL x$-5+VAL o$,20; FLASH 1;c$(VAL x$,VAL o$
)
8130 PAUSE 100
8300 FOR i=1 TO a
8310 IF n$(i,1)=" " THEN GO TO m
8320 IF k$(i,VAL x$)=o$ THEN GO SUB 260: PAUSE 99: IF INKEY
$<>" " THEN PAUSE 0
8330 NEXT i
9000 CLS : PRINT INK 3;TAB 4;"FICHERO DE DIRECCIONES","**
*****",TAB 12; INVERSE 1;" ? pcb "
9010 LET y=4: PRINT AT 5,y;"1 = Entrada de datos"
9020 PRINT AT 7,y;"2 = Leer datos"
9030 PRINT AT 9,y;"3 = Corregir datos"
9040 PRINT AT 11,y;"4 = Borrar datos"
9050 PRINT AT 13,y;"5 = Entrada de datos basicos"
9060 PRINT AT 15,y;"6 = F I N"
9070 PRINT AT 21,0;"Por favor, introduzca una cifra"
9100 PAUSE 0: LET m$=INKEY$
9110 IF CODE m$<49 OR CODE m$>54 THEN GO TO 9100
9120 GO TO 1000*VAL m$
9996 REM ERASE "m";1;"Busydat"
9997 REM SAVE $"m";1;"Busydat" LINE 1

```

Aplicación de Busydat

Cuando Vd. haya introducido y memorizado este programa, se inicia la primera vez con GO TO 20. Después siempre con GO TO 1 y, al memorizar, debería serlo con LINE 1. Si se introduce RUN, se pierden todos los datos memorizados.

Después de GO TO 20 se visualiza el menú sobre la pantalla. Lo mejor es introducir las localidades con el número directivo postal, los nombres de las categorías y los comentarios. Si no tiene todos los nombres de las localidades se pueden ampliar o modificar después.

En la categoría 5 no se puede introducir ningún comentario, ya que aquí lo que se pide es un número. Pero esto se lo indica el programa, en caso de que Vd. intente introducir otra cosa. Cuando Vd. haya arreglado estos puntos, puede volver al menú principal y comenzar a introducir nombres.

Esto no debería ofrecer ninguna dificultad, ya que al efectuar la introducción se indican las posibilidades. Cuando Vd. quiera leer datos, puede leer todos o sólo una selección de los mismos. También en estos casos, el programa le conduce de tal manera que no debería haber problemas. Cuando se están leyendo los nombres, se visualiza automáticamente cada lectura en la pantalla, antes de pasar a la siguiente. Si Vd. quiere ver una lectura durante un tiempo más largo o corregirla, pulsa la tecla ENTER, y el transcurso es interrumpido. Adicionalmente surge en la parte inferior de la pantalla una línea de comando, que le ofrece las opciones permitidas. De esta forma puede efectuar en cualquier momento correcciones, borrar datos, o proseguir la lectura, así como volver al menú principal.

Cuando haya finalizado su labor, el programa le pregunta si se han de asegurar los datos. En este punto Vd. puede realizar copias de seguridad. Después de esto y sin ser ya necesario el asegurar los datos, es cargado automáticamente el programa menú que hemos explicado al principio del capítulo. Después puede Vd., por ejemplo, trabajar con el programa de administración de almacén y de volumen de ventas. Pero para esto, primero debe existir este programa. Esto es lo que vamos a realizar ahora.

Programa de administración de almacén y volumen de ventas

El programa Busyman deberá ocuparse de las siguientes tareas: El stock de artículos, que pueden ser hasta 750, deberá ser clasificado según grupos de artículo y número en este grupo, memorizado y administrado con los precios de compra. De entrada se puede comprobar en todo momento la existencia de artículos de un tipo, y naturalmente el valor que representa esta existencia. Además deberá ser posible un resumen de todos los grupos de artículos y la existencia total.

Para estar siempre al corriente, es necesario introducir al final de cada día las ventas efectuadas. De cada artículo se memoriza el grupo de artículo, número de artículo y precio de venta. Estos datos se utilizan para restar los artículos vendidos de la existencia. Así es posible tener al día la existencia de artículos.

Se ha de contabilizar durante todo el mes la cantidad de ventas de cada artículo. Así es posible saber qué artículos 'van bien' y cuáles no. Al final del mes se efectúa un resumen, en el cual se memorizan el volumen de ventas y las ganancias. Esto se hace a través de todo el año. Estas cifras pueden activarse en cada momento en el programa.

Naturalmente es aconsejable introducir el volumen de ventas y las ganancias, así como otros datos, en el programa gráfico, para que sea posible una representación gráfica de los mismos. Después de fijar nuevamente el color en la primera línea, en la línea 9000 se inicia el menú. Este ofrece las opciones entre 'negocio diario', resumen de existencia y volumen de ventas, introducción y cambio de datos referentes a la existencia, introducción de la fecha, y naturalmente, el fin del trabajo.

Aparte de la pregunta por la fecha, que se aclara por sí misma, se pasa a un submenú después de elegir una opción, el cual indica las posibilidades disponibles. De esta forma en el punto 'ventas diarias', se indican las siguientes opciones: Introducción de ventas, visualizar listado de ventas y efectuar resumen del día. En este último punto se restan de las existencias las ventas efectuadas en este día, de manera que la lista de stocks esté al día. Este proceso se realiza a partir de la línea 1000. Todas las subrutinas necesarias se encuentran entre las líneas 20 y 99.

Desde la línea 2000 comienza la administración de las existencias y del volumen de ventas. Las diferentes posibilidades se indican en un submenú y son concretamente: resumen de existencias, resumen del mes, resumen anual, así como efectuar la liquidación al final del mes.

A partir de la línea 5000 se lleva a cabo la introducción de datos básicos, así como la modificación de existencias por ejemplo en caso de devoluciones, cambios o envíos retardados. En la línea 6000, finalmente, se realiza la memorización de datos y el retorno al programa 'run'.

Las variables y campos esenciales del programa son los siguientes:

max	Indica el número máximo de tipos de productos que pueden ser memorizados
b\$(max,2)	Indica para cada tipo de artículo el grupo de producto y el número de artículo.
w\$(wg,12)	Indica el nombre de los grupos de productos
wg	Cantidad de grupos de productos
e(max)	Indica el precio de compra del artículo

z(max)	Contiene la existencia en almacén de los artículos
y\$(max)	Memoriza las cantidades vendidas de cada artículo durante el mes corriente
anz	Determina la cantidad máxima de productos, que pueden ser introducidos diariamente como ventas.
g\$(anz,2)	Memoriza los grupos de productos y número de artículos de las ventas diarias.
v(anz)	Indica los precios correspondientes a las ventas
o(wg)	Es una magnitud auxiliar, que determina en qué lugar de la memoria existen los diferentes grupos de productos. Esto ayuda a tener acceso más rápido a los determinados artículos.
mwst	Manipula el impuesto sobre el valor añadido
j(12,6)	Indica las cifras de ventas, volúmenes de ventas y ganancias en el mes y año corrientes.
r(wg,3)	Es una memoria intermedia para cifras de ventas, precios de compra y venta para la liquidación diaria.
u(wg,3)	Como r(wg,3), pero para el volumen mensual.

A parte de estas variables indicadas se aplican una serie de variables, cuyo significado queda claro al leer el listado.

Veamos aquí el listado del programa Busyman.

```

1 REM *****
2 REM *****
3 REM
4 REM F E R R E - M O R E T
5 REM
6 REM *****
7 REM *****

9 BORDER 0: PAPER 0: INK 6: CLS : REM ? pcb
10 LET m=9000: GO TO m
15 INPUT "Fecha = ENTER=";(d$);' LINE m$
16 IF m$<>"" THEN LET d$=m$
17 RETURN
20 LET d$="9 Julio 1985": LET anz=20: GO SUB 75: GO SUB 15
30 LET e$="EneFebMarAbrMayJunJulAgoSepOctNovDic": LET e$=e
$+e$
40 LET em=1: DIM j(12,6): DIM w$(1,12)
50 LET wg=1: DIM o(wg): LET o(1)=1
60 DIM u(wg,3): LET max=100: LET mwst=.14: LET hu=100: LET
xx=10000: LET m1=1+mwst
62 GO SUB 70
65 LET stkj=0: LET eks=0: LET vks=0: LET ekjs=0: LET vkjs=
0
67 GO TO m
70 DIM b$(max,2): DIM e(max): DIM z(max): RETURN
75 DIM g$(anz,2): DIM v(anz): RETURN
80 PRINT AT 21,0;"****ENTRADA INCORRECTA !*****"
82 FOR i=1 TO 10: BEEP .05,i: NEXT i
84 PRINT AT 21,0;,,: RETURN
90 PRINT ' INK 5;"No"+CHR$(8); OVER 1;"_": PRINT INK 5;
" de art. piezas vend. resto": RETURN
95 PRINT 'TAB 8;"FIN DE LA LISTA": RETURN
99 CLS: PRINT AT 9,6;"POR FAVOR, ESPERE ": RETURN
100 LET f=0: IF m$="" THEN GO TO 108: REM wg
104 FOR l=1 TO wg: IF VAL m$=1 THEN RETURN
106 NEXT l
108 LET f=1: RETURN
110 LET f=0: IF m$="" THEN GO TO 118: REM numero de articu
lo
111 FOR l=1 TO LEN m$
112 IF CODE m$(l)<48 OR CODE m$(l)>57 THEN GO TO 118
113 NEXT l

```

```

114 IF VAL m$>=1 AND VAL m$<=250 THEN RETURN
118 LET f=1: RETURN
120 LET f=0: LET su1=0: REM ek/vk
121 IF m$="" THEN GO TO 128
123 FOR l=1 TO LEN m$
124 IF su1=0 AND m$(1)="." THEN LET su1=1: GO TO 126
125 IF CODE m$(1)<48 OR CODE m$(1)>57 THEN GO TO 128
126 NEXT l
127 IF su1<=1 THEN RETURN
128 LET f=1: RETURN
130 LET f=0: IF m$="" THEN GO TO 138: REM stk
134 FOR l=1 TO LEN m$
135 IF CODE m$(1)<48 OR CODE m$(1)>57 THEN GO TO 138
136 NEXT l
137 RETURN
138 LET f=1: RETURN
140 GO SUB 130: IF f=1 THEN GO TO 5100
141 IF VAL m$>750 OR VAL m$<1 THEN GO TO 5100
143 LET max=VAL m$: GO SUB 70
144 GO TO 5100
145 GO SUB 130: IF f=1 THEN GO TO 5100
146 IF VAL m$>150 OR VAL m$<1 THEN GO TO 5100
148 LET anz=VAL m$: GO SUB 75
149 GO TO 5100
150 INPUT "Borrar datos ? (s/n)";m$: IF m$<>"s" THEN GO T
O 5000
154 LET b$(i)=" ": LET z(i)=0: LET e(i)=0: LET y$(i)=" "
156 GO SUB 99
158 FOR k=i TO max-1
160 LET b$(k)=b$(k+1): LET z(k)=z(k+1): LET e(k)=e(k+1): LE
T y$(k)=y$(k+1)
162 IF b$(k,1)=" " THEN GO TO 5000
164 NEXT k
166 LET b$(max)=" ": LET z(max)=0: LET e(max)=0: LET y$(ma
x)=" "
167 GO SUB 350
168 GO TO 5000
170 GO SUB 120: IF f=1 THEN GO TO 5100
172 LET mwst=VAL m$/100: LET m1=1+mwst: GO TO 5100
180 INPUT "Primer mes anualidad comercial ?(Enero = 1) "; L
INE a$: GO SUB 130
184 FOR l=1 TO LEN a$

```

```

186 IF CODE a$(1)<48 OR CODE a$(1)>57 THEN GO TO 182
188 NEXT 1
190 LET em=VAL a$
192 IF em<1 OR em>12 THEN GO TO 182
194 RETURN
200 IF INKEY$<>" " THEN GO TO 200: REM correccion
201 POKE 23689,PEEK 23689+1
202 PRINT ,,
203 POKE 23689,PEEK 23689+1
204 RETURN
250 LET per=0: REM ordenar
252 FOR l=1 TO max-1: IF b$(l+1)=" " THEN GO TO 260
253 IF b$(l)>b$(l+1) THEN LET per=per+1
254 NEXT l
260 IF per=0 THEN GO TO 285
264 FOR s=1 TO 2 STEP -1
268 FOR k=1 TO s-1
270 IF b$(s)<b$(k) THEN GO TO 291
272 NEXT k
280 NEXT s
285 GO SUB 350
290 GO TO 5000
291 LET m$b$(s): LET pe=e(s): LET pz=z(s): LET x$=y$(s)
292 FOR j=s TO k+1 STEP -1
293 LET b$(j)=b$(j-1): LET e(j)=e(j-1): LET z(j)=z(j-1): LET y$(j)=y$(j-1)
294 NEXT j
295 LET b$(k)=m$: LET e(k)=pe: LET z(k)=pz: LET y$(k)=x$
296 GO TO 250
300 CLS : PRINT INK 5;" GRUPOS DE PRODUCTOS :": FOR i=
1 TO wg: PRINT INK 4;i;" ";w$(i): NEXT i: RETURN
350 REM orden
352 LET k=2
355 FOR i=1 TO wg: LET o(i)=1: NEXT i
360 FOR i=2 TO max
365 IF b$(i,1)=" " THEN GO TO 390
370 IF b$(i,1)>b$(i-1,1) THEN LET o(k)=i: LET k=k+CODE b$(
i,1)-CODE b$(i-1,1)
375 IF k>wg THEN GO TO 390
380 NEXT i
392 FOR i=2 TO wg: IF o(i)<o(i-1) THEN LET o(i)=o(i-1)
394 NEXT i

```

```

399 RETURN
400 DEF FN r(a,b,c)=INT (100*a/b-c)/100: DEF FN k(a,b,c)=IN
T (10000*a*b/c)/100: DEF FN l(a,b)=INT (10000*a/b)/100
500 CLS : PRINT "          Problema          "
510 LET a1=i
520 REM GA = Grupo Articulos          wg es la variable
que le corresponde
525 PRINT ' INK 5;"Este articulo no esta en almacen"' INK
2;" GA ART.No";CHR$ (8); INK 2; OVER 1;"_ PVP"

530 PRINT : PRINT i;" ";w$(t);" ";CODE g$(i,2);" Pts ";v(i
)
550 PRINT '''1 = Introducir de nuevo recibo","2 = Borrar r
ecibo","3 = Admitir articulo en almacen","4. = Menu"
560 PAUSE 0: LET m$=INKEY$: IF m$="4" THEN GO TO m
570 IF m$="1" THEN GO TO 600
572 IF m$="2" THEN GO TO 700
574 IF m$="3" THEN GO TO 800
578 GO TO 560
600 PRINT "***** Nueva entrada *****"
610 INPUT "GA =";m$: GO SUB 100
615 IF f=1 THEN GO TO 610
620 PRINT m$;" ";: LET g$(i,1)=CHR$ (VAL m$)
622 INPUT "Articulo No";CHR$ (8); OVER 1;"_ =" ;m$: GO SUB
110
624 IF f=1 THEN GO TO 622
628 LET g$(i,2)=CHR$ (VAL m$): PRINT m$;
640 INPUT "PVP = ";m$: GO SUB 120
645 IF f=1 THEN GO TO 640
648 LET v(i)=VAL m$: PRINT "Pts ";v(i)
680 PAUSE 50: GO TO 1360
700 PRINT "POR FAVOR, ESPERE"
720 FOR k=i TO anz-1: LET g$(k)=g$(k+1): LET v(k)=v(k+1): N
EXT k
740 LET g$(anz)=" ": LET v(anz)=0: GO TO 1360
800 FOR k=o(wg) TO max
820 IF b$(k,1)=" " THEN GO TO 850
830 NEXT k
840 GO TO 999
850 LET b$(k)=g$(i)
860 INPUT "CP (compra) = ";m$: GO SUB 120
864 IF f=1 THEN GO TO 860

```

```

866 LET e(k)=VAL m$
870 INPUT "Cantidad antes de venta = ";m$: GO SUB 130
874 IF f=1 THEN GO TO 870
876 LET z(k)= VAL m$
899 GO TO 1360
900 LET x$= CHR$ wag+CHR$ fn
920 FOR l=o(wag) TO max
922 IF b$(l,1)=" " THEN GO TO 935
925 IF b$(l)=x$(1 TO 2) THEN GO TO 950
930 NEXT l
935 PRINT AT 21,8;"no existe": PAUSE 150: CLS : GO SUB 300:
GO TO 1110
950 IF z(1)<1 THEN GO TO 935
960 GO TO 1120
999 CLS : PRINT "no hay espacio": PAUSE 200: GO TO m
1000 CLS : PRINT INK 3;TAB 7;"ventas diarias"" INK 5;;d$:
PRINT "" 1 = Introducir ventas"" 2 = Lista de ventas
"" 3 = Liquidacion del dia"" 4 = Menu": PAUSE 0
1010 LET u$="0"
1030 LET m$=INKEY$
1032 IF m$="1" THEN GO TO 1100
1034 IF m$="2" THEN GO TO 1200
1036 IF m$="3" THEN GO TO 1300
1038 IF m$="4" THEN GO TO 9000
1039 GO TO 1030
1100 CLS : IF g$(1,1)=" " THEN GO TO 1104
1102 INPUT "nuevo dia? (s/n) "; LINE m$
1103 IF m$="s" THEN DIM g$(anz,2): DIM v(anz)
1104 FOR i=1 TO anz: IF g$(i,1)=" " THEN GO TO 1106
1105 NEXT i
1106 LET n=i: GO SUB 300
1110 PRINT "ENTRADA      ventas      "; INK 5;d$
1112 PRINT AT wg/2+4,0; INK 7;"R=Repite F=Fin C=Correcci
on"
1114 PRINT INK 3;" Gr.art.      Art.No";CHR$ (8); OVER 1;"
PVP"
1116 PRINT AT wg/2+6,0;
1119 LET su1=0: LET i=0: LET t$=""
1120 PRINT " ";CHR$ 8;
1122 PAUSE 0: LET m$=INKEY$: BEEP .01,1
1124 IF CODE m$=13 THEN GO TO 1150
1125 IF m$="r" THEN PRINT wag;TAB 11;fn;TAB 20;vvp: GO TO 1

```

```

186
1126 IF m$="c" THEN GO TO 1104
1127 IF m$="f" THEN GO TO 1000
1128 IF i=2 AND m$="." THEN GO TO 1130
1129 IF CODE m$>57 OR CODE m$<48 THEN GO TO 1121
1130 PRINT m$; LET t$=t$+m$
1140 IF INKEY$<>"" THEN GO TO 1140
1145 GO TO 1120
1150 IF t$="" THEN GO TO 1104
1151 LET i=i+1; IF i>1 THEN GO TO 1160
1154 FOR k=1 TO wg: IF VAL t$=k THEN GO TO 1158
1155 NEXT k
1156 GO SUB 80: GO TO 1104
1158 LET wag=VAL t$: LET t$=""
1159 PRINT TAB 11;: GO TO 1120
1160 IF i>2 THEN GO TO 1170
1162 LET m$=t$
1164 GO SUB 110: IF f=1 THEN GO SUB 80: GO TO 1104
1166 LET fn=VAL t$: LET t$=""
1168 PRINT TAB 20;: GO TO 900
1169 GO TO 1120
1170 REM precio
1172 LET m$=t$: GO SUB 120
1174 IF f=1 THEN GO SUB 80: GO TO 1104
1176 LET vkp=VAL t$
1178 PRINT " ": LET t$=""
1186 LET g$(n,1)=CHR$ wag
1188 LET g$(n,2)=CHR$ fn
1190 LET v(n)=vkp
1191 LET n=n+1
1192 IF n>anz THEN CLS : PRINT FLASH 1;"Lista ventas compl
eta. LIQUIDAR!": PAUSE 100: GO TO 1000
1193 IF INT (n/10)>=n/10 THEN GO TO 1104
1195 GO TO 1119
1200 CLS : LET su1=0: PRINT INK 3;TAB 5;"Lista de ventas"E
XP 5;d$
1220 FOR i=1 TO anz
1230 IF g$(i,1)=" " THEN GO TO 1270
1240 PRINT i;TAB 3;w$(CODE g$(i,1));TAB 16;CODE g$(i,2);TAB
28;"Pts"
1250 LET su1=su1+v(i)
1260 NEXT i

```

```

1270 PRINT ' INK 4;"Venta total = ";su1;" Pts"
1280 PAUSE 0: GO TO 1000
1300 CLS : IF u$="s" THEN GO TO 1310
1305 LET a1=1: LET u$="0"
1310 PRINT INK 3;" Volumen de ventas diario", INK 5;d$
1320 PRINT INK 4;"Grupo prod. Cant. Venta Compra"
1330 FOR i=1 TO wg: PRINT AT i+2,0;w$(i): NEXT i
1340 LET a2=anz
1350 DIM r(wg,3)
1360 FOR i=a1 TO a2
1361 IF g$(i,1)=" " THEN GO TO 1380
1362 LET t=CODE g$(i,1)
1365 LET r(t,1)=r(t,1)+1: LET r(t,3)=r(t,3)+v(i)
1367 GO SUB 1500
1368 IF l>max THEN LET l=max
1370 LET u(t,1)=u(t,1)+1: LET u(t,3)=u(t,3)+v(i)
1375 NEXT i
1380 LET p=o: LET su1=0: LET su2=0
1385 FOR i=1 TO wg
1386 PRINT AT i+2,13;r(i,1);TAB 16;r(i,3);TAB 24;r(i,2)
1387 LET p=p+r(i,1): LET su1=su1+r(i,2): LET su2=su2+r(i,3)
1388 NEXT i
1389 PRINT INK 2;" S U M A = ";TAB 13;p;TAB 16;su2;TAB 24;s
u1
1390 PRINT "Calculo ":" IF su1>0 AND su2>0 THEN LET re=FN r
(su2,m1,su1): PRINT "RE=Pts ";re;"Calc = ";FN k(re,m1,su2);"
/";FN l(re,su1)
1395 PRINT "Mes corriente""Ventas=";vks;" Pts.";TAB 13,"Co
mpra=";eks;" Pts.": IF eks>0 AND vks>0 THEN LET re=FN r(vks
,m1,eks):: PRINT "RE = Pts ";re;"Calc = ";FN k(re,m1,vks);"
/";FN l(re,eks)
1400 IF u$="s" THEN DIM g$(anz,2): DIM v(anz): PAUSE 0: GO
TO 1000
1470 IF u$="0" THEN INPUT "Renovar existencias ? (s/n)"; LI
NE u$: IF u$="s" THEN CLS : GO TO 1310
1490 GO TO 1000
1500 REM buscar compras
1505 LET su1=CODE g$(i,1)
1510 FOR l=o(su1) TO max
1515 IF b$(l,1)=" " THEN GO TO 500
1520 IF g$(i)>b$(l) THEN NEXT l
1535 IF z(l)<=0 THEN GO TO 1580

```

```

1540 LET r(t,2)=r(t,2)+e(1)
1545 IF u$<>"s" THEN GO TO 1570
1550 LET y$(1)=CHR$(CODE y$(1)+1)
1555 LET u(t,2)=u(t,2)+e(1)
1560 LET z(1)=z(1)-1
1565 LET eks=eks+e(1): LET vks=vks+v(i)
1566 LET stkj=stkj+1: LET eksj=eksj+e(1): LET vksj=vksj+v(i)
1570 RETURN
1580 NEXT i
1590 GO TO 500
1700 FOR k=1 TO max-1
1705 IF b$(k,1)=" " THEN RETURN
1710 IF z(k)>0 THEN GO TO 1750
1720 FOR l=k TO max-1
1730 LET y$(1)=y$(1+1): LET b$(1)=b$(1+1): LET e(1)=e(1+1):
LET z(1)=z(1+1)
1740 NEXT i
1745 LET b$(max)=" ": LET k=k-1
1750 NEXT k
1760 RETURN
2000 CLS : PRINT INK 3;"VOLUMEN DE VENTAS Y EXISTENCIAS ";
INK 5;d$"'
2010 PRINT " 1 = Resumen de existencias"" 2 = Resumen me
nsual"" 3 = Detalle del mes"" 4 = Resumen anual"" 5
= Liquidacion/comienzo mes"" 6 = Menu"
2020 PAUSE 0: LET m$=INKEY$: IF m$="6" THEN GO TO m
2026 IF m$="1" THEN GO TO 5800
2028 IF m$="5" THEN GO TO 2900
2032 IF m$="2" THEN GO TO 2100
2034 IF m$="3" THEN GO TO 2400
2036 IF m$="4" THEN GO TO 2600
2040 GO SUB 80: GO TO 2000
2100 CLS : PRINT INK 3;" VOLUMEN DE VENTAS MENSUAL " INK 6
;d$
2110 PRINT INK 5;"Gr.articulo  Pzas Ven  Com  ": LET p=
0: LET su1=0: LET su2=0
2150 FOR i=1 TO wg
2160 PRINT w$(i);u(i,1);TAB 19;u(i,3);TAB 26;u(i,2)
2180 IF u(i,2)<=0 OR u(i,3)<=0 THEN GO TO 2195
2190 LET re=FN r(u(i,3),m1,u(i,2)): PRINT "RE=Pts ";re"Calc
: ";FN k(re,m1,u(i,3));"/";FN l(re,u(i,2))"'
2200 NEXT i

```

```

2220 PRINT " INK 3;"S U M A = ";TAB 16;p;TAB 19;su2;TAB 26;
su1
2230 PRINT " INK 3;"Mes corriente :"" INK 4;"V = Pts ";vks
;TAB 16;"C = Pts ";eks: IF eks>0 AND vks>0 THEN LET re=FN r
(vks,m1,eks): PRINT "RE=Pts ";re;"Calc:";FN k(re,m1,vks);"/"
;FN l(re,vks)
2240 GO SUB 95
2300 PAUSE 0: GO TO 2000
2400 CLS : PRINT "Detalle volumen mensual ventas" INK 5;d$
,
2435 LET l=1
2440 FOR i=1 TO max
2450 IF b$(i,1)=" " THEN GO TO 2480
2454 IF i=o(1) AND i>1 THEN INPUT INK 4;"ENTER = seguir ";
LINE m$: CLS
2455 IF i=o(1) THEN PRINT " "; INK 4;l;" ";w$(1);: GO SUB 90
: IF l<wg THEN LET l=l+1: GO TO 2455
2460 PRINT CODE b$(i,2);TAB 6;e(i);TAB 15;CODE y$(i)-32;TAB
27;z(i)
2470 NEXT i
2480 GO SUB 95
2490 PAUSE 0: GO TO 2000
2600 CLS : PRINT TAB 8;"Vision anual" INK 5;d$''
2610 FOR i=1 TO 12
2615 IF j(i,2)=0 OR j(i,3)=0 THEN GO TO 2650
2620 PRINT " INK 2;"MES Pzs Ventas Compras"
2625 PRINT INVERSE 1;e$(3*(em-1+i)-2 TO 3*(em-1+i)); INVERS
E 0;TAB 8;j(i,1);TAB 14;j(i,3);TAB 24;j(i,2)
2630 LET re=FN r(j(i,3),m1,j(i,2))
2635 PRINT "RE = ";re;TAB 15;"Calc ";FN k(re,m1,j(i,3));"/"
;FN l(re,j(i,2))
2640 PRINT "Escalon";TAB 8;j(i,4);TAB 14;j(i,6);TAB 24;j(i,
5)
2645 LET re=FN r(j(i,6),m1,j(i,5)): PRINT "RE = ";re;TAB 15;
"Calc ";FN k(re,m1,j(i,6));"/";FN l(re,j(i,5))
2650 NEXT i
2655 PRINT " INK 2;"Anyo corriente:"; INK 4;"V=Pts.";vksj;T
AB 16;"C=Pts.";eksj
2660 IF eksj>0 AND vksj>0 THEN LET re=FN r(vksj,m1,eksj): P
RINT "RE=Pts";re: PRINT "Calc = ";FN k(re,m1,vksj);"/";FN l(
re,eksj)
2665 GO SUB 95
2670 PAUSE p: GO TO 2000
2900 CLS : INPUT "Introducir todos los datos?(s/n)"; LINE m$
: IF m$<>"s" THEN GO TO m
2920 PRINT "Por favor, espere"
2921 FOR k=1 TO 12

```

```

2922 IF j(k,1)=0 THEN GO TO 2925
2923 NEXT k
2925 LET j(k,1)=p: LET j(k,2)=eks: LET j(k,3)=vks
2926 LET j(k,4)=stkj: LET j(k,5)=eksj: LET j(k,6)=vksj
2930 DIM u(wg,3): DIM y$(max)
2955 LET eks=0: LET vksj=0: DIM g$(anz,2): DIM v(anz)
2960 GO SUB 1700
2965 GO SUB 350
2970 INPUT "Fin anyo comercial? (s/n)"; LINE a$
2975 IF a$<>"s" THEN GO TO 2000
2976 INPUT "Introduzca fin, por favor"; LINE m$
2977 LET a$="0"
2978 IF m$="fin" THEN LET a$="s"
2979 CLS : PRINT AT 10,13; FLASH 1;"Fin": PAUSE 150
2980 IF a$="s" THEN LET eksj=0: LET vksj=0: LET stkj=0
2985 IF a$="s" THEN DIM j(12,6)
2999 GO TO 2000
5000 CLS : PRINT INK 3;" Entrada/modificacion datos ": P
RINT "" INK 6;"1 = Modificar existencias""2 = Introducir e
xistencias""3 = Componer Grupos Productos""4 = Ordenar e
xistencias""5 = M E N U"
5020 PAUSE 0: LET m$=INKEY$: IF m$="5" THEN GO TO m
5050 IF m$="4" THEN CLS : PRINT FLASH 1;"Por favor, espere
": GO TO 250
5054 IF m$="1" THEN GO TO 5600
5056 IF m$="3" THEN GO TO 5100
5058 IF m$="2" THEN GO TO 5300
5060 GO SUB 80: GO TO 5000
5100 CLS : PRINT "Introducir GP"
5124 PRINT ""1 = Numero grupos de productos"
5126 PRINT ""2 = Nombres grupos de productos"
5127 PRINT ""3 = Comienzo anyo comercial=";e$(3$em-2 TO 3$em
)
5128 PRINT ""4 = IVA = ";100$mwt$;"%""5 = Numero de tipos
de prod. ";max
5129 PRINT ""6 = Max. ventas diarias = ";anz""7 = M E N U"
5130 PAUSE 0: LET m$=INKEY$: IF m$="7" THEN GO TO 5000
5150 IF m$="4" THEN INPUT "Cual es el IVA ?(en %) ";m$: GO
TO 170
5154 IF m$="5" THEN INPUT "Cantidad de tipos de productos
(1-750)";m$: GO TO 140
5156 IF m$="6" THEN INPUT "Cantidad de ventas por dia
(1-150)";m$: GO TO 145
5160 IF m$="3" THEN GO SUB 180: GO TO 5100
5180 IF m$="2" THEN GO TO 5220
5190 IF m$="1" THEN GO TO 5200
5195 GO TO 5130
5200 INPUT "Cantidad de GP (1-20)= ";m$
5210 GO SUB 130: IF f=1 THEN GO TO 5200
5212 LET wg=VAL m$: IF wg<1 OR wg>20 THEN GO TO 5200
5213 DIM o(wg): DIM w$(wg,12): DIM u(wg,3)
5214 FOR i=1 TO wg: LET o(i)=1: NEXT i
5215 PRINT AT 3,28;" ";AT 3,28;wg
5216 GO TO 5100
5220 IF wg<=0 THEN PRINT ""Cantidad de GP = 0": PAUSE 150:
GO TO 5100
5222 CLS : PRINT
5224 FOR i=1 TO wg

```

```

5226 PRINT : PRINT "GP ";i;" = ";TAB 8;w$(i);: INPUT LINE m
$;: IF m$<>" THEN LET w$(i)=m$
5227 POKE 23689,PEEK 23689+1: PRINT PAPER 1;i;TAB 8;w$(i)
5228 NEXT i
5290 PAUSE 0: GO TO 5000
5300 CLS
5303 FOR s=o(wg) TO max
5304 IF b$(s,1)<>" " THEN GO TO 5306
5305 GO TO 5310
5306 NEXT s
5308 GO TO 999
5310 GO SUB 300: PRINT : PRINT INK 3;"INTRODUCIR E X I S T
E N C I A S"
5330 INPUT "Numero del GP (0=fin) = "; LINE m$
5332 IF m$="0" THEN GO TO 250
5333 GO SUB 100
5334 IF f=1 THEN GO SUB 80: GO TO 5310
5336 LET i=VAL m$
5400 CLS : PRINT "Entrada "; INK 4;w$(i)
5402 PRINT "Art.-No";CHR$(8); OVER 1; "_"; " precio compra
cantidad"
5405 INPUT "Articulo-No";CHR$(8); OVER 1; "_"; " (0=fin)= ";
LINE m$: IF m$="0" THEN GO TO 5300
5406 GO SUB 110
5407 IF f=1 THEN GO SUB 80: GO TO 5405
5410 LET fn=VAL m$: PRINT fn;
5415 LET m$=CHR$(i)+CHR$(fn)
5420 FOR l=o(i) TO max: IF b$(l,1)=" " THEN GO TO 5430
5421 IF b$(l)=m$ THEN GO TO 5425
5422 NEXT l
5424 GO TO 5430
5425 PRINT " existe ";TAB 14;e(1): PAUSE 250: GO TO 5400
5430 INPUT "Precio compra = "; LINE m$: GO SUB 120
5432 IF f=1 THEN GO SUB 80: GO TO 5430
5436 LET ek=VAL m$: PRINT TAB 14;ek;
5440 INPUT "Cantidad = "; LINE m$: GO SUB 120
5443 IF f=1 THEN GO TO 5440
5446 LET az=VAL m$: PRINT TAB 26;az
5460 INPUT "Entrada ok? (s/n)"; LINE m$
5465 IF m$="s" THEN GO TO 5500
5470 GO SUB 201
5480 GO TO 5405
5500 LET b$(s,1)=CHR$(i)
5502 LET b$(s,2)=CHR$(fn)
5508 LET e(s)=ek
5510 LET z(s)=az
5520 LET s=s+1
5530 IF s>max THEN PRINT "No hay espacio para mas entradas"
;m150: GO TO m
5550 GO TO 5405
5600 GO SUB 300
5610 PRINT AT 14,0;"1 = Completar entrega""2 = Devolucion""
"3 = Cambio""4 = Correccion de una entrada""5 = M E N U"
5615 PAUSE 0: LET m$=INKEY$: IF m$="5" THEN GO TO 5000
5616 IF m$>"4" OR m$<="0" THEN GO SUB 80: GO TO 5615
5617 LET a$=m$
5620 IF m$="1" THEN PRINT AT 14,4;"Completar entrega"
5621 IF m$="2" THEN PRINT AT 15,4;"Devolucion"
5622 IF m$="3" THEN PRINT AT 16,4;"Cambio"

```

```

5623 IF m$="4" THEN PRINT AT 17,4;"Correccion de una entrada"
5650 INPUT "GP = "; LINE m$: GO SUB 100
5651 IF f=1 THEN GO SUB 80: GO TO 5650
5652 LET w=VAL m$
5658 PRINT AT 19,0; INK 5;w$(w); INK 6;TAB 17;" ";
5660 INPUT "Articulo No";CHR$(8); OVER 1;"_ = "; LINE m$: GO SUB 110
5662 IF f=1 THEN GO SUB 80: GO TO 5660
5668 LET fn=VAL m$: PRINT fn
5690 FOR i=o(w) TO max
5695 GO SUB 130: IF f=1 THEN GO SUB 80: GO TO 5794
5700 IF CODE b$(i,1)<>w THEN GO TO 5720
5702 IF CODE b$(i,2)<>fn THEN GO TO 5720
5710 GO TO 5730
5720 NEXT i
5725 PRINT "No lo he hallado": PAUSE 150: GO TO 5000
5730 PRINT AT 20,0; INK 3;"C:";e(i); " Pts";TAB 17;"existencias:";z(i)
5750 IF a$="1" THEN INPUT "cantidad = ";su1: LET z(i)=z(i)+su1
5755 IF a$="2" THEN INPUT "cantidad = ";su1: LET z(i)=z(i)-su1
5758 IF a$="3" THEN LET z(i)=z(i)+1
5759 IF a$="4" THEN GO TO 5778
5760 PRINT TAB 11; INK 2;"Nuevas existencias:";z(i)
5770 INPUT "M = MENU ** ENTER = seguir"; LINE m$
5775 IF m$="m" THEN GO TO 5000
5776 GO TO 5600
5778 PRINT AT 21,0,,,: INPUT "(ENTER=OK B=borrar) GP="; LINE m$: IF m$="" THEN GO TO 5783
5779 IF m$="b" THEN GO TO 150
5781 GO SUB 100: IF f=1 THEN GO SUB 80: GO TO 5780
5782 LET b$(i,1)=CHR$(VAL m$)
5783 PRINT AT 19,0; PAPER 3;w$(CODE b$(i,1)): INPUT "Nuevo articulo No";CHR$(8); OVER 1;"_ = (ENTER = OK) "; LINE m$: IF m$="" THEN GO TO 5786
5784 GO SUB 110: IF f=1 THEN GO SUB 80: GO TO 5783
5785 LET m$(i,2)=CHR$(VAL m$)
5786 PRINT AT 19,18;" ";AT 19,18; PAPER 3;CODE b$(i,2)
5790 INPUT "Compra (ENTER=OK) "; LINE m$: IF m$="" THEN GO TO 5797
5792 GO SUB 120: IF f=1 THEN GO SUB 80: GO TO 5790
5793 LET e(i)=VAL m$
5794 PRINT AT 20,3;" ";AT 20,3; PAPER 3;e(i); " Pts"
: INPUT "Nuevo num. piezas = (ENTER=OK)"; LINE m$: IF m$="" THEN GO TO 5794
5796 LET z(i)=VAL m$
5797 PRINT AT 20,25; PAPER 3;z(i)
5799 GO TO 250
5800 CLS : PRINT INK 3;" Vision existencias "
INK 4;TAB 7;"1 = Vision""TAB 7;"2 = Detalle GP""TAB 7;"3 = MENU"
5810 PAUSE 0: LET m$=INKEY$: IF m$="3" THEN GO TO 5810
5811 IF m$<>"1" AND m$<>"2" THEN GO TO 5810
5812 IF m$="1" THEN CLS : GO TO 5818
5814 GO SUB 300: INPUT "GP (0= todos) = ";su3: IF su3>wg THEN GO SUB 80: GO TO 5814

```

```

5815 LET w=su3
5816 GO TO 5900
5818 DIM r(wg,2): PRINT INK 3;TAB 7;"VISION-EXISTENCIAS" I
NK 2'"Existencias Valor"
5820 FOR i=1 TO max
5822 IF b$(i,1)=" " THEN GO TO 5840
5824 LET k=CODE b$(i,1)
5826 LET r(k,1)=r(k,1)+z(i): LET r(k,2)=r(k,2)+z(i)*e(i)
5830 NEXT i
5840 LET su1=0: LET su2=0
5850 FOR i=1 TO wg: PRINT w$(i);TAB 15;r(i,1);TAB 21;r(i,2):
LET su1=su1+r(i,1): LET su2=su2+r(i,2): NEXT i
5860 PRINT "' INK 3;" S U M A =" ;TAB 15;su1;TAB 21;su2
5870 PAUSE 0: GO TO 2000
5900 IF w=0 THEN FOR w=1 TO wg
5902 CLS : PRINT INK 6;d$; INK 3'"Existencias ";w$(w)' INK
5;"Art.No";CHR$(8); OVER 1;"_ precio com. cantidad"
5905 LET su1=0: LET su2=0
5910 IF b$(i,1)=" " THEN GO TO 5932
5920 IF CODE b$(in1)=w THEN PRINT CODE b$(i,2);TAB 15;e(i)
;TAB 26;z(i): LET su1=su1+e(i)*z(i): LET su2=su2+z(i)
5930 NEXT i
5932 PRINT ' INK 3;" S U M A Pts ";su1;TAB 26;su2
5960 INPUT INK 2;"ENTER=seguir"; LINE m$: IF su3=0 THEN NE
XT w
5970 GO TO 2000
6000 CLS : PRINT AT 9,0;"ALMACENAR PROGRAMA/DATOS ? (s/n)":
PAUSE 0
6040 IF INKEY$="n" THEN REM LOAD $"m";1;"run"
6050 CLS : PRINT "ALMACENAR DATOS :";AT 5,0;"1 = Microdrive"
""2 = Casette""3 = M E N U": PAUSE 0: LET m$=INKEY$
6060 IF m$="1" THEN GO TO 6200
6070 IF m$="2" THEN CLS : GO TO 6100
6080 IF m$="3" THEN GO TO 9000
6100 PRINT AT 9,0;" Coloque la cinta BUSYMAN, prepare e
l cassette para GRABAR y pulse ENTER"
6110 SAVE "BUSYMAN" LINE 1
6120 PRINT " Rebobine la cinta para su VERIFICACION
y pulse PLAY": VERIFY ""
6190 GO TO 6050
6200 PRINT """Por favor, espere"
6220 REM ERASE "m";1;"Busyman"
6230 REM SAVE $"m";1;"Busyman" LINE 1
6240 REM VERIFY $"m";1;"Busyman"
6290 GO TO 6050
9000 CLS : PRINT INK 3;" Menu de administracion almacen **
** y volumen de ventas ****";AT 2,0; INK 6;d$;TAB 27;"? pcb
"
9030 PRINT "" 1 = Negocio diario"" 2 = Vision exi
st. y vol.ven."" 3 = Entrada/variacion datos"" 4
= Fecha"" 5 = F I N"
9100 PAUSE 0: LET m$=INKEY$: IF m$="4" THEN GO SUB 15: GO T
O m
9116 IF m$="1" THEN GO TO 1000
9117 IF m$="2" THEN GO TO 2000
9118 IF m$="3" THEN GO TO 5000
9119 IF m$="5" THEN GO TO 6000
9120 GO SUB 80: GO TO 9040

```

Aplicación de Busyman

Para reservar los espacios de memoria para todas las variables, hay que introducir GO TO 20 después de asegurar el programa, lo cual se efectúa nuevamente con la opción LINE 1. A continuación se sigue con la opción 3 al submenú para entrada/modificación de datos. Aquí nuevamente se elige la opción 3: Composición de los grupos de productos.

Ahora se introduce la cantidad de grupos de productos diferentes. Se debería introducir un grupo de más antes que un grupo de menos, ya que esta cantidad no puede ser modificada posteriormente sin perder datos. Luego puede asignarse un nombre a cada grupo, el cual puede tener como máximo 12 caracteres. Estos nombres pueden ser modificados posteriormente en todo momento.

La cantidad de tipos de productos también ha de fijarse. El Spectrum 48K permite administrar hasta 750 artículos con este programa. Si fueran necesarios menos artículos, es aconsejable introducir un número más pequeño, ya que algunas operaciones en el programa serían más rápidas. La cantidad de ventas por día, que pueden ser elaboradas en una liquidación, ha de ser introducida igualmente. No debería sobrepasar la cifra de 150. Esto sólo es aplicable a esta versión del programa. Naturalmente, Vd. puede elegir una cifra mayor si la capacidad de memoria lo permite, o sea si Vd. tiene que administrar menos de 750 artículos. Sólo tiene que introducir en la línea 146 la cifra correspondiente en lugar de 150. Como no todos los negociantes comienzan el año en enero, también es posible fijar este mes de inicio. En el resumen anual se asigna el nombre de mes correcto a cada mes liquidado. También es posible modificar el factor correspondiente al IVA.

Después de efectuar estos trabajos, se puede volver al menú principal. Ahora debería introducirse la fecha a través de la opción 4. Y luego puede Vd. introducir las existencias.

Para ello se elige la opción 3 del menú principal. En el submenú que entonces surge, se activa bajo la opción 3 una subrutina (introducir existencia). Primero se exige el número del grupo de artículos que ha de ser tratado, y después el número de artículo, precio de compra y cantidad. Cada entrada ha de ser confirmada. Naturalmente, todas las entradas pueden ser corregidas con la opción 'modificación datos'.

Cuando se han introducido las existencias de todos los grupos de productos, se puede efectuar una visión de todo el almacén. Esto se consigue a través de la segunda opción en el menú principal, que lleva al submenú 'Volumen de ventas y existencias'. El resumen de existencias puede ser realizado por una parte detalladamente para cada grupo de productos, y por otra, para la totalidad de grupos de productos. El resumen mensual indica las ventas para cada artículo en el mes corriente, así como el resto de existencia (opción 3), o un resumen de todos los grupos de productos (opción 2).

A final de cada mes hay que realizar la liquidación del mismo. Así se memorizan el volumen y la ganancias totales (estas cifras pueden ser activadas para el resumen anual), y las cifras de venta del mes corriente se ponen a 0.

El trabajo diario también es fácil de realizar. Desde el menú principal se elige la opción 1 y se llega nuevamente a un submenú. La opción 1 nos lleva hacia la rutina que manipula la entrada de ventas. Se han de introducir el grupo de productos, número de artículo y precio de venta.

Pulsando 'C' se puede efectuar una corrección de la entrada. Con 'R' se repite la última entrada, o sea en la venta de 10 artículos idénticos no hay que introducir 10 veces la misma información, sino únicamente una vez y las restantes nueve veces se pulsa 'R'.

Con 'F' se finaliza la introducción. Tampoco es necesario introducir todas las ventas a la vez. Si ya se ha efectuado una entrada y se elige nuevamente la opción del menú, pueden dejarse los datos en la lista. El contenido de esta lista puede ser visualizado a través de la opción 2 del submenú 'negocio diario'. La lista se borra al efectuar la liquidación diaria y restar de la existencia los artículos vendidos.

Esta liquidación (opción 3) se realiza primero sin modificar las existencias en almacén. Esto sólo se realiza cuando se desea. Así es posible efectuar una liquidación intermedia, para hacerse una idea.

Con seguridad que todos los programas no están ajustados óptimamente a las necesidades de empresas pequeñas. Pero se pueden modificar y ampliar fácilmente. De todas formas representan una estructura, sobre la cual se puede seguir construyendo.

Sin duda alguna, el Spectrum puede conllevar simplificaciones de las tareas en una empresa pequeña, y no hay nada que objetar en contra de utilizarlo para estos deberes.

APENDICE 1: EL JUEGO DE COMANDOS DEL SPECTRUM

Todos los comandos, a excepción de INPUT, DEF y DATA, pueden ser introducidos tanto en un programa como en un comando directo. Varios comandos pueden ser introducidos en una línea de programa, siempre que estén separados por un doble punto (:). Téngase en cuenta, que después de surgir el primer REM, se ignora el resto de la línea. En un comando IF, todos los demás comandos sólo se efectúan si están escritos después de THEN.

Veamos ahora una lista completa de los comandos del Spectrum:

BEEP x,y	Durante x segundos suena, a través de un altavoz incorporado, una nota cuya altura y semitonos están por encima del DO central. Si y es negativa, la altura del sonido estará por debajo del DO. La señal también está en la salida EAR y puede ser amplificada.
BORDER m	Los márgenes laterales de la pantalla, así como el color de fondo de la parte inferior de la pantalla reciben el color m. m debe estar entre 0 y 7.
BRIGHT n	El brillo se determina para los próximos caracteres. n puede ser únicamente 0 (para normal), 1(para brillante) y 8(para transparente).
CAT	Válido sólo para funcionamiento con Microdrive (ver abajo)
CIRCLE x,y,z	Dibuja un círculo con centro en (x,y) y de radio z

CLEAR	Borra todas las variables y deja libre el espacio de memoria que ocupaban. Además se efectúa un RESTORE y CLS, la posición PLOT se sitúa en la esquina izquierda inferior, y la pila de subrutinas de GO SUB es borrada.
CLEAR n	Como CLEAR pero, además, si es posible la variable del sistema RAMTOP es colocada a n. y pone allí la nueva pila de GOSUB.
CLOSE #	Válido sólo para funcionamiento con Microdrive (ver abajo)
CLS	Borra el fichero de presentación visual y, con ello, la pantalla misma.
CONT(inue)	Continúa el programa en el mismo punto en el cual fue interrumpido con un informe diferente a 0.
COPY	Escribe en la Impresora, si esta está conectada, las 22 líneas superiores de la pantalla.
DATA a,b,c,....	Lista de datos en un programa, que pueden ser leídos con READ
DEF FN f(a,b,...)=e	Define una función, que puede ser activada en el programa con FN
DELETE f	Válido sólo para funcionamiento con Microdrive (ver abajo)
DIM a(m,n,...,s)	Borra un campo a, si este existe, y reserva espacio de memoria para un campo numérico a con s-m dimensiones. Todos los valores iniciales se colocan a 0.

DIM a\$(m,n,...,s)	Como DIM a(m,n,...,s), pero para un campo de Strings. Todos los valores han sido colocados a " "(CHR\$(32))al principio.
DRAW x,y	Dibuja una recta desde la posición PLOT actual a,b hasta la posición a+x, b+y
DRAW x,y,z	Dibuja un sector de círculo, que une la posición PLOT actual a,b con el punto a+x, b+y. Para ello se toma un ángulo inicial de z radianes
ERASE	Válido sólo para funcionamiento con Microdrive (ver abajo)
FLASH n	Inicia un parpadeo de los caracteres si n=1, lo cesa si n=0. Para n=8 sigue el estado anterior.
FOR a=x TO y	Borra el valor de la variable a y coloca a como variable de control para un bucle. a toma primeramente el valor INT x y transcurre el bucle, cuyo final debe estar marcado por NEXT a.Después se aumenta en 1 el valor de a, y el bucle es recorrido de nuevo.Este proceso se repite hasta que a>y.
FOR a=x TO y STEP z	Como FOR a=x TO y, pero aquí la variable de control se aumenta en a+z después de recorrer cada vez el bucle. z puede ser negativa (si x>y es válido).

FORMAT f	Válido sólo para funcionamiento con Microdrive (ver abajo)
GOSUB n	Origina un salto hacia un subprograma, el cual comienza en la línea n. El subprograma debe finalizar con RETURN.
GO TO n	Origina un salto hacia una línea con el número n. Si esta línea no existe, se salta hacia el número de línea superior más próximo.
IF x THEN y	Si x, que puede ser una expresión, es cierto, o sea, que no es 0, el comando y es efectuado.
INK n	Define el color con el que se han de escribir los caracteres. Para n entre 0 y 7 se define el color correspondiente., n=8 significa transparente, n=9 significa contraste.
INPUT..	El programa se interrumpe y espera una entrada, que hay que terminar con ENTER.
INVERSE n	Intercambia el color de fondo con el de primer plano si n=1. Con n=0 se consigue el modo normal. No se permiten otros valores para n.
LET v = e	Asigna a la variable v el valor e, o el valor de la expresión e.
LIST	Visualiza el listado del programa sobre la pantalla.

LIST n	Visualiza sobre la pantalla el listado del programa desde la línea de programa cuyo número sea como mínimo n.
LLIST	Imprime el listado del programa en la impresora
LLIST n	Imprime, a través de la impresora, el listado del programa desde la línea de programa cuyo número sea como mínimo n.
LOAD f	Carga un programa con todas las variables
LOAD f DATA ()	Carga un campo numérico
LOAD f DATA \$()	Carga un campo de cadenas. (Alfanumérico).
LOAD f CODE m,n	Carga n Bytes en la dirección de memoria m hasta m+n-1
LOAD f CODE m	Carga Bytes comenzando en la dirección de memoria m
LOAD f CODE	Carga Bytes en las direcciones de memoria de las que fueron salvadas(SAVE)
LOAD f SCREEN\$	Idéntico a LOAD f CODE 16384,6912. Carga 6912 Bytes en la memoria de pantalla.
LPRINT e	Impresión de la magnitud o de la expresión e sobre la impresora.
MERGE f	Como LOAD f, pero aquí no se borran las líneas de programa existentes en el Spectrum, de no ser que sean idénticas a las cargadas.

MOVE a,b	Válido sólo para funcionamiento con Microdrive (ver abajo)
NEW	Borra el programa BASIC con todas las variables. No influye en los UDG, o el espacio de memoria asegurado con direcciones >n, después de CLEAR n.
NEXT x	Final de un bucle iniciado con FOR
OPEN E	Válido sólo para funcionamiento con Microdrive (ver abajo)
OUT m,n	Da salida al "PORT" a nivel de procesador.
OVER n	Para n=1 se escribe un carácter sobre la pantalla, sin que sea borrado el carácter anterior. Con n=0 se consigue el estado anterior, esto es, se efectúa el borrado.
PAPER n	Como INK, pero aquí se define el color de fondo.
PAUSE n	Interrumpe el programa y visualiza el contenido de la memoria de pantalla durante n segundos, donde $n = 1/50$. Si una tecla es pulsada, la PAUSE es interrumpida. Si n=0, la pausa dura hasta que se pulse una tecla.
PLOT m,n	Sitúa un punto gráfico en las coordenadas m,n. En ello vale $0 \leq m \leq 255$ y $0 \leq n \leq 175$.

POKE m,n	Escribe el valor n en la posición de memoria m.
PRINT e	Visualiza el valor o la expresión e sobre la pantalla, en la próxima posición PRINT.
PRINT AT x,y;e	Visualiza e en la línea x y en la columna y. En ello vale $0 \leq x \leq 21$ y $0 \leq y \leq 31$.
PRINT TAB x;e	Visualiza e en la columna x de la próxima línea disponible
RAND(omize) n	Coloca el valor inicial para el generador aleatorio en n, si $n < 0$. Si $n = 0$, o n no se escribe, se elige para n la cantidad de cuadros visualizados hasta este momento en la pantalla.
READ a,b,c...	Lee los valores de un comando DATA y asigna estos a las variables a,b etc.
REM	Abreviación de Remark (comentario). Todos los caracteres en una línea de programa se ignoran a partir de REM.
RESTORE n	Sitúa el indicador para un comando READ sobre la línea n. Si n no se escribe, significa RESTORE 0.
RETURN	Es el comando al final de un subprograma. Después de este comando, el programa salta a la línea inmediata después del correspondiente GO SUB.

RUN n	Borra todas las variables e inicia el programa en la línea n. Si n no se escribe, el programa se inicia en la primera línea.
SAVE f	Graba el programa con todas las variables
SAVE f LINE n	Graba el programa con todas las variables. Si es cargado un programa grabado de esta forma, se efectúa un salto automático hacia la línea n.
SAVE f DATA ()	Graba un campo numérico
SAVE f DATA \$()	Graba un campo de cadena alfanumérico.
SAVE f CODE m,n	Graba m Bytes de la dirección m
SAVE f SCREEN\$	Graba la memoria de pantalla. Este comando es idéntico con SAVE f CODE 16384,6912
STOP	Interrumpe el programa con el informe 9. Si después se introduce CONT, el programa continúa con el comando siguiente al STOP.
VERIFY	Como LOAD, pero aquí los datos no se cargan en la memoria, sino que son comparados con el contenido de la misma.

Además de los comandos, el Spectrum entiende una serie de funciones, que están listadas a continuación:

ABS x	Valor absoluto de x
ACS x	Arco coseno de x en radianes. Tiene que valer $-1 \leq x \leq 1$
AND	Unión lógica. x AND y es cierta, si $x = y$.
ASN x	Arco seno de x en radianes. Tiene que valer $-1 \leq x \leq 1$
ATN x	Arco tangente de x en radianes.
ATTR (x,y)	Número que indica en forma binaria los atributos de la posición de la pantalla con el número de línea x y de columna y. Bit 7, el más significativo, es 1 para Flash 1 y 0 para Flash 0. Bit 6 es para Bright, o para normal, Bit 5 hasta 3 indican el color de fondo y Bit 2 hasta 0 el color de primer plano.
BIN	Si se quiere representar un número en forma binaria, se puede efectuar introduciendo BIN seguido de ceros y unos.
CHR\$	Resulta el carácter cuyo código es INT x
CODE x\$	Indica el código del primer carácter de la cadena x\$. Si se trata de una cadena vacía, el resultado es 0
COS x	Coseno x
EXP x	Función exponencial de x, o sea e elevada a x, donde e es la base de los logaritmos naturales o neperianos. $e=2,718282$

FN e(x)	Con esto se activa una función definida por DEF FN. El argumento x siempre ha de estar entre paréntesis. Si no existe ningún argumento, hay que escribir los paréntesis de todas formas.
IN x	Indica el valor de la lectura de datos a nivel de procesador desde el "PORT" x.
INKEY\$	Comprueba si ha sido pulsada alguna tecla. El resultado es el carácter representado por la tecla. Si no se ha pulsado ninguna tecla, el resultado es "x", o sea una cadena vacía.
INT x	Indica la parte entera del número x. $x - INT\ x$ Indica la parte de x detrás de la coma.
LEN x\$	Indica la longitud de la cadena x\$.
LN x	Resulta el logaritmo natural del número x
NOT x	Unión lógica. La expresión es 0, cuando $x \neq 0$ y 1, si vale $x=0$
OR	Unión lógica. $x\ OR\ y$ significa que como mínimo x ó y deben ser ciertas.
PEEK x	Indica el valor del Byte en la dirección INT(x)
PI	Representa la constante 3.14159265....
POINT (x,y)	Resulta 1, si para el punto x,y se fijó un color de primer plano, y 0, si este punto representa un color de fondo. Tiene que valer $0 \leq x \leq 255$ y $0 \leq y \leq 175$.

RND Indica un número pseudoaleatorio que resulta mayor o igual a cero y estrictamente menor que uno.

Un número (pseudo) aleatorio entre a y b es producido por el comando $a + (b-a)*RND$ (nunca será igual a b).

SCREEN\$ Resulta el carácter situado en la línea x y columna y. Si el carácter no puede ser reconocido, se edita una cadena vacía.

SGN x Resulta el signo de la variable x. El resultado es -1 si $x < 0$, 0 si $x = 0$, y 1 si $x > 0$.

SQR x Resulta la raíz cuadrada de x

STR\$ x El número es transformado en una cadena.

TAN x Tangente de x en radianes

USR x Activa una rutina escrita en código máquina, cuya dirección inicial está en x

USR x\$ Indica la dirección para la configuración de Bits de un carácter propio definido.

VAL x\$ Indica el valor numérico de la cadena x\$

VAL\$ x\$ Indica el valor numérico de la cadena x\$ nuevamente como cadena

- x Cambia de signo la variable x

El Spectrum puede realizar las siguientes operaciones binarias:

+ Adición de números o concatenación de cadenas.

- Sustracción

$*$	Multiplicación
$/$	División
$^$	Potenciación
$=$	Igualdad
$>$	Mayor que
$<$	Menor que
$>=$	Mayor o igual
$<=$	Menor o igual
$<>$	Desigual

Con el Interface 1, el juego de comandos del Spectrum se amplía. Existen ahora 15 Streams (o corrientes), señalizadas con los números 1 hasta 15. Además existen siete formas de canales (Channels), que se denominan: "k" para teclado, "s" para pantalla, "p" para impresora, "t" para interface RS232 de texto, "b" para el interface RS 232 binario, "n" para la red y "m" para los Microdrives. Son posibles los siguientes comandos:

CAT x	Indica el contenido en el cassette del Microdrive x. La lista se edita alfabéticamente. Además se indica el nombre de la cassette y el espacio de memoria libre en Kbytes.
CAT #z;x	Indica el contenido de la cassette del Microdrive x a través de la corriente z
CLOSE # x	Desconecta todos los canales de la corriente x. Si aún quedan datos en la memoria intermedia (buffer), estos se transmiten (en la Red) o se registran (en el Microdrive).
ERASE "m"; x; f	Borra el fichero con nombre f del cassette del Microdrive x
FORMAT "m" ; x; f	Formatea el cassette del Microdrive x, o sea que lo prepara para su empleo en BASIC. En ello se asigna el nombre f a la cassette, que se edita junto con el contenido. Si la cassette estaba escrita se borran todos los datos.
FORMAT "n";x	Asigna al Spectrum el número x en la red.

FORMAT "t"; x	Fija la velocidad de transmisión de datos a través de la conexión RS232 en Baudios, x. x puede ser un valor de 50, 110, 300, 600, 1200, 2400, 4800, 9600 ó 19.200
FORMAT "b";x	Como FORMAT "t";x. Para el canal t- y b- la velocidad de transmisión (Baud) es siempre igual
INKEY#x	Indica el carácter que viene de la corriente x. Si no existe ningún carácter, se devuelve la cadena vacía, o sea "".
INPUT #x;y	Una magnitud y es introducida a través de la corriente. La opción LINE también es posible aquí.
LOAD*....	Carga un programa, datos o Bytes del canal especificado. Sólo están permitidos los canales "b", "n" y "m". Todas las opciones que se conocen del LOAD normal, están permitidas.
MERGE*...	Como el MERGE ordinario, sólo desde el canal especificado
MOVE a TO b	Transmite datos desde la fuente a hasta el destino b. No es necesario abrir un canal antes, o cerrarlo después de este comando.
OPEN #x,y	El canal y es enlazado con la corriente x. Así es posible la entrada y salida en BASIC a través de este canal. Corrientes permitidas son "k", "s" y "p"
PRINT #x..... ante-	Ejecuta una salida hacia la corriente especificada. Este debe ser abierto anteriormente como corriente de salida.

SAVE*...

Almacena un programa, datos o Bytes para canal especificado. Sólo los canales "b" , "n" y "b" pueden ser utilizados. Todas las opciones del Save 'normal' están permitidas.

VERIFY*...

Como el VERIFY 'normal', sólo que los datos vienen del canal especificado

APENDICE 2: INFORMES

Cada vez que el Spectrum ha llevado a cabo un comando o un programa, surge un informe en la parte inferior de la pantalla. Este informe consiste en un código y una explicación corta, la cual facilita la búsqueda del error.

Por regla general, el Informe más deseado es '0 OK, L(línea): C(comando)'. Esto quiere decir que la operación ha sido realizada de la forma deseada y que no ha surgido ningún error formal. Pero desgraciadamente surgen con frecuencia otras indicaciones. La lista siguiente contiene todas las indicaciones:

- 0 OK - Transcurso con éxito del programa
- 1 NEXT without FOR
En este caso existe el final de un bucle, pero el inicio, o sea el comando FOR, ha sido olvidado.
- 2 Variable not found
En conjunto con una variable simple, este error surge cuando la variable ha sido utilizada, antes de ser definida con LET, READ o INPUT. Cuando se trata de un campo, este error surge cuando el comando DIM ha sido olvidado.
- 3 Subscript wrong
Una variable de campo tiene un subíndice erróneo respecto al que fue determinado con el comando LET o DIM.
- 4 Out of memory
La capacidad de memoria del Spectrum no es suficiente.
- 5 Out of screen
Con un comando INPUT se produjeron más de 23 líneas

en la parte inferior de la pantalla, o un comando PRINT exige un número de línea más alto que 21.

- 6 Number too big
En un cálculo surge un número mayor que 10 elevado a la 38a. potencia. Casi siempre sucede esto al dividir por un número, que tiene asignado el valor 0.
- 7 RETURN without GO SUB
El programa se encuentra con un RETURN sin haber recibido antes un comando GO SUB.
- 8 End of file
El final de un fichero ha sido encontrado. Esto sólo tiene importancia para usuarios de Microdrives.
- 9 STOP Statement
El programa ha sido interrumpido por STOP. Si se introduce CONTINUE, el programa sigue un comando después del STOP.
- A Invalid argument
El argumento de una función matemática no es válido
- B Integer out of range
Un número entero ha resultado demasiado grande
- C Nonsense in BASIC
Un comentario no muy agradable para programadores de BASIC. La expresión calculada por VAL no es válido
- D BREAK - CONT repeats
El programa fue interrumpido pulsando la tecla BREAK. Si se introduce CONT se repite el comando interrumpido, a diferencia del caso de la indicación L (ver más abajo)

- E Out of DATA
Se pretenden leer más magnitudes con un comando READ que fueron introducidos con DATA. A veces sólo se ha suprimido un RETURN.
- F Invalid file names
Se ha intentado grabar un fichero sin nombre o con un nombre de más de 10 caracteres.
- G No room for line
No hay capacidad en memoria para la siguiente línea de programa.
- H STOP in INPUT
Un comando INPUT fue contestado con la tecla STOP. CONTINUE repite el comando INPUT.
- I FOR without NEXT
El principio de un bucle se efectúa a través de FOR, pero el Spectrum dice que falta el final con NEXT.
- J Invalid I/O device
Dispositivo de entrada o salida no válido para un Microdrive
- K Invalid colour
El número introducido para un color no es válido
- L BREAK into program
Entre la ejecución de dos líneas, el programa ha sido interrumpido pulsando la tecla BREAK. En este caso, introduciendo CONTINUE, la siguiente orden es ejecutada.
- M RAMTOP no good

El número indicado para RAMTOP es o muy grande, o muy pequeño.

- N Statement lost
Se indica un salto hacia un comando, que ya no existe.
- O Invalid streams
Al utilizar Microdrives ó el Interface 1, se ha introducido un comando de corriente no válido.
- P FN without DEF
Ha sido utilizada una función, sin ser definida antes por DEF FN
- Q Parameter error
En una función definida por DEF FN, se ha introducido una cantidad equivocada o un tipo erróneo de parámetros.
- R Tape loading error
Un error de lectura al cargar o verificar de un cassette.

Si Vd. ha conectado un Interface 1 al Spectrum, éste puede producir 21 informes de error más:

- Code error
Se ha intentado cargar con un comando LOAD un bloque de código más grande que el área de destino especificada por su sentencia LOAD
- Drive 'write' protected
El intento de escribir en un cassette protegido ha fallado (sin la lengüeta de plástico)
- File not found
El fichero que se ha de cargar con LOAD no existe, o no puede ser leído completamente. El último caso surge, cuando la cassette está deteriorada, o cuando un fichero no se ha cerrado debidamente.
- Invalid device expression
Una unidad ha sido especificada con una letra diferente a las permitidas: s, p, k, m, n, t o b. Este error puede surgir cuando al especificar un canal s, p, o k, se ha escrito un punto y coma en lugar de una coma.
- Invalid name
A un fichero no se le ha asignado ningún nombre, o el nombre tiene más de 10 caracteres.
- Invalid station number
A una estación de red se le ha asignado un número menor que 0, o mayor que 64, o con el comando FORMAT se ha utilizado un número menor que 1 o mayor que 64.
- Invalid stream number
Se ha asignado un número menor que 0 o mayor que 15 a una corriente.

- Merge error
Se ha intentado mezclar datos y códigos. MERGE sólo está permitido para programas, y aquí sólo para los que no fueron memorizados con un comando LINE.
- Microdrive full
No hay más espacio en el cartucho de Microdrive para ejecutar el comando SAVE. Un fichero posiblemente abierto, debería ser borrado del cassette.
- Microdrive not present
Se ha activado un Microdrive que contiene un cassette sin formatear, que no contiene ningún cassette, o no está conectado
- Missing baud rate
Se ha olvidado de fijar la velocidad de transmisión en baudios con el comando FORMAT
- Missing drive number
Al intentar acceder a un Microdrive, se ha olvidado de introducir el número de éste
- Missing name
El nombre de fichero no ha sido indicado
- Missing station number
El número de estación no ha sido indicado
- Program finished
Se ha intentado ejecutar una línea de programa que es mayor que la última existente. Este informe también surge cuando se introduce RUN y no hay ningún programa en la memoria.

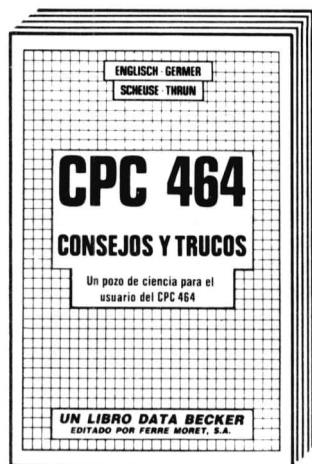
- Reading a 'write file'
Se ha intentado leer datos de un fichero que no existe o de uno que se abrió solamente para escribir.
- Stream already open
Se ha intentado abrir una corriente, que ya está asignada a un nuevo canal (m,n,t ó b). En este caso en primer lugar hay que cerrar la corriente.
- Verification has failed
Se ha detectado una diferencia entre un fichero memorizado y el correspondiente en la memoria durante la verificación.
- Writing to a 'read' file
Se ha intentado enviar datos nuevamente a un fichero ya existente, sin tratar de borrar el fichero antiguo.
- Wrong file type
El tipo de fichero no es adecuado para la operación deseada. Esto surge por ejemplo, cuando el Código se ha intentado como programa, o un programa se ha de tratar con MOVE.



MSX_EL MANUAL ESCOLAR

P.V.P. 2.800,- ptas.

Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje, descritos de una forma muy completa y fácil de comprender. Teorema de Pitágoras, progresiones geométricas, escritura cifrada, crecimiento exponencial, verbos irregulares, igualdades cuadráticas, movimiento pendular, estructura de moléculas, cálculo de interés y muchas cosas más.



CPC-464 CONSEJOS Y TRUCOS

P.V.P. 2.200,- ptas.

Ofrece una colección muy interesante de sugerencias, ideas y soluciones para la programación y utilización de su CPC-464: Desde la estructura del hardware, sistema de funcionamiento - Tokens Basic, dibujos con el joystick, aplicaciones de ventanas en pantalla y otros muchos interesantes programas como el procesamiento de datos, editor de sonidos, generador de caracteres, monitor de código máquina hasta listados de interesantes juegos.



MSX GRAFICOS Y SONIDOS, 250 pág.

P.V.P. 2.800,- ptas.

Las computadoras MSX no sólo ofrecen una relación precio/rendimiento sobresaliente, sino que también poseen unas cualidades gráficas y de sonido excepcionales. Este libro expone las posibilidades de los MSX de forma completa y fácil. El texto se completa con numerosos y útiles programas ejemplo.



ROBOTICA PARA SU COMMODORE 64, 230 pág.
P.V.P. 2.800,- ptas.

En el libro de los robots se muestran las asombrosas posibilidades que ofrece el CBM 64, para el control y la programación, presentadas con numerosas ilustraciones e intuitivos ejemplos. El punto principal: Cómo puede construirse uno mismo un robot sin grandes gastos. Además, un resumen del desarrollo histórico del robot y una amplia introducción a los fundamentos cibernéticos.

Gobierno del motor, el modelo de simulación, interruptor de pantalla, el Port-Usuario cómodo del modelo de simulación, Sensor de infrarrojos, concepto básico de un robot, realimentación unidad cibernética, Brazo prensor, Oír y ver.



TODO SOBRE EL FLOPPY 1541, 482 pág.
P.V.P. 3.200,- ptas.

La obra Standard del floppy 1541, todo sobre la programación en disquettes desde los principiantes a los profesionales, además de las informaciones fundamentales para el DOS, los comandos de sistema y mensajes de error, hay varios capítulos para la administración práctica de ficheros con el FLOPPY, amplio y documentado Listado del Dos. Además un filón de los más diversos programas y rutinas auxiliares, que hacen del libro una lectura obligada para los usuarios del Floppy.



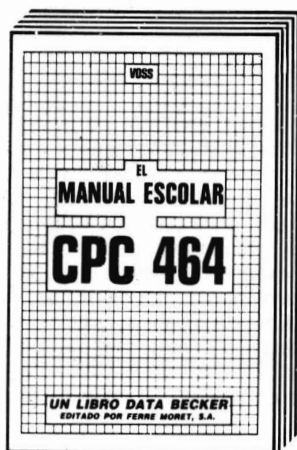
MANTENIMIENTO Y REPARACION DEL FLOPPY 1541,
200 pág.
P.V.P. 2.800,- ptas.

Saberse apañar uno mismo, ahorra tiempo, molestias y dinero, precisamente problemas como el ajuste del floppy o reparaciones de la platina se pueden arreglar a menudo con medios sencillos. Instrucciones para eliminar la mayoría de perturbaciones, listas de piezas de recambio y una introducción a la mecánica y a la electrónica de la unidad de disco, hay también indicaciones exactas sobre herramientas y material de trabajo. Este libro hay que considerarlo en todos sus aspectos como efectivo y barato.



ZX SPECTRUM CONSEJOS Y TRUCOS, 211 pág.
P.V.P. 2.200,- ptas.

Una interesante colección de sugestivas ideas y soluciones para la programación y utilización de su ZX ESPECTRUM. Aparte de muchos peeks, pokes y USRs hay también capítulos completos para, entre otros, entrada de datos asegurado sin bloqueo de ordenador, posibilidades de conexión y utilización de microdrives y lápices ópticos programas para la representación de diagramas de barra y de tarta, el modo de utilizar óptimamente ROM y RAM.



CPC-464 EL LIBRO DEL COLEGIO
P.V.P. 2.200,- ptas.

Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje, descritos de una forma muy compleja y fácil de comprender. Teorema de Pitágoras, progresiones geométricas, escritura cifrada, crecimiento exponencial, verbos irregulares, igualdades cuadráticas, movimiento pendular, estructura de moléculas, cálculo de interés y muchas cosas más.



ZX SPECTRUM EL MANUAL ESCOLAR
P.V.P. 2.200,- ptas.

Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje, descritos de una forma muy completa y fácil de comprender. Teorema de Pitágoras, progresiones geométricas, escritura cifrada, crecimiento exponencial, verbos irregulares, igualdades cuadráticas, movimiento pendular, estructura de moléculas, cálculo de interés y muchas cosas más.



METODOLOGIA DE LA PROGRAMACION

P.V.P. 2.200,- ptas.

El primer libro recomendado para escuelas de enseñanza de informática y para aquellas personas que quieren aprender la programación. Cubre las especificaciones del Ministerio de Educación y Ciencia para Estudios de Informática. Realizado por un alto mando del ejército Español, un Dr. Ingeniero y Diplomado en Informática y profesor de la UNED y por un oficial técnico especialista en informática de gestión. Utilizado en todos los institutos politécnicos del ejército español. Es un seguro candidato a ediciones en lengua inglesa, alemana y francesa. Es el primer libro que introduce a la lógica del ordenador. Es un elemento de base que sirve como introducción para la programación en cualquier otro lenguaje. No se requieren conocimientos de programación ni siquiera de informática. Abarca desde los métodos de programación clásicos a los más modernos.



MANUAL ESCOLAR PARA SU COMMODORE 64, 351 pág.

P.V.P. 2.800,- ptas.

Este libro, escrito especialmente para escolares de grado medio y superior, contiene muchos interesantes programas de aprendizaje para solucionar problemas, descritos detalladamente y de manera fácilmente comprensible. Facilitan un aprendizaje intensivo y ameno, con, entre otros, los siguientes temas: Teorema de pitágoras, progresiones geométricas, palanca mecánica, crecimiento exponencial, verbos irregulares, ecuaciones de segundo grado, movimientos de péndulo, formación de moléculas, aprendizaje de vocablos, cálculo de interés y su capitalización. Una corta repetición de los elementos BASIC más importantes y una introducción a los rasgos esenciales del análisis de problemas, entre otros, completan el conjunto.



64 EN EL CAMPO DE LA TECNICA Y LA CIENCIA, 296 pág.

P.V.P. 2.800,- ptas.

Ofrece un campo fascinante y amplio de problemáticas científicas. Para esto el libro contiene muchos listados interesantes: Análisis de Fourier y síntesis, análisis de redes, exactitud de cálculo, formateado de números, cálculo del valor PH, sistemas de ecuaciones diferenciales, modelo ladrón presa, cálculo de probabilidad, medición de tiempo, integración, etc.



PEEKs y POKEs, 177 pág.
P.V.P. 1.600,- ptas.

Con importantes comandos PEEK y POKE se pueden hacer también desde el Basic muchas cosas, para las que se necesitarían normalmente complejas rutinas en lenguaje máquina. Este libro explica de manera sencilla el manejo de PEEKs y POKEs. Con una enorme cantidad de POKEs importantes y su posible aplicación. Para ello se explica perfectamente la estructura del Commodore 64: Sistema operativo, interpretador, página cero, apuntadores y stacks, generador de caracteres, registros de sprites, programación de interfaces, desactivación del interrupt. Además una introducción al lenguaje máquina. Muchos programas ejemplo.



EL MANUAL DEL CASSETTE, 190 pág.
P.V.P. 1.600,- ptas.

Un excelente libro, que le mostrará todas las posibilidades que le ofrece su grabadora de cassettes. Describe detalladamente, y de forma comprensible, todo sobre el Datassette y la grabación en cassette. Con verdaderos programas fuera de serie: Autostart, Catálogo (¡busca y carga automáticamente!), backup de y a disco, SAVE de áreas de memoria, y lo más sorprendente: un nuevo sistema operativo de cassette con el 10-20 veces más rápido FastTape. Además otras indicaciones y programas de utilidad (ajuste de cabezales, altavoz de control).



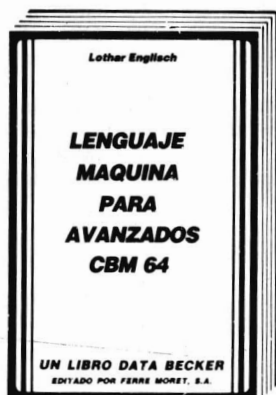
MSX PROGRAMAS Y UTILIDADES, 1985, 194 pág.
P.V.P. 2.200,- ptas.

El libro contiene una amplia colección de importantes programas que abarcan, desde un desensamblador hasta un programa de clasificaciones deportivas. Juegos superemocionantes y aplicaciones completas. Los programas muestran además importantes consejos y trucos para la programación. Estos programas funcionan en todos los ordenadores MSX, así como en el SPECTROVIDEO 318 328. ETRACTO DEL CONTENIDO: Volcado memoria hexadecimal. Editor gráficos. Editor de sonido. Escritura de ordenador. Lista referencia de variables. Calendario. Desensamblador. ADMINISTRACION de una colección de discos L.P. HOLLOW - JUEGO DE LAS CEREZAS. DIAGRAMAS DE BARRAS. TABLAS DEPORTIVAS.



LENGUAJE MAQUINA PARA
COMMODORE 64, 1984, 201 pág.
P.V.P. 2.200,- ptas.

¡Por fin una introducción al código máquina fácilmente comprensible! Estructura y funcionamiento del procesador 6510, introducción y ejecución de programas en lenguaje máquina, manejo del ensamblador, y un atractivo muy especial: ¡un simulador de paso a paso escrito en BASIC!



LENGUAJE MAQUINA PARA AVANZADOS
CBM 64, 1984, 206 pág.
P.V.P. 2.200 ptas.

¿Ud. ha logrado iniciarse en código máquina? Entonces el «nuevo English» le enseñará cómo convertirse en un profesional. Naturalmente con muchos programas ejemplo, rutinas completas en código máquina e importantes consejos y trucos para la programación en lenguaje máquina y para el trabajo con el sistema operativo.

EL CONTENIDO:

ZX-SPECTRUM CONSEJOS Y TRUCOS contiene una interesantísima colección de sugestivas ideas y soluciones para la programación y utilización de su ZX-SPECTRUM.

Extracto del contenido:

- 64 Caracteres por línea
- Entrada de datos asegurada sin bloqueo del ordenador
- Mastermind
- Representación gráfica de un mapamundi
- Posibilidades de conexión y utilización de microdrives y lápices ópticos
- Programa para representación de diagramas de barra y de tarta
- Control de stocks y ventas
- Fichero clientes
- Programa para obtener un display publicitario
- y, por supuesto, con multitud de PEEKs, POKes, USRs, y el modo de utilizar óptimamente ROM y RAM

ESTE LIBRO HA SIDO ESCRITO POR:

Dr. Peter Bosetti, Físico de la Universidad de Aachen. La falta de agilidad en la realización de programas en los grandes ordenadores le llevaron al ZX Spectrum, el cual le fascinó de tal modo, que decidió hacer públicos mediante este libro sus conocimientos, ideas y trucos.

ISBN 84-86437-10-5

