

MICROHOBBY

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO II N.º 22

160 Ptas.

(incluido I.V.A.)

Canarias, Ceuta y Melilla **165** ptas.

**MATRICES DE
DOS DIMENSIONES:
RESOLVER PROBLEMAS
DE LA VIDA REAL.**

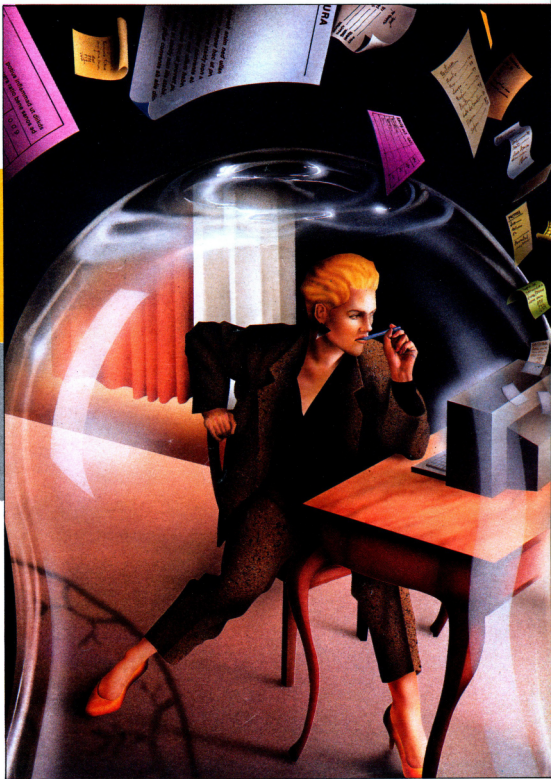
**LA
CONTABILIDAD
DOMESTICA
RESUELTA
CON
SUPERCONT**

**COMO
CONSEGUIR
CARACTERES
GIGANTES
EN 3D**

**ROTACION Y
DESPLAZAMIENTO
DE BITS.**

SOFTWARE

**Sabre Wulf,
el héroe de la jungla,
en busca del
Talismán Sagrado.**

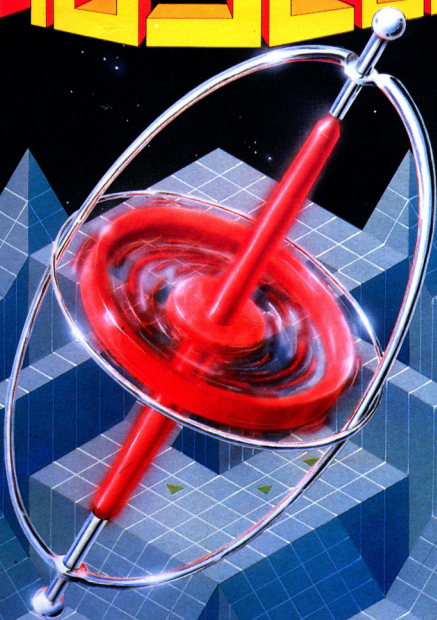


HOBBY PRESS, S.A.

SI BUSCAS LO MEJOR **ERBE** Software LO TIENE

LA REPLICA AL CELEBRE "ROLLING" DE LAS MAQUINAS.
EL JUEGO MAS ADICTIVO QUE PUEDES ENCONTRAR

GYROSCOPE



DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA. ERBE SOFTWARE, SANTA ENGRACIA, 17. Tel: 447 34 10. DELEGACION BARCELONA, Avd. MISTRAL, 10. Tel. (93) 432 07 31

Steiner

AMSTRAD

Sumario

Año II • Número 22 • 28 de Enero al 4 de Febrero de 1985
160 ptas. (incluido I.V.A.)
Precio Canarias, Ceuta y Melilla 155 + 10 ptas. sobretasa aérea.

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Victor Prieto

Subdirector
José María Díaz

Redactora Jefe
Marta García

Diseño
José Flores

Colaboradores
Francisco Portalo, Pedro Sudán
Miguel Sepúlveda,
Francisco Martín,
Jesús Alonso, Pedro S. Pérez
Amalio Gómez
Juan J. Martínez,
David Sopena, Alberto Suárez,
Eduardo R. Velasco

Secretaría Redacción
Carmen Santamaría

Fotografía
Carlos Candel
Javier Martínez

Portada
M. Barco

Ilustradores
Javier Igual, J. Pons, F. L. Frontán, J. Septien, Pejo, J. J. Mora, Luigi Pérez

Edita

HOBBY PRESS S.A.

Presidente

María Andrino

Consejero Delegado

José I. Gómez-Centurión

Jefe de Publicidad

Cancha Gutiérrez

Publicidad Barcelona

José Galán Cortes

Tel.: (93) 303 10 22/313 71 62

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

La Granja, 39

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Carlos Peropadre

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Impreme

ROTEDEC, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Salana, 16

Déposito Legal:

M-28468-1985

Derechos exclusivos

de la revista

COMPUTING with**the AMSTRAD**

Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud America 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace necesariamente solidario de las opiniones vertidas por sus colaboradores en los artículos firmados. Reservados todos los derechos.

Se solicitará control OJD

5 Primera plana

Amstradrobótica. Nuevos compatibles IBM AT.

6 Primeros pasos

Los problemas de la vida real a los que podemos aplicarles un tratamiento informático son, por desgracia, bastante complejos. Muchos de ellos requieren una estructura organizada de datos, es decir, una matriz, pero no basta siempre con una matriz de una sola dimensión. El Basic permite una aproximación más potente, al dotar a las matrices de todas las dimensiones que queramos. Veremos en Primeros pasos como se utilizan.



Serie Orc 10

Para aquellos que estén hartos de llevar las múltiples cuentas de gastos e ingresos de su hogar a mano, Supercont les viene que ni pintado. Tecleen y verán...

15 Amstravagancia

Los discos necesitan maternales cuidados para que permanezcan sanos y salvos, con toda su información incólume. En interés de todos los usuarios, **AMSTRAD Semanal** revela el ABC del arte del cuidado de los discos.



Mr. Joystick 18

Sabre Wolf, un clásico de Ultimate al alcance de todos los foros de los juegos de aventuras que tengan un Amstrad.

20 ProgramAcción

Easydraw, publicado en el número 1 de nuestra revista, es un gran programa, pero sin duda admite mejoras. El programa magnificará de caracteres es una, y muy importante. Pronto se convertirá en una herramienta esencial para tus diseños gráficos.

Análisis 25

Después de haber hablado de algoritmos de ordenación, le toca el turno a los métodos de búsqueda de datos. Uno de ellos, la búsqueda binaria, es muy sencillo y rápido, y puede aplicarse tanto a números como a cadenas.

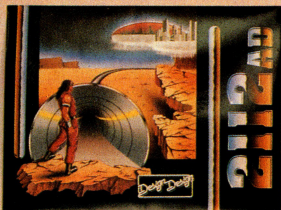


26 Código máquina

Continuamos analizando el grupo de instrucciones de rotación y desplazamiento, uno de los «sets» de órdenes más útiles del Z80.

AmstradIdeas 30

Una excelente idea que muestra el código ASCII de la tecla pulsada en la pantalla de forma rápida y elegante.



2112: Nuevo. De Design Desgin. Con este juego debes usar toda tu inteligencia para vencer las dificultades que te ofrece. Es increíble la velocidad con que los gráficos van cambiando en la pantalla.
Spectrum 48 ó 128 K. P.V.P. **1.800**



Forbidden Planet. De Design Desgin. *Nuevo.* Este juego fabuloso es tan misterioso que debes descubrirlo por ti mismo. ¡Vive las aventuras!
Para Spectrum 48 y 128 K. P.V.P. **1.800**

Wizard Lair: de Bubble Bus. Te encuentras atrapado en una cueva muy profunda peligrando tu vida. Tienes que buscar la salida venciendo muchos obstáculos.
Spectrum 48 y 128 K.
Commodore 64 y 128 K. Amstrad CPC. P.V.P. **1.900**



Starquake: de Bubble Bus. *Nuevo Hit.* Tu misión es encontrar el camino de Starquake y desactivar los núcleos de los planetas antes de que el mundo estalle. ¡Más de 400 pantallas! Para Spectrum 48, 128 K. Próximamente Amstrad CPC. P.V.P. **2.100**



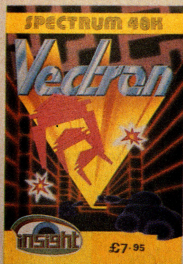
Hi Rise: de Bubble Bus. *Nuevo.* Tú eres el jefe de una empresa de construcción. Tus trabajadores no están contentos contigo. Tienes que escarpate para evitar enfrentamientos.
Para Amstrad CPC. P.V.P. **2.100**



Instrucciones en castellano. Importados de Inglaterra. Todos los precios incluido IVA. Tiendas y distribuidores. Tel. (965) 26 35 93. Pedidos contra-reembolso. Tel. (965) 26 35 93. Disponible directamente por British Soft o en los mejores establecimientos de software.

British Soft. Rocafel, 19. Albufereta (Alicante)
Distribuidor en Madrid:

Computique
Embajadores, 90
28012 Madrid
Tel. (91) 227 09 80



Vectron: de Insight. *Nuevo Hit* en Inglaterra. Vas a ser transportado a un extraño ordenador que genera el universo. Tienes que luchar contra el robot. Rom, tanques y Randomiser. Para Spectrum 48 y 128 K. P.V.P. **1.800**

Jep Set Willy II: de Software Projects.

Ahora una segunda versión del famoso héroe Willy. Más de 130 pantallas. Otro hit de la misma casa que Manic Miner. Para Spectrum 48 y 128 K. Commodore 64 y 128 K. Amstrad CPC y MSX. P.V.P. **1.800**



EL AMSTRAD PUEDE CONTROLAR ROBOTS

F

ischertechnik

ha desarrollado un interface específico para los diferentes tipos de ordenadores Amstrad que los convierte en una unidad de control de robots.

Luego está el propio equipo de construcción, que contiene los componentes para construir cualquiera de los siguientes:

— Elevador de cargas: eleva tres alturas diferentes, con tres funciones y comandos.

— Herramienta mecánica: combina mediante el trabajo dos procesos diferentes.

— Plotter: compila gráficos mediante el ordenador.

— Torre de Hanoi: un robot de ejes de giro coloca las fichas para jugar a la Torre de Hanoi.

— Panel gráfico: unidad de entrada de gráficos para dibujar en la pantalla.

— Robot aprendiz: aprende secuencias de movimientos y los repite automáticamente.

La última parte ofrecida por Fischertechnik es el software. Los formatos con los que se proporciona son los siguientes: Cassette o Diskette para Amstrad CPC 464-664-612.

Por último su precio, incluyendo la caja de construcción, el software, el interface y el adaptador para conectar a la red eléctrica, es de 34.900 ptas. El distribuidor para España es Master Computer, Centro Comercial, Ciudad Santo Domingo, Circa. Burgos, km 28, Algete (Madrid).



COMPATIBLE AT DE HEWLETT PACKARD

Hewlett Packard ha contribuido con su grano de arena a intentar arrebatarle a IBM la parte del león del mercado del ordenador personal, monopolizado por la multinacional americana.

El grano de arena se llama HP Vectra, y es una maravilla: las especificaciones son las habituales; más memoria que el IBM AT, el mismo microprocesador (Intel 80286) y sistema operativo (MS-DOS 3.X y/o XENIX), aproximado un 30 por 100 más veloz que el AT y entre un 20 por 100 y un 60 por 100 más barato (esto último según Hewlett Packard, claro).

IBM DESCOLONIZA ZAMBIA

En plan noticia breve, parece ser que IBM se ha visto obligada a abandonar a Zambia, uno de sus mayores mercados en África (sí, sí, como lo oyen), debido a la política de ventas extremadamente agresiva de sus principales competidores en el sector: ICL e HITACHI.

Primera plana

Problemas en Control Data

Se prevén unas pérdidas de Control Data para este año de unos 50 millones de dólares. Como medida de urgencia, la empresa aprobó dar a sus empleados 4 días de vacaciones sin sueldo, que además son obligatorias.

Además, 1.500 puestos de trabajo en Magnetic Peripherals, filial de la empresa, se volatilizan por momentos.

Tiene gracia que alguien dé vacaciones obligatorias. En fin, la crisis.

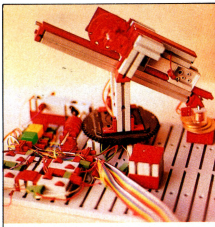
NUEVOS SISTEMAS DE ALMACENAMIENTO 3M

3M, la conocida marca líder en el campo de los sistemas de almacenamiento masivo de datos (discos, con perdón), ha presentado su nueva gama de cartuchos y diskettes, introduciendo importantes novedades en este campo.

Concretamente un disco de 3,5 pulgadas (formato Sony) de un megabyte de capacidad, por supuesto en doble cara, doble densidad. Un disco de 5,25 pulgadas, también de alta densidad, capaz para almacenar 1,6 megabytes.

Ambos dos serán comercializados con la famosa garantía 3M para toda la vida.

En cuanto a los cartuchos, destaca el modelo DC 300XLP. Este nombre, tan fácil de recordar, esconde una pequeña maravilla que puede almacenar 45 megabytes de información.



MATRICES DE DOS DIMENSIONES

En este momento ya somos capaces de saber qué es y para qué sirve una variable con un subíndice entre paréntesis.

Vamos a meternos en más profundidades.

Si añadimos una dimensión —subíndice— a estas variables habremos descubierto una herramienta de gran utilidad en el mundo de la programación: las matrices bidimensionales.



En el anterior artículo de la serie vimos cómo se puede definir y dimensionar una colección de elementos del mismo tipo numérico o alfanumérico.

DIM conjunto (7)

o

DIM grupo\$ (16)

son dos órdenes que dimensionan o guardan en memoria el espacio suficiente para contener 8 números o 17 valores alfanuméricos respectivamente. El número que está entre paréntesis indica el valor máximo que puede tener un elemento cualquiera de estas colecciones.

No se olvide, y tenga muy en cuenta que el elemento con subíndice 0 también existe y que lo podemos utilizar como otro cualquiera.

«Bidimensional» una matriz...

Bien, avancemos. A partir de ahora vamos a utilizar matrices con dos subíndices. La forma de indicarle al ordenador que nos reserve espacio para este tipo de variables es dimensionarlas dándoles un nombre seguido de dos subíndices separados por una coma y colocados entre paréntesis. Fácil, ¿no?

DIM conjunto (3,2)

nos guardaría en la memoria el espacio necesario para almacenar 12 números (4*3),

DIM array\$ (2,6)

definiría un conjunto de 21 elementos alfanuméricos a los que podemos acceder mediante estos dos subíndices.

Podemos asociar estas variables con dos índices a la idea de un espacio de dos dimensiones. Sus elementos se agrupan en tantas filas como nos indique el primer subíndice y en tantas columnas como nos diga el segundo.

DIM grupo\$ (3,2)

sería una matriz de dos dimensiones que tendría 4 filas —3 más la fila 0— y 3 columnas —2 más la columna 0.

Si queremos dirigirnos a un elemento en concreto nos bastaría con hacer:

grupo\$ (2,1)

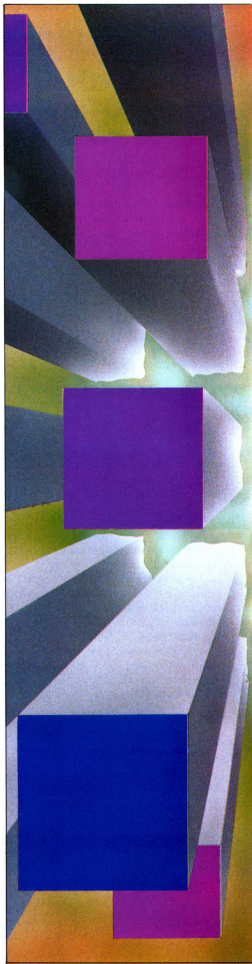
y tendríamos localizado el que está colocado en la fila 2 y columna 1.

Estamos seguros que no tendrá ningún problema en decirnos cómo se pondrá el elemento de este array que está situado en la fila 1, columna 2.

...e inicialarla

La forma de almacenar un valor en un lugar determinado dentro de una matriz bidimensional es semejante a como se hacía para las de una dimensión:

grupo\$ (2,1) = «nombre»



Primeros pasos

colocaría el valor alfanumérico «nombre» en el lugar reservado en memoria para el elemento del array situado en la fila 2 y la columna 1,

conjunto (3,1) = número

almacenaría el valor que contiene la variable «número» en el elemento situado en la fila 3, columna 1 de la matriz dimensionada con el nombre «conjunto».

De la misma forma, si queremos asignar el valor de un elemento determinado a una variable podemos hacerlo así:

número = conjunto (3,1)

y si queremos sacarlo en pantalla, con una sencilla instrucción:

```
PRINT conjunto (3,1)
```

nos sería suficiente.

Vamos a analizar un pequeño programa que nos intentará despejar las posibles dudas que tengamos sobre estos conceptos y que es un resumen de la asociación de estas variables —filas, columnas.

Programa uno

Debe resultarles sencillo de seguir. Está dividido en tres partes:

En la primera nos pide los valores máximo de los subíndices y los almacena en las variables «filas» y «columnas» líneas 30 y 40.

Dimensiona la matriz según estos valores en la línea 50.

La segunda parte —líneas 60 a 140— se encarga de dar el valor que nosotros queremos a cada uno de los elementos de la matriz que anteriormente hemos dimensionado.

Para ello utiliza dos bucles anidados cuyas variables de control son «i» y «j», oscilando desde 1 hasta los valores máximos indicados por las variables «filas» y «columnas».

En la línea 120 asigna el número que nosotros damos por el teclado a cada uno de los elementos.

La tercera parte —líneas 150 a 230— es la que utilizamos para localizar los valores almacenados en la fila y columna que nosotros queremos.

Las líneas 160 y 230 son los límites de un bucle WHILE ... WEND en el que la condición se está cumpliendo constantemente: la variable «condición» vale siempre O ya que nuestro

Amstrad así la inicializa y nosotros no cambiamos su valor en el programa.

Esto quiere decir que nunca saldremos del bucle. La única forma de hacerlo, cuando ya estemos cansados de investigar valores, es pulsando la tecla ESC, produciendo un BREAK en el programa.

En las líneas 190 y 200 metemos en el ordenador los números de la fila y la columna del elemento cuyo valor queremos encontrar. Los almacenamos en las variables «fila» y «columna».

Mediante la línea 210 asignamos a la variable «número» el valor del elemento pedido y en la línea 220 lo imprimimos.

Estamos seguros de que no habrá ninguna dificultad en seguir este programita. Habrá podido observar que los subíndices están en este caso constituidos por variables numéricas y hubieran podido ser también expresiones tan complicadas como nosotros quisiéramos.

Le hacemos ahora una sugerencia: la mejor forma de comprender exactamente la filosofía de este tipo de variables es manejándolas usted mismo. No se limite a leer lo que nosotros le decimos. Intente cambiar los programas que le proponemos. Por ejemplo: ¿Qué ocurriría si uno de los subíndices de un elemento es mayor que los especificados en la instrucción DIM? ¿Qué mensaje nos daría el ordenador? ¿Qué ocurre si...? No lo dude y decidase a realizar sus propias investigaciones.

Póngase en plan de severo profesor que acaba de realizar una examen a sus alumnos. Llega la hora de calificar y ordenar los aprobados y suspensos. Le proponemos que se ejercite escribiendo un programa que nos permita almacenar y buscar después las notas de cada alumno y además contabilizar los aprobados y suspensos por asignatura. Vamos a hacerlo juntos.

Lo primero que necesitamos es conocer el número de alumnos y el de asignaturas para así definir adecua-

damente dos tablas, una que contenga el nombre del alumno —por tanto será una matriz de cadena— y otra que contenga las notas que tiene cada alumno en cada asignatura— será bidimensional y numérica.

¿De acuerdo?

Una vez definidas las matrices llega la hora de ir almacenando los datos. Como hemos dicho antes en una guardaremos los nombres y en la otra las notas por asignatura.

Solamente nos queda contabilizar los aprobados y suspensos mediante la sencilla condición de que la nota —almacenada en un lugar determinado de la tabla de dos dimensiones— sea o no menor que cinco.

Veámoslo sobre el Programa 11.

Programa dos

La mecánica de funcionamiento de este programa es muy sencilla. Con las líneas 40 y 50 conocemos las dimensiones de las matrices. En ellas damos al ordenador los valores máximos que han de tener los subíndices de las variables «alumnos» y «asignaturas».

Mediante las órdenes DIM de las líneas 60 y 70 reservamos en la memoria el espacio necesario para contener los elementos de las dos matrices «nombre\$» y «notas» que necesitamos definir y dimensionar.

Usaremos el bucle FOR... NEXT de las líneas 90 y 170 para la adquisición de datos de la totalidad de alumnos haciendo que la variable de control «i» tome valores comprendidos entre 1 y el contenido de «alumnos».

Para recorrer la toma de notas de cada asignatura dentro de un determinado alumno nos encontramos con el otro bucle FOR... NEXT de las líneas 120 y 170. En este caso la variable de control es «j» y vamos tomando valores comprendidos entre 1 y el número de asignaturas almacenadas en «asignaturas».

Con estos dos bucles FOR... NEXT anidados estamos seguros de recoger todos los datos sobre las notas de cada asignatura para todos los alumnos. Si usted cree que no es así, admitámoslo y discutiremos sus sugerencias.

En la línea 150 incrementamos el valor del elemento, de la fila 0 en la columna correspondiente a una determinada asignatura cada vez que nos encontremos un suspenso en ella —dentro de la matriz bidimensional de las notas— otra utilización de dicho tipo de elementos.

Sacamos el resultado del número de aprobados con un sencillo FOR... NEXT comprendido entre las líneas 210 y 230.

Quizá la novedad de este programa II se encuentre en la parte que se encarga de todos los datos de un alumno en particular.

En la línea 260 se realiza la entrada (INPUT) del nombre del alumno cuyos datos queremos encontrar. Este valor literal lo almacenamos en la variable «nombre».

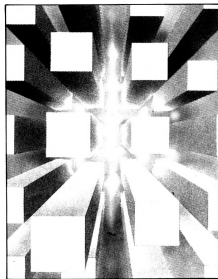
El bucle WHILE... WEND localiza entre las líneas 270 y 290 es el que explora dentro de la matriz que contiene los nombres hasta que encuentra un elemento igual al que nosotros le hemos dado en la línea 260. Su misión es localizar un índice que nos diga concretamente el número de la fila en la que se encuentran todos los elementos que contienen datos sobre el alumno por el que estamos preguntando en la tabla bidimensional de las notas.

Observen atentamente esta línea:

```
270 WHILE nombre$(indice) < + nombre$
```

En ella encontramos dos variables que aparentemente tiene el mismo nombre —«nombre\$». ¿Qué ocurre aquí? ¿Es posible que esto suceda o nos saldrá en la pantalla un mensaje de error?

No se preocupe, no ocurrirá nada. Nuestro Amstrad es capaz de admitir una variable numérica o alfanumérica que tenga el mismo nombre que una matriz. Como tenemos una máquina muy observadora, rápidamente se dará cuenta de que en uno de los casos se utilizan subíndices y en el otro no y sabrá distinguir entre ambos.



Finalmente, una vez encontrado el índice correspondiente al alumno que nosotros buscábamos, utilizamos este valor contenido en la variable «índice» para situarnos en la fila deseada dentro de la matriz de las notas y sacar el informe pedido, líneas 300 a 320.

Antes de continuar, vamos a pensar dos cosas de una cierta importancia. ¿Es necesario dimensionar las matrices que tienen más de un subíndice? ¿Podemos, una vez utilizados sus elementos, dimensionar de otra forma una tabla? ¿Qué se le ocurre?

Vayamos paso por paso. Cuando vimos los arrays de una dimensión, ¿en qué casos era necesario dimensionarlos? Su respuesta, imaginamos que habrá sido acertada, podemos aplicarla también a las matrices de más de una dimensión.

Siempre que no se diga nada el Amstrad supondrá que el mayor subíndice que vamos a utilizar será el 10. Si utilizamos uno mayor tendremos que haber definido previamente la matriz:

```
DIM matriz (100,80)
```

o si no estaremos expuestos al conocido mensaje:

```
Subscript out of range in 1000
```

por ejemplo.

Por tanto, si no dimensionamos una matriz, nuestro ordenador reserva en memoria el espacio equivalente al que guardaría de haberle dado la orden:

```
DIM matriz (10,10)
```

No obstante, le recomendamos que, aunque sólo sea por claridad, defina siempre al comienzo del programa las matrices cuyos elementos se van a manipular a lo largo de su ejecución.

Podemos hacerlo en cualquier parte del mismo, antes de la utilización de uno de sus elementos, pero es conveniente dimensionarlas al principio.

Para contestar a la segunda pregunta veamos el programa 111.

Programa tres

Es un programa que de momento la única aplicación que tiene es contestarle a su pregunta.

En las líneas 20 y 60 dimensionamos de dos maneras diferentes dos matrices que tienen el mismo nombre. Si ejecutamos este programa nos aparecerá en pantalla el siguiente mensaje:

Array already dimensioned in 60 que nos indica que una de las tablas declaradas en una orden DIM ya ha sido dimensionada anteriormente.

Sobre todo si utilizamos listas grandes, puede darse el caso que necesitemos el espacio reservado a una de ellas, cuyos elementos ya han sido utilizados, para otra que vamos a usar a partir de un determinado momento. Pero hemos visto que si la redimensionamos, el **Amstrad** nos dará un mensaje de error. **¿Cómo solucionamos la papeleta?**

Pruebe a meter en el programa III la siguiente línea:

50 ERASE conjunto

Si ejecuta ahora el programa verá que se le ha arreglado su problema y que su ordenador no le regañará con un mensaje de error.

¿Qué hace esta nueva línea que hemos añadido? Es muy sencillo. Esta nueva orden borra el contenido de las tablas que especificamos a continuación y libera la memoria del espacio que estaba ocupado por ellas. De esta forma podrá volver a dimensionarlas a su gusto sin gastar nueva memoria ni usar variables nuevas. Pero ¡ojó!, no se olvide que esta orden borra todos los contenidos anteriores de la matriz no sea que vayamos a tener un «fatal accidente».

Para terminar se nos ocurre proponerle una pequeña «tarea» ¿Se acuerdan de las votaciones para elegir la mejor revista informática? ¿Qué le parecieron los resultados?

Nuestra idea es que, para ver si les ha quedado suficientemente clara toda esta filosofía sobre tablas, matrices arrays o como queramos llamar a los conjuntos de variables con subíndices, hagan las modificaciones oportunas al programa para que sea capaz de hacer un recuento de votos semejante al anterior, pero con la diferencia de que ahora se añada una pregunta más al cuestionario:

¿Es usted usuario de Amstrad o de Spectrum?

Si es de Amstrad contestará con un 1, si es de Spectrum con un 2.

Le sugerimos que defina una matriz de dos dimensiones en la que el primer subíndice indique la revista

elegida y el segundo el tipo de usuario.

Es decir, en la representación bidimensional las filas serían los números correspondientes a cada revista y las columnas serían los de cada usuario. Por tanto, los subíndices serían respectivamente 4 y 2.

No se olvide poner la marca de final de respuestas (—1) en las líneas de DATAS y que cada respuesta está ahora compuesta por dos números. Vigile la cantidad de cifras que hay en estas líneas para que el nú-

mero de datos que leemos sea correcto.

Cuando ya lo tenga compárelo con el resultado que nosotros le proponemos como programa IV y vea los puntos comunes.

Programa cuatro

Bueno, no queremos atosigarle más. Practique...

PROGRAMAS

```

10 REM PROGRAMA I
20 CLS
30 INPUT "NUMERO DE FILAS: ",filas
40 INPUT "NUMERO DE COLUMNAS: ",columnas
50 DIM conjunto(filas,columnas)
60 REM DAR VALORES A LOS ELEMENTOS
70 PRINT
80 FOR i=1 TO filas
90 FOR j=1 TO columnas
100 PRINT"fila:";i; ", columna:";j;
110 INPUT valor
120 conjunto(i,j)=valor
130 NEXT j
140 NEXT i
150 REM CONOCER ALGUN VALOR
160 WHILE condicion=0
170 PRINT
180 PRINT "VALOR DE UN ELEMENTO"
190 INPUT "fila: ",fila
200 INPUT "columna: ",columna
210 numero=conjunto(fila,columna)
220 PRINT "conjunto(";fila";";columna)";":";numero
230 WEND

```

```

280 indice=indice+1
290 WEND
300 FOR i=1 TO asignaturas
310 PRINT "NOTA ASIGNATURA NUMERO";
320 NEXT i
330 ;:";notas(indice,i)
340 NEXT indice

```

```

10 REM PROGRAMA III
20 DIM conjunto(15,6)
30 conjunto(8,5)=2345
40 PRINT conjunto(8,5)
60 DIM conjunto(30,90)
70 FOR n=1 TO 100
80 INPUT a
90 conjunto(n,1)=a
100 NEXT n

```

```

10 REM PROGRAMA IV
20 DIM votos(4,2)
30 CLS
40 REM RECUESTO DE VOTOS
50 READ revista,usuario
60 WHILE revista<>-1
70 votos(0,usuario)=votos(0,usuario)+1
80 votos(revista,usuario)=votos(revista,usuario)+1
90 READ revista,usuario
100 WEND
110 REM IMPRESION DE RESULTADOS
120 PRINT
130 PRINT"MICROHOBBY: ";
140 PRINT TAB(6);"USUARIO AMSTRAD-";
150 PRINT TAB(6);"VOTOS";
160 PRINT TAB(6);"USUARIO SPECTRUM-";
170 PRINT TAB(6);"VOTOS";
180 PRINT"MICROHOBBY-CASSETTE:";
190 PRINT TAB(6);"USUARIO AMSTRAD-";
200 PRINT TAB(6);"VOTOS";
210 PRINT TAB(6);"USUARIO SPECTRUM-";
220 PRINT TAB(6);"VOTOS";
230 PRINT TAB(6);"USUARIO AMSTRAD-";
240 PRINT TAB(6);"VOTOS";
250 PRINT
260 PRINT"TOTAL VOTOS EMITIDOS:";
270 DATA 1,1,3,2,2,1,4,1,3,2,1,1,2,2,3,1,1,2,3,2,4,1,2,2,3,1,4,2,3,1,-1,-1

```

CONTABILIDAD DOMESTICA

Este programa, creado por nuestro lector Daniel Palomo, pretende resolver, y de hecho resuelve, el siempre molesto problema de llevar las cuentas (alias contabilidad) de nuestra casa, permitiéndonos llevar un exacto control de nuestros ingresos y gastos, resúmenes anuales incluidos.

E

ste programa, no implementa los cientos de funciones de las contabilidades profesionales, ni posee un sistema indexado de ficheros, ni lleva código máquina, ni... falta que le hace.

El **Amstrad Basic** es lo suficientemente potente para permitirnos crear una aplicación que tan sólo pretende cubrir las necesidades de la gestión de un hogar.

Contabilidad doméstica está pensado para manejar datos de una forma muy consistente, y esto implica el recurrir a ficheros (*secuenciales, de los sencillitos*) para atesorar la información y poder rehusarla cuando haga falta.

Esto quiere decir que el programa funcionará perfectamente en cualquier **Amstrad**, con o sin disco, pero creemos no exagerar mucho al decir que un disco, si no es imprescindible, es altamente recomendable. Recorrerse toda la cinta del cassette buscando los datos del mes de julio del 85 puede ser bastante desagradable.

Contabilidad doméstica se rige por una estructura jerárquica de menús, es decir, existe un menú principal de opciones, del cual, una vez elegida cualquiera de ellas, pasamos a un menú secundario o a la propia introducción de datos.

[C]ARGAR	[A]BANDONAR
[O]BRAR	[M]ODIFICACION
[E]NTRADAS	[R]ESUMEN ANUAL
* ENERO	
FEBRERO	
MARZO	
ABRIL	
MAYO	
JUNIO	
JULIO	
AGOSTO	
SEPTIEMBRE	
OCTUBRE	
NOVIEMBRE	
DICIEMBRE	
UTILIZA EL CURSOR PARA ELEGIR	
MES DESPUES <ENTER>	

[C]ARGAR	[A]BANDONAR
[O]BRAR	[M]ODIFICACION
[E]NTRADAS	[R]ESUMEN ANUAL
JUNIO	
INGRESOS	
CAJA	10000
BANCO	5321
SUELDO	60000
VARIOS	1000
CORRECTO <S/N>	

Las opciones del menú principal son las que cabría esperar de un programa de este tipo: [E]ntrar datos, [M]odificarlos, [G]rabar y [C]argar de disco, [R]esumen anual, etc.

Como ya se habrán imaginado, se accede a una opción pulsando la inicial entre corchetes.

Aunque el programa en todo momento suministra información en pantalla suficiente, vamos a examinar con cierto detalle la opción de [E]ntrar datos.

Al introducirnos en ella, lo primero que se observa es un submenú con los doce meses del año: tenemos que elegir, moviendo el cursor, en el cual queremos anotar nuestras cuentas.

M. BARCO





Serie Oro

A su vez, después de escogido el mes pertinente, hay que introducir dos tipos de datos más: lo que podríamos llamar saldo, o sea, el dinero que tenemos en ese mes en el Banco, en Caja, etc., y luego las partidas de gastos domésticas, como luz, gas, agua, coche, etc.

En fin, un programa no excesivamente largo, elegantemente programado, sencillo de usar y que cumple perfectamente su función.

ESTRUCTURA DEL PROGRAMA

70	Trampa para capturar errores.
80 -170	Inicialización de matrices, definición de colores y ventanas.
170 -200	Menú principal.
210 -230	Bucle principal.
240 -700	Rutina general de introducción y verificación de datos.
720 -840	Grabar fichero de datos en cinta/disco.
870 -1020	Cargar fichero de datos de cinta/disco.
1040-1070	Modificación de datos.
1090-1130	Fin del programa.
1150-1280	Gastos a cargo de Banco o Caja.
1300-1370	Impresión de ingresos.
1390-1520	Impresión de gastos.
1540-1720	Rutina de elección.
1740-2030	Resumen anual.

CARGAR	IMPRIMIR
GRABAR	MODIFICACION
ENTRADAS	RESUMEN ANUAL
ENERO	
GASTOS	
CAS	1000
LPS	1000
CAJA	b
CAJA	1000
MEDICO	b
TELEFONO	1000
ESTUDIOS	b
UTILIDADES	1000
AUTOMOVITO	b
IMPRESVISTOS	1000
MUESTRAR AL MENU	
CARGAR GASTOS	
ELEGIR INGRESOS	

```

10 REM *****
*****
20 REM **          CONTABLE
**
30 REM *****
***
40 REM
50 REM por Daniel Falomo
60 REM
70 ON ERROR GOTO 2040
80 MODE 1:INK 1,24:PEN 1:INK 0,1:PA
PER 0
90 CLS:CLER:RESTORE:PEN 1:PAPER 0
100 is=INKEY$:us="*****":DIM qs(
12):FOR w=1 TO 12:READ qs(w):NEXT:D
IM a(24)
110 READ sus,vs,ls,bs,cs,qs,ag$,l$
,tls,ga$,ca$,ve$,au$,dr$,es$,im$
120 DATA ENERO,FEBRERO,MARZO,ABRIL,
MAYO,JUNIO,JULIO,AGOSTO,SEPTIEMBRE,
OCTUBRE,NOVIEMBRE, DICIEMBRE
130 DATA SUELDO,VARIOS,INGRESOS,BAN
CO,CAJA,GASTOS,AGUA,LUZ,TELEFONO,GA
S,CASA,VESTUARIO,AUTOMOVIL,MEDICO,E
STUDIOS, IMPREVISTOS
140 WINDOW 1,40,23,25:PEN 1
150 WINDOW #1,1,40,5,21:PEN #1,1
160 WINDOW #2,1,40,1,3:PEN #2,1
170 INK 1,26:INK 2,10:PAPER #1,2:PA
PER #2,2:PAPER 2:CLS #1:CLS #2:CLS
180 LOCATE #2,5,1:PRINT #2,"[C]ARGA
R":LOCATE #2,20,1:PRINT #2,"[A]BAND
ONAR"
190 LOCATE #2,5,2:PRINT #2,"[G]RABA
R":LOCATE #2,20,2:PRINT #2,"[M]ODIFI
CACION"
200 LOCATE #2,5,3:PRINT #2,"[E]NTRA
DAS":LOCATE #2,20,3:PRINT #2,"[R]ES
UMEN ANUAL"
210 GOSUB 1610
220 ON z GOSUB 240,740,880,1760,106
0,1090
230 z=0:GOTO 210
240 CLS
250 LOCATE 6,1:PRINT"UTILIZA EL CUR
SOR PARA ELEGIR"
260 LOCATE 10,3:PRINT"DESPUES <
ENTER>:"
270 FOR N=1 TO 12:LOCATE #1,15,n+3:
PRINT #1,q$(n):NEXT
280 s=4:LOCATE #1,13,s:PEN #1,3:PRI
NT#1,CHR$(243)
290 WHILE INKEY(18)=-1
300 IF INKEY(2)=0 THEN s=s+1:LOCATE
#1,13,s-1:PEN #1,2:PRINT#1," " :LOC
ATE #1,13,s:PEN #1,3:PRINT#1,CHR$(2
43)
310 IF INKEY(0)=0 THEN s=s-1:LOCATE
#1,13,s+1:PEN #1,2:PRINT#1," " :LOC
ATE #1,13,s:PEN #1,3:PRINT#1,CHR$(2
43)
320 IF s>15 THEN s=4:LOCATE #1,13,1
6:PEN #1,2:PRINT#1," " :LOCATE #1,13
,s:PEN #1,3:PRINT#1,CHR$(243)
330 IF s<4 THEN s=15:LOCATE #1,13,3
:PEN #1,2:PRINT#1," " :LOCATE #1,13,
s:PEN #1,3:PRINT#1,CHR$(243)
340 FOR n=1 TO 200:NEXT:WEND
350 s=s-3:CLS:CLS #1
360 WHILE INKEY<>="" :WEND
370 LOCATE 5,2:PRINT"UTILIZA SOLO E
L TECLADO NUMERICO":FOR N=1 TO 2000
:NEXT:CLS
380 GOSUB 2000
390 GOSUB 1320
400 LOCATE 18,2:PRINT c$:LOCATE #1,
30,8:GOSUB 1230:c=es:e=0:PRINT #1,US
ING us$:c
410 LOCATE 17,2:PRINT b$:LOCATE #1,
30,10:GOSUB 1230:b=es:e=0:PRINT #1,U
SING us$:b
420 LOCATE 17,2:PRINT su$:LOCATE #1
,30,12:GOSUB 1230:su=es:e=0:PRINT #1
,USING us$:su
430 LOCATE 17,2:PRINT v$:LOCATE #1,
30,14:GOSUB 1230:v=es:e=0:PRINT #1,U
SING us$:v
440 CLS:LOCATE 13,2:PRINT"CORRECTO
(S/N)":GOSUB 1270:IF INKEY(46)=0 TH
EN CLS:GOTO 390
450 GOSUB 2020
460 CLS #1:GOSUB 1410
470 CLS:LOCATE 18,2:PRINT ga$:LOCAT
E #1,30,6:GOSUB 1170:IF r=1 THEN ga=
es:e=0:PRINT#1,USING us$:ga ELSE gal
=es:e=0:PRINT#1,USING us$:gal
480 LOCATE 18,2:PRINT lu$:LOCATE #1
,30,7:GOSUB 1170:IF r=1 THEN lue=es
:e=0:PRINT#1,USING us$:lu ELSE luf=es
:e=0:PRINT#1,USING us$:luf
490 LOCATE 18,2:PRINT ag$:LOCATE #1
,30,8:GOSUB 1170:IF r=1 THEN age=es
:e=0:PRINT #1,USING us$:ag ELSE age=
es:e=0:PRINT #1,USING us$:agi
500 LOCATE 18,2:PRINT ca$:LOCATE #1
,30,9:GOSUB 1170:IF r=1 THEN ca=es:e
=0:PRINT #1,USING us$:ca ELSE cal=es
:e=0:PRINT #1,USING us$:cal
510 LOCATE 17,2:PRINT dr$:LOCATE #1
,30,10:GOSUB 1170:IF r=1 THEN dr=es
:e=0:PRINT #1,USING us$:dr ELSE dr1=
es:e=0:PRINT #1,USING us$:dr1
520 LOCATE 16,2:PRINT tl$:LOCATE #1
,30,11:GOSUB 1170:IF r=1 THEN tl=es
:e=0:PRINT #1,USING us$:tl ELSE tll=
es:e=0:PRINT #1,USING us$:tll
530 LOCATE 16,2:PRINT es$:LOCATE #1
,30,12:GOSUB 1170:IF r=1 THEN es=es
:e=0:PRINT #1,USING us$:es ELSE es1=
es:e=0:PRINT #1,USING us$:es1
540 LOCATE 15,2:PRINT ve$:LOCATE #1
,30,13:GOSUB 1170:IF r=1 THEN ve=es
:e=0:PRINT #1,USING us$:ve ELSE ve1=
es:e=0:PRINT #1,USING us$:ve1
550 LOCATE 15,2:PRINT au$:LOCATE #1
,30,14:GOSUB 1170:IF r=1 THEN au=es
:e=0:PRINT #1,USING us$:au ELSE au1=
es:e=0:PRINT #1,USING us$:au1
560 LOCATE 14,2:PRINT im$:LOCATE #1
,30,15:GOSUB 1170:IF r=1 THEN im=es
:e=0:PRINT #1,USING us$:im ELSE im1=
es:e=0:PRINT #1,USING us$:im1
570 GOSUB 1560
580 CLS:LOCATE 13,2:PRINT"CORRECTO
(S/N)":GOSUB 1270:IF INKEY(46)=0 TH
EN CLS:GOTO 460
590 WHILE INKEY<>="" :WEND
600 b=b+su-(agi+luf+tl+gal+cal+vel
+au1+dr1+es1+im1)
610 c=c+iv-(ag+luf+tl+ga+ca+ve+au+dr
+es+im)
620 IF b<0 THEN CLS:LOCATE 9,1:PRIN
T"NO HAY FONDOS EN EL BANCO":LOCATE
5,3:PRINT"PULSA UNA TECLA PARA CON
TINUAR":GOSUB 1270

```

Serie Oro

```
630 IF c<0 THEN CLS:LOCATE 9,1:PRIN
T"NO HAY FONDOS EN LA CAJA":LOCATE
5,3:PRINT"PULSA UNA TECLA PARA CONT
INUAR":GOSUB 1270
640 CLS:LOCATE 12,1:PRINT"M:VUELVE
AL MENU":LOCATE 11,2:PRINT"G:VISUAL
IZA GASTOS":LOCATE 10,3:PRINT"I:VIS
UALIZA INGRESOS"
650 IF INKEY(38)=0 THEN 690
660 IF INKEY(52)=0 THEN CLS #1:GOSU
B 1410:GOSUB 1560
670 IF INKEY(35)=0 THEN CLS #1:GOSU
B 1320
680 GOTO 650
690 WHILE INKEY<>"":WEND
700 RETURN
710 '
720 REM          GRABAR
730 '
740 CLS:LOCATE 10,2:PRINT"GRABANDO:
"q$(s1):q$(s2)=LEFT$(q$(s1),7):OPEN
UT q$(s)
750 a[12]=v:a[13]=a[14]=i:m:a[15]
]=g:a[16]=l:a[17]=a[18]=c:a[19]=d:r:
a[20]=t:l:a[21]=e:s:l:a[22]
]=v:e:l:a[23]=a[24]=i:m
760 a[1]=b:a[2]=c:a[3]=s:a[4]=v:a[5]
]=g:a[6]=l:a[7]=a[8]=c:a[9]=d:r:a[10]=t:l:a[11]=e
s
770 FOR n=1 TO 24:WRITE #9,a[n]:NEX
T
780 a[12]=v:a[13]=a[14]=i:m:a[15]
]=g:a[16]=l:a[17]=a[18]=c:a[19]=d:r:
a[20]=t:l:a[21]=e:s:l:a[22]
]=v:e:l:a[23]=a[24]=i:m
790 b=a[1]:c=a[2]:s=a[3]:v=a[4]:g=
a[5]:l=a[6]:a[7]=a[8]:c=a[9]:d=r:a[10]=t:l:a[11]=e
s
800 CLOSEOUT:CLS:LOCATE 10,2:PRINT"
GRABACION TERMINADA"
810 LOCATE 12,3:PRINT"M:VUELVE AL M
ENU"
820 WHILE INKEY(38)=-1:WEND
830 WHILE INKEY<>"":WEND
840 RETURN
850 '
860 REM          CARGAR
870 '
880 CLS:LOCATE 11,1:PRINT"NUMERO DE
L FICHERO":LOCATE 13,2:PRINT"MAX. B
LETRAS"
890 CLS #1:WINDOW SWAP 0,1:CAT:WIND
OW SWAP 1,0
900 LOCATE 15,3:INPUT n$:IF LEN(n$)
>8 THEN 880
910 OPENIN n$:s=0:q$(s)=n$
920 FOR n=1 TO 24:INPUT #9,a[n]:NEX
T
930 v=a[12]:a[13]=a[14]:i=m:a[15]=g:a[16]=l:a[17]=a[18]=c:a[19]=d:r:
a[20]=t:l:a[21]=e:s:l:a[22]
]=v:e:l:a[23]=a[24]=i:m
940 b=a[1]:c=a[2]:s=a[3]:v=a[4]:g=
a[5]:l=a[6]:a[7]=a[8]:c=a[9]:d=r:a[10]=t:l:a[11]=e
s
950 CLS #1:GOSUB 1410:GOSUB 1560
960 CLS:LOCATE 12,1:PRINT"M:VUELVE
AL MENU":LOCATE 11,2:PRINT"G:VISUAL
IZA GASTOS":LOCATE 10,3:PRINT"I:VIS
UALIZA INGRESOS"
970 IF INKEY(38)=0 THEN 1010
980 IF INKEY(52)=0 THEN CLS #1:GOSU
B 1410:GOSUB 1560
```

```
990 IF INKEY(35)=0 THEN CLS #1:GOSU
B 1320
1000 GOTO 970
1010 WHILE INKEY<>"":WEND
1020 CLOSEIN:RETURN
1030 '
1040 REM          MODIFICAR
1050 '
1060 CLS:LOCATE 4,2:PRINT"TECLEE T
O DAS LAS CIFRAS DE NUEVO":LOCATE 12,
3:PRINT"PULSA UNA TECLA
1070 GOSUB 1270:CLS:GOSUB 390
1080 '
1090 REM          ABANDONAR
1100 '
1110 CLS #1:CLS:LOCATE #1,10,7:PRIN
T#1,"SI ABANDONA PERDERA":LOCATE #1
,12,8:PRINT#1,"TODOS LOS DATOS"
1120 LOCATE #1,11,9:PRINT#1,"CONFIR
ME PULSANDO":LOCATE #1,18,10:PRINT#
1,"[S]"
1130 GOSUB 1270:IF INKEY(60)=0 THEN
CLS:NEW ELSE RETURN
1140 '
1150 REM          ENTRADAS B/C
1160 '
1170 LOCATE 7,3:PRINT"CARGO A BANCO
O CAJA (B/C)
1180 IF INKEY(54)=0 THEN R=0:GOTO 1
210
1190 IF INKEY(62)=0 THEN R=1 ELSE G
OTO 1180
1200 PEN 2:LOCATE 7,3:PRINT STRING$(
26," ") :PEN 1
1210 WHILE INKEY<>"":WEND
1220 PEN 2:LOCATE 6,3:PRINT STRING$(
30," ")
1230 PEN 1:LOCATE 15,1:PRINT "INTRO
DUCE"
1240 LOCATE 15,3:INPUT e:es=STR$(e)
:]>LEN(es):IF #]8 THEN LOCATE 15,3:
PRINT STRING$(20," ") :GOTO 1240
1250 PEN 2:LOCATE 15,1:PRINT STRING
$(10," ") :PEN 1
1260 CLS:RETURN
1270 WHILE INKEY<>"":WEND
1280 RETURN
1290 '
1300 REM          INGRESOS
1310 '
1320 PEN #1,3,1=(10-LEN(q$(s)))/2:L
OCATE #1,15,2:PRINT#1,SPC(1)Q$(s):L
OCATE #1,16,5:PRINT #1,i$
1330 PEN #1,1:LOCATE #1,10,8:PRINT
#1,c$:LOCATE #1,30,8:PRINT #1,USING
u$:c
1340 LOCATE #1,10,10:PRINT #1,b$:LO
CATE #1,30,10:PRINT #1,USING u$:b
1350 LOCATE #1,10,12:PRINT #1,su$:L
OCATE #1,30,12:PRINT #1,USING u$:s
1360 LOCATE #1,10,14:PRINT #1,v$:LO
CATE #1,30,14:PRINT #1,USING u$:v
1370 RETURN
1380 '
```

```

1390 REM          GASTOS
1400 '
1410 FEN #1,3:1=(10-LEN(q$(s)))/2:L
OCATE #1,15,2:PRINT#1,SPC(1)Q$(s):L
OCATE #1,17,4:PRINT #1,q$;FEN #1,1
1420 LOCATE #1,10,6:PRINT#1,ga$:LOC
ATE #1,30,6:PRINT#1,USING u$;ga+ga1
1430 LOCATE #1,10,7:PRINT#1,lu$:LOC
ATE #1,30,7:PRINT#1,USING u$;lu+lu1
1440 LOCATE #1,10,8:PRINT#1,ag$:LOC
ATE #1,30,8:PRINT#1,USING u$;ag+ag1
1450 LOCATE #1,10,9:PRINT#1,ca$:LOC
ATE #1,30,9:PRINT#1,USING u$;ca+ca1
1460 LOCATE #1,10,10:PRINT#1,dr$:LO
CATE #1,30,10:PRINT#1,USING u$;dr+d
r1
1470 LOCATE #1,10,11:PRINT#1,t1$:LO
CATE #1,30,11:PRINT#1,USING u$;t1+t
11
1480 LOCATE #1,10,12:PRINT#1,es$:LO
CATE #1,30,12:PRINT#1,USING u$;es+e
s1
1490 LOCATE #1,10,13:PRINT#1,ve$:LO
CATE #1,30,13:PRINT#1,USING u$;ve+v
e1
1500 LOCATE #1,10,14:PRINT#1,au$:LO
CATE #1,30,14:PRINT#1,USING u$;au+a
u1
1510 LOCATE #1,10,15:PRINT#1,im$:LO
CATE #1,30,15:PRINT#1,USING u$;im+i
m1
1520 RETURN
1530 '
1540 REM          OPCIONES
1550 '
1560 a[12]=ve:a[13]=au:a[14]=im:a[1
5]=ga:l:a[16]=lu:l:a[17]=ag:l:a[18]=ca
:l:a[19]=dr:l:a[20]=t:l:a[21]=es:l:a[2
2]=ve:l:a[23]=au:l:a[24]=im:l
1570 a[1]=b:a[2]=c:a[3]=s:a[4]=v:a
[5]=g:a[6]=l:a[7]=a[8]=c:a[9]=d:r:a[10]=t:l:a[11]=es
1580 FOR n=5 TO 14:IF a[n]=0 AND a[
10+n]<>0 THEN LOCATE #1,28,n+1:PRIN
T #1,"b"
1590 NEXT
1600 RETURN
1610 CLS:LOCATE 6,1:PRINT"PARA ELEG
IR OPCION PULSA LA"
1620 LOCATE 6,3:PRINT"LETRA METIDA
ENTRE CORCHETES"
1630 CLS #1:WHILE INKEY$=""
1640 IF INKEY(58)=0 THEN z=1
1650 IF INKEY(52)=0 THEN z=2
1660 IF INKEY(62)=0 THEN z=3
1670 IF INKEY(50)=0 THEN z=4
1680 IF INKEY(38)=0 THEN z=5
1690 IF INKEY(68)=0 THEN z=6
1700 WEND
1710 IF z>7 OR z<1 THEN 1630
1720 RETURN
1730 '
.1740 REM          RESUMEN

```

```

1750 '
1760 CLS:LOCATE 5,1:PRINT"INTRODUZ
A EL DISCO QUE CONTENGA":LOCATE 7,2
:PRINT"LOS FICHEROS DE LOS MESES" LA
OCATE 8,3:PRINT"DESPUES PULSE UNA T
ECLA":GOSUB 1270
1770 CLS:LOCATE 2,1:PRINT"ESTAN TOD
OS LOS MESES EN ESTE DISCO?":LOCATE
17,2:PRINT"(S/N)"
1780 WINDOW SWAP 0,1:CAT:WINDOW SWA
P 1,0
1790 IF INKEY(46)=0 THEN RETURN
1800 IF INKEY(60)=0 THEN 1820
1810 GOTO 1790
1820 CLS:LOCATE 11,2:PRINT"CARGANDO
MES DE:"
1830 FOR k=1 TO 12:LOCATE 15,3:PRIN
T Q$(k);" " :O$(k)=LEFT$(Q$(k),7
):OPENIN O$(k)
1840 FOR n=1 TO 24:INPUT #9,a[n]:NE
XT
1850 ve=ve+a[12]:au=au+a[13]:im=im+
a[14]:ga=ga+a[15]:lu=lu+a[16]:a
g=ag+a[17]:ca=ca+a[18]:dr=dr+l+a
a[19]:t1=t1+a[20]:es1=es1+a[21]:v
e1=ve1+a[22]:au1=au1+a[23]:im1=im1+
a[24]
1860 b=b+a[1]:c=c+a[2]:s=s+a[3]:v
=v+a[4]:g=g+a[5]:l=l+a[6]:a[7]=a
a[7]:ca=ca+a[8]:dr=dr+a[9]:t1=t1+a
[10]:es=es+a[11]
1870 CLOSEIN:NEXT
1880 s=0;q$(0)="RESUMEN"
1890 CLS:LOCATE 12,2:PRINT"MI VUELVE
AL MENU":FOR N=1 TO 3000:NEXT
1900 CLS:LOCATE 13,1:PRINT"S:GRABA
RESUMEN":LOCATE 11,2:PRINT"G:VISUAL
IZA GASTOS":LOCATE 10,3:PRINT"1:VIS
UALIZA INGRESOS"
1910 IF INKEY(60)=0 THEN 1960
1920 IF INKEY(38)=0 THEN RETURN
1930 IF INKEY(52)=0 THEN CLS #1:GOS
UB 1410
1940 IF INKEY(35)=0 THEN CLS #1:GOS
UB 1320
1950 GOTO 1910
1960 CLEAR INPUT
1970 CLS:LOCATE 13,1:PRINT"CON QUE
NOMBRE":LOCATE 13,2:PRINT"(MAX. 8 L
ETRAS)"
1980 LOCATE 16,3:INPUT n$:s=0;q$(s)
=n$:GOSUB 720
1990 RETURN
2000 b=0:c=0:v=0:su=0
2010 RETURN
2020 ga=0:gal=0:lu=0:lu1=0:ag=0:ag1
=0:ca=0:cal=0:dr=0:dr1=0:t1=0:t11=0
:es=0:es1=0:ve=0:ve1=0:au=0:au1=0:i
m=0:im1=0
2030 RETURN
2040 CLS:CLS #1:LOCATE 11,2:PRINT"H
A HABIDO UN ERROR":FOR N=1 TO 1000:
NEXT
2050 CLS:LOCATE 13,2:PRINT"VUELVO A
L MENU":FOR N=1 TO 2000:NEXT
2060 RESUME 220

```



P ara que tus dedos no realicen el trabajo duro, M.H. AMIS TRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicítalo.

DESDE EL PRIMERO HASTA EL ULTIMO...



... te falta
alguno?

¡APROVECHATE!

Ahora es la ocasión de completar tu colección, porque sólo hasta el 28 de febrero de 1986 podrás adquirir los ejemplares que te faltan...

...sin tener que pagar el IVA

¡Realiza hoy mismo tu pedido!

RECORTA O COPIA ESTE CUPON Y MANDALO A HOBBY PRESS, S. A.
APARTADO DE CORREOS 232. ALCOBENDAS (MADRID)

En números sueltos sólo se cobran gastos de envío en los pedidos contra reembolso, a razón de 50 ptas. por cada envío (no por cada ejemplar)

Nombre _____

Apellidos _____

Dirección _____

Localidad _____ Provincia _____ C. Postal _____

Teléfono _____ Profesión _____ Edad _____

¿Eres suscriptor de MICROHOBBY AMSTRAD? _____

N.º de suscriptor (si lo recuerdas) _____

Deseo recibir en mi domicilio los siguientes ejemplares atrasados de MICROHOBBY AMSTRAD al precio de 150 ptas. del n.º 1 al 18, Cuyos números indico _____

Forma de pago:

Mediante talón bancario adjunto a nombre de Hobby Press, S. A.

Mediante giro postal a nombre de Hobby Press, S. A. n.º _____

Contra reembolso del envío

Mediante tarjeta de crédito _____

Número de la tarjeta _____

Fecha de caducidad de la tarjeta _____

Fecha y firma: _____

HOBBY PRESS, S.A.
Editamos para gente inquieta.

¡CUIDADO CON LOS DISCOS!

Una de las características del Amstrad que le dotan, de cara a los usuarios, de más potencia y popularidad, es sin duda alguna su unidad de discos integrada, que le capacita para almacenar y recuperar rápidamente grandes cantidades de información.



Los discos propiamente dichos son una pequeña maravilla tecnológica, que requieren un cierto cuidado y atención para que puedan seguir desempeñando su importante misión durante mucho tiempo.

Casi todas las personas que hayan tratado con estos artefactos saben más o menos las normas a seguir para el mantenimiento en perfecto estado de sus diskettes, pero lo que nadie sabe es el grupo de medidas a tomar para destruirlos completamente y sin remisión.

AMSTRAD Semanal, en su afán de huir de los caminos trillados, enuncia por primera vez en la historia del periodismo informático «**La Teoría General del Aniquilamiento Selectivo de los Discos**», creada por el genial matemático Gaffe McNazas, del MUTIS (*Massachusetts Technological Institute*).

En el momento de estremecer al mundo de la información con los postulados de su teoría, McNazas contaba en su haber con el increíble récord de pulverizar completamente las 5 bases de datos relacionales de máxima seguridad del MUTIS en un plazo no superior a dos meses.

Por primera vez en toda la historia del Instituto Tecnológico, los miembros del personal científico tuvieron que recurrir a la palabra hablada para saludarse (*y como son mudos, ya me contarás*), y al papel y a la pluma de ganso para escribir los informes: las computadoras habían callado.

Como enérgica protesta, el análisis de sistemas del MUTIS se suicidó disparándose un bolígrafo BIC a la cabeza.

Insensible a este tipo de infantiles críticas, McNazas observó una particularidad muy importante de su dilatada experiencia de destructor de discos, responsable en gran parte de la inmensa popularidad que alcanzó su teoría entre el personal informático: ni uno solo de sus propios diskettes se había visto afectado por la aplicación de sus reglas, ni uno solo de ellos había perdido tan siquiera un byte de información.

Gaffe McNazas comprendió que había descubierto la Aniquilación Selectiva de Discos, es decir, era capaz de destruir exclusivamente los discos de los demás.

Como toda teoría científica, la Aniquilación Selectiva se aplica a multitud de campos y parte de una serie de hipótesis de trabajo, de axiomas, aplicables casi con plena certeza a cada uno de ellos.

En el caso de la informática doméstica, se parte de la hipótesis de que nuestro amigo más íntimo ha cometido el lamentable error de confiarnos sus discos claves en los que almacena información vital, como por ejemplo, una lista de teléfonos de «ya sabes quiénes».

Para mayor simplicidad, aplicaremos el Pentálogo que resume la teoría de McNazas a este modelo:

- 1. Principio de las Condiciones Límite:** se confundirán los diskettes con los tranchettes y se introducirán en el horno a 50 grados centígrados dentro de una hamburguesa, ofreciéndoselos acto seguido a nuestro amigo como muestra del mayor aprecio. Corolario: al morderla, encima perderá un diente o más.

Nota: incredulidad cuando se queje de que saben a plástico.

- 2. Axioma de la Variabilidad Situacional:** debido a un genuino cruce de cables, el disco maestro e irremplazable se colocará cuidado-

M. BARCO



Amstra- vagancia



samente en la cisterna del inodoro, a guisa de pastilla desinfectante, con idea de que tiña primorosamente de azul las aguas de tan imprescindible aparato.

3. Ley del Encadenamiento Lógico: en el momento de desesperación que sucede inmediatamente después de ver la apariencia de nuestra esposa recién levantada de la cama, golpearemos con un glorioso directo el horrible jarrón de porcelana, regalo de nuestra suegra, en el cual previsoriamente habíamos almacenado los discos para ponerlos fuera del alcance de los niños. Los discos quedarán completamente destruidos, pero el jarrón, ni un rasguño. Llanto y crujir de rulos como telón de fondo.

4. Teoría del Almacenaje Aleatorio: bastará colocar los discos encima de nuestra mesa de trabajo y esperar a que transcurra el tiempo, dando permiso a nuestra familia y/o señora de la limpieza para que la «ordene un poco». Desaparecerán sin dejar rastro.

5. Postulado del Estúpido Desconocido: en el improbable caso de que ninguno de los cuatro principios anteriores encuentre aplicación en un plazo de tiempo razonable, no hay que preocuparse: cualquier idiota destruirá los discos en las condiciones más inverosímiles y con la misma eficacia que nosotros.

La Teoría General de Aniquilamiento Selectivo de Discos, debido a su complejidad matemática, queda más allá del alcance de este artículo, pero nos llena de gozo el saber que McNazas y sus discípulos siguen trabajando en su perfeccionamiento y desarrollo en el exilio pagado por sus compañeros de trabajo del MUTIS. Dentro de muy poco tiempo, no habrá un solo disco que pueda considerarse a salvo.

SABRE WULF

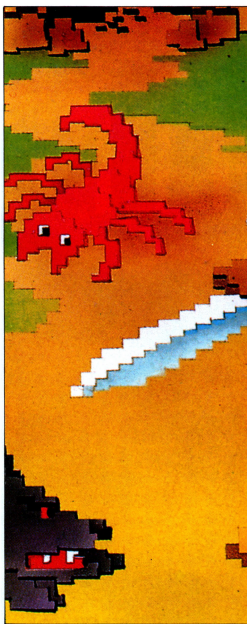
El intrépido explorador se adentra en la selva amazónica, en busca del secreto mágico del hechicero. Antes de llegar a la oculta cueva que lo alberga, tendrá que descubrir los recónditos senderos que conducen a ella y enfrentarse a los innumerables peligros que acechan entre la maleza.

S

abre Wulf es el programa que convirtió a la pequeña firma de software Ultimate, en la marca de reconocido prestigio que todos conocemos.

Nacido para el Spectrum, es el primero de la trilogía que engloba a Sabre Wulf, Underwulde y Knight Lore, los cuales aparecieron por este orden, revolucionando el mundo del software.

En las versiones para Amstrad, el orden ha sido alterado: en primer lugar el Knight Lore y meses después



el Sabre Wulf, estando todavía inédito el Underwulde.

«La senda es larga y tortuosa. El peligro amenaza en la guarida del lobo. Por el camino de la selva, al encuentro de tu destino, un amuleto debes encontrar. Fue roto en cuatro partes, y escondido en claros de la jungla. El guardián nunca te permitirá entrar en él».

Esta leyenda, encontrada en una antigua losa, es leída por nuestro explorador, el cual cautivado por el misterioso secreto escondido en la cueva del hechicero, no puede resistir la tentación y comienza la búsqueda.

Rodeado por la exuberante vegetación, Sabreman elige una senda, por la que encamina sus pasos hacia un destino desconocido. Cocoteros, palmeras y demás plantas tropicales, nos acompañan en nuestro recorrido, escondiendo los peligros que nos acechan.

A nuestro encuentro salen las terribles criaturas de la selva; monos, arañas, pájaros multicolores, alacranes, caca tuas, iguanas, surgen de los sitios más inesperados.

El machete de Sabreman da buena cuenta de ellos, eliminando bichos molestos de un solo tajo.

Otros enemigos no son tan fáciles de aniquilar: hipopótamos, watusis hambrientos, rinocerontes, etc., sólo pueden ser ahuyentados a golpe de sable, aunque ésta no es un arma capaz de aniquilarlos.

El lobo dientes de sable es inmune al poder de nuestro machete, constituyendo el enemigo más peligroso que hemos de encontrar en nuestro vagar por la tupida foresta.

Jalonando nuestro recorrido, encontramos exóticas orquídeas salvajes, las cuales podemos usar en nues-



Compatible: CPC1464, CPC1664 y CPC16128

Mr. Joystick

podemos eludir al guardián de la cueva del hechicero.

Una vez recogidos los cuatro, podemos decir que la aventura está en nuestras manos, sólo nos queda dirigirnos a la cueva y desvelar su secreto.

En Sabreman se ha utilizado el modo de resolución más bajo, lo cual ha dotado al programa de un colorido francamente notable, aunque esto haya provocado una pérdida de resolución.

En programas como el Knight Lore y Alien 8, se utiliza la resolución intermedia, con cuatro colores y una definición mayor.

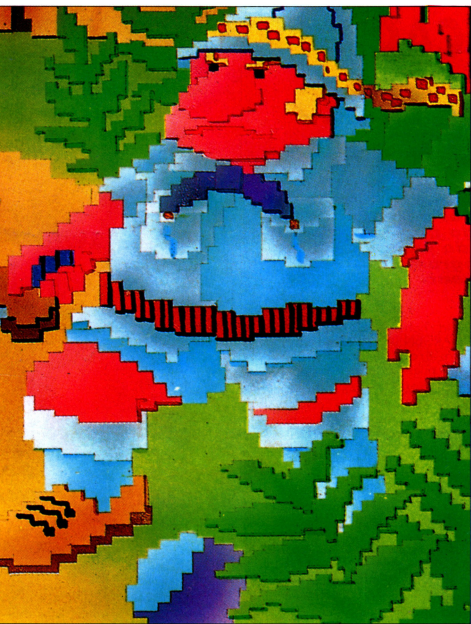
Utilizando dieciséis colores, los gráficos adquieren gran brillantez y la selva se muestra en todo su esplendor ante los atónitos ojos de Sabreman.

El escenario que recorre el intrépido explorador, está formado por 256 pantallas las cuales hemos de recorrer para encontrar los preciados trozos del talismán.

La estructura laberíntica de los caminos de la jungla y la distribución de los claves, nos obliga a elaborar un mapa de las distintas pantallas, para saber dónde nos encontramos y la ruta a seguir.

Sin éste, vagaríamos por las mismas sendas continuamente, dejando zonas inexploradas, que nos impedirían acabar el juego.

La consecución de la aventura costará horas y horas de entretenimiento.



tro favor, aunque otras nos afecten gravemente.

La orquídea púrpura, nos hace perder el sentido de la orientación.

La azul, pone alas a nuestro hombre, haciéndole correr tres veces más rápido.

Orquídea roja: hace a Sabreman inmune a los ataques de las fieras de la jungla.

Adormidera amarilla: hace que nuestro hombre caiga redondo perdiendo el conocimiento por unos instantes.

Orquídea blanca, anula el efecto de las anteriores.

Otros objetos distribuidos por los parajes de la foresta, incrementan nuestra puntuación.

Escondidos en los claros de la jungla, se encuentran las cuatro partes del mágico talismán, sin el cual no

MAGNIFICACION DE CARACTERES

Como su nombre indica, este programa le permite diseñar una pantalla de caracteres magnificados o aumentados de tamaño, para usarla posteriormente en presentaciones de juegos o de cualquier tipo de programa que se le ocurra crear.



El juego completo de caracteres estándar del Amstrad más los símbolos gráficos puede emplearse, o bien usted, si así lo desea, rediseñará y utilizará otro completamente nuevo.

MAGNIFICADOR es compatible con EASYDRAW (ver **AMSTRAD Semanal número 7**), así que se puede introducir texto aumentado de tamaño en diseños creados con él. La subrutina que comienza en la línea 820 se puede incluir en su propio programa y una demostración de este método se da en la «demo» con la que Magnificador comienza. Después de ésta, que puede evitarse si se desea suprimiendo la línea 130, aparecen en pantalla una serie de preguntas que el programa nos hace para averiguar una serie de datos vitales:

1) CARGAR JUEGO DE CARACTERES Y/N: un juego especial de caracteres puede cargarse en memoria seleccionando Y. Dicho juego de-

bería haberse diseñado usando CHARGEN (**AMSTRAD Semanal número 4**), o redefinido con Magnificador y puesto a salvo en cinta/disco.

2) FORMATO EASYDRAW Y/N: si se selecciona formato Easydraw, la pantalla útil es más pequeña, rodeada por un rectángulo rojo. El resultado de nuestro diseño puede cargarse: luego Easydraw, pero si se responde N a esta pregunta parte de los caracteres pueden perderse.

3) MODE 0, 1 Y 2: esta pregunta resulta obvia. Recuerde, sin embargo, que el número de colores disponible está en función del modo de pantalla elegido.

4) VOLCAR PANTALLA Y/N: el programa reserva 16 K de memoria para guardar una posible pantalla gráfica. Si existe, puede volcarse en pantalla y manipularse.

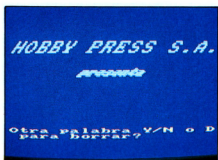
5) TECLEE PALABRA - ENTER: introduzca el mensaje que quiere magnificar seguido de ENTER. Alternativamente, si se pulsa primero la tecla ENTER pequeña del teclado numérico, los códigos ASCII entre 126 y 256 pueden introducirse y producirán el adecuado carácter gráfico.

Si se pulsa ENTER antes de que se haya introducido una palabra, aparecerá el mensaje «OTRA PALABRA Y/N»; respondiendo N tendremos acceso a las opciones load/save.

6) ESCALA 1 A N: dependiendo de la longitud de la palabra, se calcula una escala de impresión de la frase en pantalla. La escala 1 es menor que el tamaño normal de los caracteres, y sólo se puede usar en MODE 2.

7) INCLINACION Y/N: producir o no caracteres inclinados.





8) SOMBRA Y/N: si se selecciona «sombra», los caracteres tendrán un efecto tridimensional. Las teclas del cursor se usan para posicionar la sombra, en cualquiera de 8 posiciones. Para producir una sombra sobre la palabra, pulsa la tecla superior del cursor 2 veces. Cada vez que se pulsa una tecla del cursor, se produce un beep, así que debe oír dos de ellos.

9) PLUMA 0 A N: elección de pluma de dibujo. Los números 16 y 17 en MODE 0 son multicolores, así como los 4 y 5 en MODE 1. Cuando se elige la opción sombra, el color de la misma puede escogerse también.

$xt = yt$: a medida que la palabra se dibuja en la pantalla, la posición de comienzo se refleja en estas dos variables. Tome nota de estos valores si piensa usar la subrutina que comienza en la línea 820 en sus propios programas basic.

10) OTRA PALABRA Y/N O D PARA BORRAR: y para dibujar otra palabra, N para siguientes opciones o D para borrar la última palabra.

11) SALVAR PANTALLA Y/N O F CHARACTER: Y causa que la pantalla se almacene en la memoria de reserva del programa, N pasa a las siguientes opciones y F vuelve a cinta/disco el juego de caracteres redefinidos. Una opción posterior permite salvar la pantalla guardada en la memoria reservada en cinta/disco.

12) BORRAR PANTALLA Y/N O R REDEFINE: si se pulsa R, el programa preguntará por números de 0 a 225 en grupos de 8. Diseñe el nuevo juego en una parrilla de 8×8 y dele

Program Acción

los números al programa de arriba a abajo.

13) CARGAR A MEMORIA Y/N O C CARACTERES: una pantalla salva previamente por este programa o por Easydraw puede introducirse en la memoria de reserva usando Y. C permite cargar un nuevo juego de caracteres. La pantalla salva puede cargarse en memoria sin el auxilio de ningún programa mediante

LOAD «NOMBRE», &C000.

Por último, para incluir la subrutina de magnificación en su propio programa, construya la pantalla que sea usándolo. Tome nota de los valores de xt e yt y de las plumas usadas, etc.

Luego, emplee SYMBOL AFTER 32 al comienzo del programa y la subrutina que va de las líneas 820 a 1350.

VARIABLES PRINCIPALES DEL PROGRAMA

topmen	Posición de comienzo del juego de caracteres.
screenmode	Modo de pantalla.
movx	Movimiento de pixels.
choicepen	Plumas disponibles.
letter\$	Carácter de entrada.
mag\$	Palabra a ser magnificada.
mag	Escala.
slope	1 = caracteres inclinados.
shadow	1 = sombra sí, 0 = sombra no.
enlarge	Máxima escala posible.
xt	Posición horizontal del cursor de gráficos.
yt	Posición vertical.
rubx ruby	Coordenadas de borrado.
blockcol	Bloques coloreados = pluma + 2.
random	Bloques multicolores = pluma + 1.
bnum	Número de carácter.
num\$	Carácter ASCII gráfico en una cadena.
p	Número de pluma.
bit	Carácter redefinido.

```

10 REM *** MAGNIFICADOR ***
11
20 REM
30 REM *** by Glynne Davies ***
40 REM
50 REM(C)AMSTRAD SEMANAL
60 IF HINEM=24410 THEN INPUT "PULSE
D para disco":y$=d" OR y$="
D" THEN topmem=40955:ELSE topmem=42
239
70 IF HINEM=24410 THEN GOTO 110:REM
permite re-run
80 IF HINEM=43903 THEN topmem=42239
ELSE topmem=40955:REM # cinta/disc
O #
90 SYMBOL AFTER 32:REM esto introdu
ce juego de caracteres en ram en 42
240 O 40955(disco)
100 MEMORY 24410:REM # reserva memo
ria para volcar pantalla y para el
disco #
110 GOSUB 2070:REM # carga codigo m
aquina para el volcado de memoria #
120 REM *****
130 REM GOSUB 2190:REM demostracion
,quites si se desea mas velocidad
al arrancar el programa
140 CLS:y=200:x=1:left=0:bottom=32
150 LOCATE 10,10:INPUT #0,"Cargar j
uego de caracteres: Y/N":y$IF y$="Y
" OR y$="Y" THEN GOSUB 2140
160 CLS
170 LOCATE 9,12:INPUT "Formato Easy
draw Y/N":y$
180 IF y$="Y" OR y$="y" THEN bottom
=48:left=64:p=1
190 CLS
200 REM ** formato de pantalla **
210 LOCATE 9,12:INPUT "mode 0 1 o
2":screenmode:IF screenmode>2 THEN
GOTO 210
220 MODE screenmode
230 IF y$="d" OR y$="D" THEN GOTO 5
10:REM volvemos a preguntar
240 IF screenmode=2 THEN screenmode
=4:movx=1:choicepen=1:cp=1
250 IF screenmode=1 THEN screenmode
=2:movx=2:choicepen=3:cp=1
260 IF screenmode=0 THEN screenmode
=1:movx=4:choicepen=15:cp=14
270 WINDOW #0,1,20:screenmode,24,25
280 ORIGIN left,bottom,left,640,400
,bottom
290 IF e=1 THEN DRAW 574,0,3:DRAW 5
74,350,3:DRAW 0,350,3:DRAW 0,0,3
295 PRINT CHR$(2)
300 WHILE 1
310 CLS #0
320 LOCATE #0,1,1:INPUT #0,"Volcar
pantalla Y/N":y$IF y$="Y" OR y$="y
" THEN UNL 248:3
330 CLS #0:LOCATE #0,1,1:PRINT "Tec
lee palabra-ENTER":
340 mag$=""
350 ON ERROR GOTO 360
360 word=0
370 WHILE word=0
380 letter$=INKEY$:IF letter$="" TH
EN GOTO 380
390 IF INKEY(23)=128 THEN GOTO 380
400 IF INKEY(6)=0 THEN SOUND 1,110,
10:GOSUB 1530
410 IF INKEY(18)=0 THEN word=1:lett
er$=""GOTO 410:REM esperar hasta q
ue se suelta la tecla ENTER
420 IF INKEY(79)=0 THEN 1=LEN(mag$)
:mag$=LEFT$(mag$,1-1):letter$=""LD
CATE #0,1,2:PRINT #0,SPACES(20):
430 IF letter$<>"" THEN mag$=mag$+l
etter$:letter$=""
440 LOCATE #0,1,2:PRINT #0,mag$
450 IF mag$="" THEN GOTO 510:REM g
o a opciones
460 WEND:word=0
470 GOSUB 1360:REM # posicion de la

```

```

palabra elegida con el cursor #
480 GOSUB 620:REM comprobar la long
itud para dar las opciones de magni
ficacion
490 GOSUB 820:REM leer la cadena y
dibujarla magnificada
500 blockcol=0:random=0
510 CLS #0:LOCATE #0,1,1:INPUT #0,"
Otra palabra Y/N o D para borrar":y
$:IF y$="Y" OR y$="y" THEN GOTO 610
:REM repite bucle
520 IF y$="d" OR y$="D" THEN xt=ru
b
xt:yt=rub:y:p=0:GOSUB 820:IF shadow=1
THEN xt=shadxt:yt=shadyt:GOSUB 820
:shadow=0:REM borrar
530 IF y$="d" OR y$="D" THEN GOTO 5
10:REM volvemos a preguntar
540 CLS #0:LOCATE #0,1,1:INPUT #0,"
Salvar pantalla Y/N o F character":y
$:IF y$="Y" OR y$="y" THEN GOSUB 18
80
550 IF y$="f" OR y$="F" THEN GOSUB
2550
560 CLS #0:LOCATE #0,1,1:INPUT #0,"
Borrar pantalla Y/N o R redefinir":y
$:IF y$="Y" OR y$="y" THEN CLG
570 IF y$="r" OR y$="R" THEN GOSUB
2300
580 IF e=1 THEN DRAW 574,0,3:DRAW 5
74,350,3:DRAW 0,350,3:DRAW 0,0,3
590 CLS #0:LOCATE #0,1,1:INPUT #0,"
Cargar a memoria Y/N o C caracteres
":y$IF y$="Y" OR y$="y" THEN GOSUB
2020
600 IF y$="c" OR y$="C" THEN GOSUB
2140
610 slope=0:shadow=0:WEND
620 REM ** opciones de magnificac
ion **
630 length=LEN(mag$)
640 size=641-left-

```

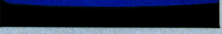
```

750 CLS #0:LOCATE #0,1,1:INPUT "Som
bra Y/N":y$IF y$="Y" OR y$="y" THE
N GOSUB 1640
760 CLS #0:PRINT "Pluma O A":choic
pen=2:INPUT p$:IF p$="" OR p$=" " T
HEN 760
770 IF ASC(p)<48 OR ASC(p)>57 THE
N GOTO 760
780 p=VAL(p$)
790 IF p=choicepen+2 THEN 760
800 CLS #0:AND bnum>128 THEN bnum
=bnum-128
810 RETURN
815 REM *****
820 REM *** lee cadena y magnifica
***
825 REM *****
830 IF p=choicepen+2 THEN blockcol=
1:choicepen=choicepen+1
840 IF choicepen=15 THEN choicepen=12
850 IF nodisplay=1 THEN GOTO 870:RE
M #apriete no xt,yt
860 CLS #0:LOCATE #0,1,1:PRINT #0,"
xt":xt:"yt":yt
870 FOR letter=1 TO LEN(mag$)
880 IF slope=1 THEN xt=xt+B:movx
890 MOVE xt,yt
900 letter=RID$(mag$,letter,1)
910 position=ASC(letter$)
920 mem=topmem+1+(position-32)*8
930 FOR count=mem TO mem+7
940 bnum=PEEK(count)
950 IF slope=1 THEN xt=xt-movx
970 IF bnum=128 THEN GOSUB 1260:b
num=bnum-128
1000 xt=xt+mag:MOVE xt,yt
1010 IF bnum=64 THEN GOSUB 1260:b
num=bnum-64
1030 xt=xt+mag:MOVE xt,yt
1040 IF bnum=32 THEN GOSUB 1260:b
num=bnum-32
1060 xt=xt+mag:MOVE xt,yt
1070 IF bnum=16 THEN GOSUB 1260:b
num=bnum-16
1090 xt=xt+mag:MOVE xt,yt
1100 IF bnum=8 THEN GOSUB 1260:b
num=bnum-8
1120 xt=xt+mag:MOVE xt,yt
1130 IF bnum=4 THEN GOSUB 1260:b
num=bnum-4
1150 xt=xt+mag:MOVE xt,yt
1160 IF bnum=2 THEN GOSUB 1260:b
num=bnum-2
1180 xt=xt+mag:MOVE xt,yt
1190 IF bnum=1 THEN GOSUB 1260
1200 bnum=0
1210 xt=xt-(7*mag):yt=yt-mag:MOVE x
t,yt
1220 NEXT
1230 xt=xt*(B*mag):yt=yt+(B*mag)
1240 NEXT
1250 RETURN
1260 REM *** dibuja bloques ***
1265 IF blockcol=1 THEN p=(p+1) MOD
choicepen:IF p=0 THEN p=1
1270 IF p=choicepen+1 THEN random=1
1275 xl=mag-movx
1280 WHILE inc<mag
1290 IF random=1 THEN p=RND(1)+CHO
icepen:p=p%2:IF p=0 OR p>=14 THEN p=
1
1300 DRAW xl,0,p
1310 inc=inc+1
1320 MOVE xt,yt-inc
1330 WEND
1340 MOVE xt,yt:inc=0
1350 RETURN
1355 REM ***** fin de la subrutina
*****
1360 REM # eleccion de la posicion
del cursor #
1370 tes=TEC(x,y)
1380 LOCATE #0,1,2:PRINT #0,"Teclas
del cursor - Enter"

```

GIGANTES DEL BASKETBALL

Otra palabra Y/N o D para borrar?



```

650 enlarge=size*(length*8)
660 IF enlarge#B>y THEN size=size-1
:O:GOTO 650
670 CLS #0:LOCATE 1,1:PRINT "Escala
1 a ";enlarge:INPUT enlarge:IF e
nlarge="" OR enlarge="" THEN 670
680 IF ASC(enlarge)<49 OR ASC(ENLA
rge)>57 THEN GOTO 670
690 mag=VAL(enlarge)
700 xt=x:yt=y
710 IF x<2 THEN xt=(640-left-length
*mag)/2
720 rubx=xt:rubxy=yt
730 CLS #0:LOCATE #0,1,1:INPUT "Inc
linacion Y/N":y$IF y$="Y" OR y$="y
" THEN slope=1
740 IF choicepen =1 THEN GOTO 760

```

```

1390 WHILE curset=0
1400 IF INKEY(1)=0 THEN PLOT x,y,te
sites=TEST(x+movx,y):x:=+movx:PLOT
x,y,cp
1410 IF INKEY(B)=0 THEN PLOT x,y,te
sites=TEST(x-movx,y):x:=-movx:PLOT
x,y,cp
1420 IF INKEY(O)=0 THEN PLOT x,y,te
sites=TEST(x,y+2):y:=+2:PLOT x,y,cp
1430 IF INKEY(2)=0 THEN PLOT x,y,te
sites=TEST(x,y-2):y:=2:PLOT x,y,cp
1440 IF x<0 THEN x:=PLOT O,y,cp
1450 IF x>639-left THEN x:=639-left:
PLOT 639-left,y,cp
1460 IF y>398-bottom THEN y=398-bot
tom:PLOT x,398-bottom,cp
1470 IF y<1 THEN y:=1:PLOT x,1,cp
1480 IF INKEY(B)=0 THEN PLOT x,y,t
es:curset=1
1490 IF INKEY(18)=0 THEN GOTD 1490:
REM esperar hasta que la tecla ENTE
R deje de pulsarse
1500 WEND
1510 curset=0
1520 RETURN
1530 REM # entrada de datos #
1540 CALL @BB1B:IF INKEY(6)=0 THEN
GOTO 1540
1550 LOCATE #0,1,1:PRINT #,SPACE#

```



```

20)
1560 LOCATE #0,1,1:INPUT #0,"Introd
uzca numero:#num:# IF num=#" THEN G
OTO 1560
1570 IF LEN(num)<>3 THEN GOTD 1550
1580 IF VAL(num)<127 OR VAL(num)>
255 THEN GOTD 1550
1590 LOCATE #0,1,1:PRINT #,SPACE#(
20):LOCATE #0,1,1:PRINT #0,"Teclée
palabra"
1600 letters=CHR$(VAL(num))
1610 IF INKEY(B)=0 THEN GOTD 1610
1620 IF INKEY(6)=0 THEN GOTD 1610
1630 RETURN
1640 REM *** Sombra ***
1650 ytt=yt:xtt=xt
1660 IF mag<4 THEN reduce=1 ELSE re
duce=2
1670 shadow=1
1680 CLS #0:LOCATE #0,1,1:PRINT "do
s teclas de cursor"
1690 WHILE ks<1
1700 IF INKEY(O)=0 THEN SOUND 1,100
,25:yt=yt+mag\reduce:ks=ks+0,5
1710 IF INKEY(2)=0 THEN SOUND 1,100
,25:yt=yt-mag\reduce:ks=ks+0,5
1720 IF INKEY(B)=0 THEN SOUND 1,100
,25:xt=xt+mag\reduce:ks=ks+0,5
1730 IF INKEY(1)=0 THEN SOUND 1,100

```

```

,25:xt=xt-mag\reduce:ks=ks+0,5
1740 FOR n=1 TO 300:NEXT
1750 WEND
1760 TITLE hold<1:REM espera hasta
que se suelte tecla
1765 IF INKEY(O)<>0 AND INKEY(2)<>0
AND INKEY(B)<>0 AND INKEY(1)<>0 TH
EN hold=1
1770 WEND
1780 hold=0
1800 ks=0:shadxt=xt:shadyt=yt
1810 FOR n=1 TO 100:NEXT:CALL @BB1B
1815 PRINT:PRINT
1820 CLS #0:LOCATE #0,1,1:PRINT #0,
"Sombra O=":ichoicepen+2: " ":INPUT
#0,ps:IF ps=" " OR ps=# THEN GOTD
1820
1830 IF ASC(ps)<48 OR ASC(ps)>57 TH
EN GOTD 1820
1840 p=VAL(ps)
1850 GOSUB 820:REM # letras de fond
o #
1860 xtt=xt:ytt=yt:random=0
1870 RETURN
1880 REM # Salvar pantalla a memori
a #
1890 SOUND 1,100,20:MOVE 0,0
1900 CLS #0
1910 IF e=1 THEN DRAW 574,0,3:DRAW
574,350,3:DRAW 0,350,3:DRAW 0,0,3
1920 CALL 2441:REM codigo maquin
a
1930 CALL 24423
1940 LOCATE #0,1,1:INPUT #0,"Salvar
memoria Y/N":y:# IF y=#" OR y=#"
THEN GOSUB 1970
1950 CLS #0
1960 RETURN
1970 REM # salvar pantalla a cinta/
disco #
1980 SPEED WRITE 1
1990 CLS #0:LOCATE #0,1,1:PRINT #0,
"Nombre de fichero":INPUT names
2000 SAVE name$,B,MSF75,4000
2010 RETURN
2020 REM *** cargar a memoria desde
cinta/disco ***
2030 CLS #0:LOCATE #0,1,1:INPUT #0,
"Nombre de fichero":INPUT names
2040 LOAD picname$,MSF75
2050 RETURN
2060 REM *** codigo maquina para el
volcado de pantalla ***
2070 FOR n=2441 TO 24434
2080 READ x
2090 POKE n,x
2100 NEXT n
2110 RETURN
2120 DATA 1,0,64,33,0,192,17,117,95
,237,176,201
2130 DATA 1,0,64,33,117,95,17,0,192
,237,176,201
2140 REM # cargar juego de caracter
es #
2150 CLS #0:LOCATE #0,1,1:INPUT #0,
"Nombre de fichero":fil#
2160 IF LEN(fil#)>8 THEN GOTD 2150
2170 LOAD fil#,topmem-1
2180 RETURN

```



Para que los dedos no realicen el trabajo duro, H. H. AMIS TRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette manual, solicitálos.

Program Acción

```

2190 REM *** Demostracion de uso de
ntro de tus propios programas ***
2200 MODE 0:ichoicepen+15:ORIGIN 0,3
2,0,640,400,32:nodiisplay=1
2210 mag=#"Magnifica":mag=6:slope=0
:pt:=xt+104:yt=278:GOSUB 820:REM #
dibuja escala 6 en amarillo #
2220 mag=#"CARACTERES":mag=7:slope=
11:movx=4:pm=5:xt=28:yt=200:GOSUB 82
0:REM fondo
2230 mag=#"CARACTERES":mag=7:slope=
11:movx=4:pm=16:xt=32:yt=196:GOSUB 82
0:REM primer plano
2240 random=0:REM colores especial
e
2250 mag=#"por":slope=0:mag=4:xt=28
8:yt=120:pm=3:GOSUB 820
2260 mag=#"Glynnie Davies":mag=4:xt=
112:yt=60:pm=15:GOSUB 820
2270 FOR n=1 TO 500:NEXT
2280 HODE 1
2290 RETURN
2300 REM ## simple redefinidor de c
aracteres##
2310 CLS #0:LOCATE #0,1,1:INPUT #0,
"Redefine character Y/N":y:# IF y=#
"Y" OR y=#" THEN GOTD 2320 ELSE RE
TURN
2320 FOR define=1 TO 8
2330 CLS #0:LOCATE #0,1,1:PRINT #0,
"Numero #define:" ":INPUT #0,bit#
2340 IF bit=#" OR bit=#" THEN GO
TD 2330
2350 IF LEN(bit#)>3 THEN GOTD 2330
2360 IF ASC(bit#)<48 OR ASC(bit#)>5
7 THEN GOTD 2330
2370 bit=VAL(bit#):IF bit>255 THEN
GOTO 2330
2380 POKE topmem+define,bit
2390 NEXT
2400 CLS #0:INPUT #0," Y/N":y:# IF y
=#"Y" OR y=#" THEN GOTD 2410 ELSE
GOTO 2480
2410 CLS #0:INPUT #0,"Entre(CHR#)nu
mero":m#m#
2420 IF m#m#>255 OR m#m#<32 THEN
GOTO 2410
2430 m#m#=#m#m#-32
2440 FOR n=1 TO 8
2450 look=#EEK(topmem+n)
2460 POKE topmem+(m#m#n)+,look
2470 NEXT
2480 REM borrar espacio chr# 32
2490 FOR n=1 TO 8
2500 POKE topmem+n,0
2510 NEXT
2520 CALL @BB1B
2530 CLS #0:INPUT "Otro Y/N":y:# IF
y=#"Y" OR y=#" THEN GOTD 2320:REM
repetir redefinicion
2540 RETURN
2550 REM # salvar juego de caracter
es #
2560 CLS #0:LOCATE #0,1,1:INPUT #0,
"Salvar caracteres Y/N":y:# IF y=#"
" OR y=#" THEN GOTD 2570 ELSE RET
URN
2570 CLS #0:LOCATE #0,1,1:INPUT #0,
"nombre de fichero":charfil#
2580 IF LEN(charfil#)>8 THEN GOTD
2570
2590 CLS #0
2600 SPEED WRITE 1
2610 SAVE charfil$,B,topmem+1,1792
2620 CALL @BB1B
2630 RETURN

```

¡¡VUELVEN LOS AUTORES DE FRED!!

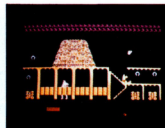
Sir Fred

SOFTWARE
ESPAÑOL

El silbido de las hojas con el viento del bosque se ve brutalmente interrumpido por la aparición de una figura solitaria entre el enramado de los arbustos. El aire cortante del norte amenaza con rasgar las vestiduras del gran caballero, SIR FRED. Su rabiosa preocupación por el rapto de la princesa, que ama no aparece reflejada en su impertérrito semblante.

Nunca vióse enfrentado a tantos y tan grandes peligros. En cada recodo, en cada estancia del gran castillo de los Beni-Gómez le, espera, acchante, la muerte.

Pero no hay ya espacio para la duda ni tiempo para echarse atrás. La decisión está ya tomada. La tierra corre, casi vuela bajo el empuje firme e implacable de sus pies...



MADE
IN
SPAIN

SOFTWARE

PARA SPECTRUM 48 K, PLUS O 128. Y AMSTRAD
(PROXIMAMENTE, COM. 64).

Pídelo a tu tienda habitual de informática o directamente
a: MADE IN SPAIN SOFT, Escuela de Informática Mr. CHIP,
Av. Cardenal Herrera Oria, 171, bajo. 28034 MADRID. Teléfono
201 64 09.

Distribuido en Inglaterra por MICRO-GEN.

BUSQUEDA BINARIA

Analisis

Frecuentemente hay que tratar en un programa con cadenas alfanuméricas, y más frecuentemente aún hay que realizar cosas con ellas para procesarlas. De estas operaciones, las más comunes son: ordenar una serie de variables de cadena según un criterio, y, una vez hecho esto, localizarla cuando sea necesaria. Analisis ya propuso en su día un algoritmo de ordenación muy sencillo y que trabaja muy bien para un número de datos de pequeño tamaño. Esta semana vamos a ocuparnos de cómo localizar una cadena determinada en una matriz alfanumérica que YA ESTA ORDENADA.

E

l método escogido se conoce como búsqueda binaria, e imita hasta cierto punto el procedimiento que una persona sigue cuando trata de localizar algo.

Si usted trata de localizar en la guía al señor Pérez, sabe que este apellido empieza por una letra tal que debe estar comprendido entre la P y la Z. No se molesta en investigar si puede estar en la zona de la guía que comienza por la letra A. Nosotros hacemos algo similar: partiendo de que nuestro array ya está ordenado, averiguamos cuál es el elemento del medio, y lo comparamos con la cadena que estamos buscando.

Pueden pasar tres cosas:

- Son iguales; lo hemos encontrado y la búsqueda termina.
- El elemento central es menor que nuestra cadena; esto implica que lo que buscamos sólo puede estar en la mitad superior del array, contando como nuevo primer elemento para la siguiente búsqueda la mitad + 1.
- Es menor, en cuyo caso estará en la mitad inferior y el último elemento para la búsqueda siguiente será la mitad - 1. Observe cómo al cumplirse **b** o **c** apartamos la mitad del array de un plumazo.

Eventualmente, o encontramos la cadena o es que no existe en el array. Sabemos que no existe porque, tras sucesivas divisiones por dos de la dimensión del array, el elemento primero se hace igual o mayor al último. En este caso, el elemento no existe y la búsqueda termina.

La descripción del programa sería así:

- Inicialización de la pantalla.
- 30-90 Declaración de variables.
- 110 Inicializa la matriz, ordenándola de paso. Esta matriz sirve sólo como prueba del algoritmo de búsqueda. No tiene aplicación práctica en un caso real.
- 120 Impresión del contenido de la matriz letra\$.

140 Creación de un bucle sin fin por el método de siempre.

150 Introducción de la palabra a buscar.

160 Esta línea permite investigar la matriz parcialmente, en lugar de buscar de la A a la Z. La razón es doble: acelerar la búsqueda y ver qué pasa cuando falla, ya que, como la matriz letra\$ tiene todas las letras del alfabeto, buscando de la A a la Z se acertaría siempre.

170 Comprobar que let1\$ y let2\$ son efectivamente letras.

180 Escogemos la primera letra de la palabra.

190 Variables intermedias para uso de la rutina de búsqueda.

200 Llamada a la rutina de búsqueda.

210 Se comprueba que la cadena existe o no mediante el valor de «mitad» devuelto por la rutina de búsqueda.

El resto del programa consta únicamente de las subrutinas mencionadas, cuyo funcionamiento es evidente excepto tal vez de la de búsqueda, comentada en el comienzo del Analisis.



INSTRUCCIONES DE ROTACION Y DESPLAZAMIENTO (I)

Esta semana vamos a hablar de una serie de instrucciones que tienen bastante en común con las que tratamos en el capítulo anterior, puesto que también se refieren al tratamiento de los bits que forman un byte.



asta ahora hemos visto cómo podemos averiguar si un cierto bit de cierto registro o posición de memoria está puesto a uno o a cero, o bien poner a uno o a cero cualquier bit que nosotros deseemos. Pues bien, las instrucciones que vamos a tratar en este capítulo nos permitirán mover de diferentes formas todos los bits que componen un byte.

Rotar el acumulador

Vamos a hablar en primer lugar de las intrucciones de rotación y desplazamiento que operan con el acumulador, o sea con el registro A. La primera de estas instrucciones se representan de la siguiente forma:

RLCA

Esta instrucción produce una rotación hacia la izquierda del acumulador. El contenido del bit 0 se mueve hacia el bit 1, el contenido previo del bit 1 se mueve hacia el bit 2, y así sucesivamente hasta llegar al bit 6. En cuanto al contenido del bit 7, éste se copia en el Flag Carry (*flag C del registro F*), y también se copia en el bit 0.

Esquemáticamente, la operación que se realiza al ejecutar dicha instrucción es:

C ← 7 ← 6 ← 5 ← 4 ← 3 ← 2 ← 1 ← 0

Vamos a ver un ejemplo práctico, y de esta forma comprenderemos

mejor cuál es el resultado de la operación. Supongamos que el contenido del acumulador es 130, y que el Flag Carry está puesto a 0. Por lo tanto antes de ejecutar la instrucción tendremos lo siguiente:

C	7	6	5	4	3	2	1	0
0	1	0	0	0	0	1	0	

Después de ejecutada la instrucción RLCA, el contenido del acumulador y del Flag Carry, será:

C	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	

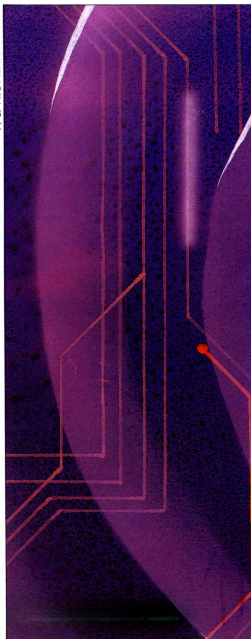
así pues en el acumulador obtendremos el valor 3, y el Flag Carry estará puesto a uno.

Programa Uno

En el primer programa que hemos preparado, podremos comprobar cómo esta instrucción de rotación, actúa sobre el Flag Carry. En primer lugar cargamos el acumulador con el valor 128, hemos puesto este valor, aunque podríamos haber puesto cualquier otro pero con la condición de que el bit 7 debe contener un 1. Luego ejecutamos la instrucción RLCA y a continuación el programa retornará al Basic puesto que el Carry está puesto a 1. Si a este programa se le suprime la instrucción RLCA, comprobaremos cómo el ordenador se cuelga irremisiblemente al ejecutar el programa, puesto que el Carry estará puesto a cero y no podrá retornar.

Programa Dos

En el segundo programa que os presentamos, queremos hacer una demostración visible de cómo actúa la instrucción que estamos estudiando.



En primer lugar, lo que hacemos es poner el ordenador en modo 2, para una mayor facilidad en el manejo de la pantalla, de eso se encargan las dos primeras líneas del programa, luego cargamos en el acumulador el valor 1 y lo cargamos en una posición de memoria visible para nosotros, o sea, una posición de memoria que corresponda a la pantalla como es la dirección #C025.

A partir de aquí el programa esperará a que pulsemos cualquier tecla para ejecutar la instrucción de rotación e imprimirá el resultado en la pantalla, y así sucesivamente hasta 7 veces, o sea que el valor final del acumulador será 128, lo que significa que el bit 7 estará puesto a uno y todos los demás bits del 0 al 6 estarán a cero.

La instrucción RLA

La siguiente instrucción de rotación de la que vamos a estudiar se representa de la siguiente forma:

RLA

es semejante a la anteriormente detallada pero con ciertos matices que la diferencian. La ejecución de esta instrucción produce una rotación a la izquierda del acumulador (*registro A*). El bit 0 se copia en el bit 1, el contenido previo del bit 1 se copia en el bit 2 y así sucesivamente. El contenido del bit 7 se copia en el Flag Carry (*Flag C del registro F*), y el contenido previo del Flag Carry se copia en el bit 0.

Vamos a ver esquemáticamente cómo afecta dicha instrucción al acumulador y al Flag Carry.

C ← 7 ← 6 ← 5 ← 4 ← 3 ← 2 ← 1 ← 0

Vamos a suponer que el contenido del acumulador y del Flag Carry sean los siguientes:

C	7	6	5	4	3	2	1	0
1	0	1	1	0	1	1	0	

Tras la ejecución de la instrucción RLA, el contenido del acumulador y del Flag Carry serán los siguientes:

C	7	6	5	4	3	2	1	0
0	1	1	0	1	1	0	1	1

así pues como podemos observar el contenido del Carry se copia en el bit 0 del acumulador.

Programa Tres

En el programa número 3 podemos ver cómo actúa esta instrucción.

En primer lugar cargamos en el acumulador el valor 1, luego no encontramos con un bucle en el que con ayuda de la instrucción BIT n,r , vista en el capítulo anterior, comprobamos si el bit 7 del acumulador está puesto a uno, de lo contrario pasa y se ejecuta la instrucción RLA, y así hasta que el bit 7 del acumulador contenga un 1, el programa muestra en pantalla con el símbolo '.' las veces que se repite el bucle.

Como es de esperar el símbolo '.' aparecerá 7 veces en pantalla, ya que hemos cargado el acumulador con el valor 1, es decir, el bit 0 del registro A está a 1 y todos los demás a 0, por lo tanto la instrucción RLA deberá ejecutarse siete veces hasta que el contenido del bit 0 pase a ocupar el bit 7.

Programa Cuatro

El programa número 4, lo que hace es mirar el contenido del Flag Carry aprovechándose de la instrucción que estamos estudiando. Así pues pasamos el contenido del Carry al bit 0 del acumulador, por lo tanto si este bit vale cero, querrá decir que no hay Carry por lo que se imprimirá una N en pantalla, si por el contrario este bit vale 1, el programa nos imprimirá una C, o sea, que hay Carry.

Instrucción RRCA

La siguiente instrucción a estudiar es igual a la vista al principio de este capítulo, con la sola diferencia de que si la primera hacia una rotación a la izquierda, ésta produce una rotación a la derecha. Esta instrucción se representa de la siguiente forma:

RRCA

y produce como hemos dicho una rotación a la derecha del acumulador. El contenido del bit 7 se copia en el bit 6, el contenido previo del bit 6 se copia en el bit 5, y así sucesivamente. El contenido del bit 0 se copia en el bit 7 y también en el Flag Carry del registro F.

Esquemáticamente dicha instrucción produce el siguiente efecto:

7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 → C

Vamos a suponer que el contenido del acumulador y del Flag Carry son los siguientes:

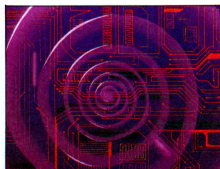
7 6 5 4 3 2 1 0 C
0 0 0 1 0 0 0 1 0

tras la ejecución de dicha instrucción obtendremos los siguientes valores:

7 6 5 4 3 2 1 0 C
1 0 0 0 1 0 0 0 1

Programa Cinco

Para ver cómo funciona esta instrucción fijémosnos en el programa 5, que es el mismo que hemos utilizado para ver visiblemente el efecto que producía la instrucción RRCA.



Ahora el lugar de ver girar el bit que tenemos en pantalla hacia la izquierda, lo veremos girar hacia la derecha por efecto de la instrucción RRCA.

Una demostración de las posibilidades de esta instrucción la tenemos en el programa número 5. En primer lugar lo que hace es poner la pantalla en modo 2 para trabajar con mayor comodidad, luego nos pinta una 'A' en pantalla y mediante un bucle hace rotar esta letra. Cada vez que pulsemos una tecla el carácter rotará un bit y así hasta que salgamos del bucle.

Esta rutina toma el primer byte que compone el carácter y lo gira un bit, luego toma el segundo byte y hace lo mismo y así sucesivamente hasta completar los 8 bytes que forman el carácter que tenemos en pantalla.

Instrucción RRA

La última instrucción que estudiamos en este capítulo es otra instrucción de rotación a la derecha que se representa de la siguiente manera:

RRA

y produce una rotación del acumulador hacia la derecha. El contenido del bit 7 se copia en el bit 0, el contenido previo del bit 6 se copia en el bit 5 y así sucesivamente. El contenido del bit 0 se copia en el Flag Carry y el contenido de éste se copia en el bit 7.

Esquemáticamente produce el siguiente efecto:

7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 → C

Vamos a ver el efecto producido suponiendo que el acumulador y en Carry tengamos los siguientes valores:

7 6 5 4 3 2 1 0 C
1 1 1 0 0 0 0 1 0

Tras la ejecución de esta instrucción obtendremos los siguientes valores:

7 6 5 4 3 2 1 0 C
0 1 1 1 0 0 0 0 1

Un ejemplo de actuación de esta instrucción lo tenemos en el programa 7. Aquí lo que hacemos únicamente es cargar el acumulador con un uno, y luego mediante RRA ponemos el Flag Carry a 1.

El programa número 8 lo que hace es mirar el contenido del bit 0 del acumulador, traspasándolo previamente al Flag Carry. Si hay Carry nos imprimirá un 1 y si no hay Carry nos imprimirá un 0.

PROGRAMAS

```

10 |PROGRAMA-1
20 |
30 | ORG #A000
A000 | TBO          LD A,12B
A002 | 07          RLCA
A003 | DB          RET C
    
```

```

10 |PROGRAMA-2
20 |
30 | ORG #A000
A000 | 3E02        LD A,2
A001 | CD8EBC     LD CALL #DCE
A002 | 0607       LD B,7
A007 | 3E01        LD A,1
A008 | F5         LD B,C
A009 | F1         LD B,C
A00A | CD18BB     LD CALL #B1B
A00B | F5         LD B,C
A00E | 3225C0     LD L,#C0251,A
A011 | 0F         LD R,C
A012 | 10F5      LD D,NZ BUC
A014 | C9         RET
    
```

```

BUC | A009
10 |PROGRAMA-3
20 |
30 | ORG #A000
A000 | 3E01        LD A,1
A002 | CBF7       50 BUC: BIT 7,A
A004 | C0         RET NZ
A005 | F5         LD A,#"
A006 | 3E2E       LD A,"
A007 | CD8A8B     LD CALL #B8A
A008 | F1         LD POP AF
A00C | 17         LD A,"
A00D | 1BF3       LD JR BUC
    
```

```

BUC | A002
10 |PROGRAMA-4
20 |
30 | ORG #A000
A000 | 17         RLA
A001 | CB47       BIT 7,A
A002 | 2009       60 JR NZ,CARRY
A005 | 3E4E       LD A,"
A007 | CD8A8B     LD CALL #B8A
A00A | C9         80 JR RET
A00B | 3E43       LD CARRY1 LD A,"
A00C | CD8A8B     LD CALL #B8A
A010 | C9         100 C9
    
```

```

CARRY | A00B
10 |PROGRAMA-5
20 |
30 | ORG #A000
A000 | 3E02        LD A,2
A001 | CD8EBC     LD CALL #DCE
A005 | 3E41       LD A,"
A007 | CD8A8B     LD CALL #B8A
A008 | 1A10       LD A,10
A00C | C5         80 OTRD: PUSH AF
A00E | CD18BB     LD CALL #B1B
A00F | 1109C0     LD HL,#C000
A010 | 1109D0     LD HL,#204B
A011 | 1109E0     LD HL,#204B
A014 | 0A08       LD B,R
A018 | 7E         LD B,(HL)
A019 | 6F         LD B,B
A01A | 77         LD HL,A
A01B | 19         ADD HL,B
A01C | 10FA      LD HL,A
A01E | C1         180 DJNZ BUC
A01F | 10EB      200 DJNZ OTRD
A021 | C9         210 RET
    
```

```

BUC | A01B OTRD A00C
10 |PROGRAMA-6
20 |
30 | ORG #A000
A000 | 3E02        LD A,2
A001 | CD8EBC     LD CALL #DCE
A005 | 0607       LD B,7
A007 | 3E80       LD A,12B
A009 | F5         LD B,C
A00A | CD18BB     LD CALL #B1B
A00B | F5         LD B,C
A00E | 3225C0     LD L,#C0251,A
A011 | 0F         LD R,C
A012 | 10F5      LD D,NZ BUC
A014 | C9         RET
    
```

```

BUC | A009
10 |PROGRAMA-7
20 |
30 | ORG #A000
A000 | 3E01        LD A,1
A002 | 1F         SET BSA
A003 | C9         RET
    
```

```

10 |PROGRAMA-8
20 |
30 | ORG #A000
A000 | 3E01        LD A,1
A002 | 1F         SET BSA
A003 | C9         RET
A004 | D410A0     LD CALL #C,UND
A007 | C9         RET
A00A | 3E31       LD A,1
A00B | CD8A8B     LD CALL #B8A
A00F | C9         LD A,1
A010 | 3E30       LD A,1
A011 | CD8A8B     LD CALL #B8A
A015 | C9         LD A,1
A016 | C9         LD A,1
A017 | C9         LD A,1
A018 | C9         LD A,1
A019 | C9         LD A,1
A020 | C9         LD A,1
A021 | C9         LD A,1
A022 | C9         LD A,1
A023 | C9         LD A,1
A024 | C9         LD A,1
A025 | C9         LD A,1
A026 | C9         LD A,1
A027 | C9         LD A,1
A028 | C9         LD A,1
A029 | C9         LD A,1
A030 | C9         LD A,1
A031 | C9         LD A,1
A032 | C9         LD A,1
A033 | C9         LD A,1
A034 | C9         LD A,1
A035 | C9         LD A,1
A036 | C9         LD A,1
A037 | C9         LD A,1
A038 | C9         LD A,1
A039 | C9         LD A,1
A040 | C9         LD A,1
A041 | C9         LD A,1
A042 | C9         LD A,1
A043 | C9         LD A,1
A044 | C9         LD A,1
A045 | C9         LD A,1
A046 | C9         LD A,1
A047 | C9         LD A,1
A048 | C9         LD A,1
A049 | C9         LD A,1
A050 | C9         LD A,1
A051 | C9         LD A,1
A052 | C9         LD A,1
A053 | C9         LD A,1
A054 | C9         LD A,1
A055 | C9         LD A,1
A056 | C9         LD A,1
A057 | C9         LD A,1
A058 | C9         LD A,1
A059 | C9         LD A,1
A060 | C9         LD A,1
A061 | C9         LD A,1
A062 | C9         LD A,1
A063 | C9         LD A,1
A064 | C9         LD A,1
A065 | C9         LD A,1
A066 | C9         LD A,1
A067 | C9         LD A,1
A068 | C9         LD A,1
A069 | C9         LD A,1
A070 | C9         LD A,1
A071 | C9         LD A,1
A072 | C9         LD A,1
A073 | C9         LD A,1
A074 | C9         LD A,1
A075 | C9         LD A,1
A076 | C9         LD A,1
A077 | C9         LD A,1
A078 | C9         LD A,1
A079 | C9         LD A,1
A080 | C9         LD A,1
A081 | C9         LD A,1
A082 | C9         LD A,1
A083 | C9         LD A,1
A084 | C9         LD A,1
A085 | C9         LD A,1
A086 | C9         LD A,1
A087 | C9         LD A,1
A088 | C9         LD A,1
A089 | C9         LD A,1
A090 | C9         LD A,1
A091 | C9         LD A,1
A092 | C9         LD A,1
A093 | C9         LD A,1
A094 | C9         LD A,1
A095 | C9         LD A,1
A096 | C9         LD A,1
A097 | C9         LD A,1
A098 | C9         LD A,1
A099 | C9         LD A,1
A100 | C9         LD A,1
    
```

MICROHOBBY

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS PARA QUE VD. NO TENGA QUE TECLEARLOS. TOTALMENTE DESPROTEGIDOS CON EL OBJETO DE FACILITAR SU COPIA EN DISCO.

Todos los programadores y aficionados a la microinformática, sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y evitar que malgastes largas horas sobre el teclado de tu ordenador tratando de descifrar incomprensiblemente mensajes de error.

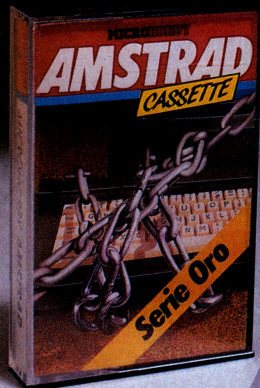
AMSTRAD SEMANAL te ofrece cada mes los programas publicados en los cuatro números correspondientes, en una cinta de cassette desprotegida, que te permitirá copiar los programas en disco y tener acceso a los listados para su estudio y posterior edición de rutinas.

Programas incluidos en la cinta número 1			
Título	Rev. n.	Título	Rev. n.
EASYDRAW	1	MAO ADDER	3
EGGRITZ	2	HEHER	3
COGIGO SECRETO	2	CHARGEN	4
YONENAS	2	PROGRAMACION	4
BIORITMOS	3		

Programas incluidos en la cinta número 2			
Título	Rev. n.	Título	Rev. n.
GRAFICOS	8	INCOGNITION	7
MUSICA	6	MONITOR	5
TECH	6	ANALISIS	5-8
ENSAMBLADOR	8	CEDRIC	8
HEXERL	8	ANIMACION1	7
TOCQUIT	8	ANIMACION2	8
PRIMEROS PASOS	7	SMALEY	7

Programas incluidos en la cinta número 3			
Título	Rev. n.	Título	Rev. n.
ANALISIS	9-12	SMITE112	11
FRUITAS	9	ANGCARD	11
MENUDISC	9	OTILO	12
RS=101	10	EVENT121	12
RS=102	10	EVENT122	12
CUATROMANA	10	FRUITES	12
SMITE111	11		

Por sólo 675 pts.
(incluidos gastos de envío)



COMPATIBLES
CON LOS MODELOS
CPC-464, CPC-664
y CPC-6128

Recíbelos en tu casa
cómodamente enviándonos
con la menor demora
posible, el cupón que se
encuentra en la última
página de la revista

CARACTERES ASCII

José Antonio García nos envía un pequeño programa cuya utilidad estriba en poder averiguar en todo momento el carácter ASCII de una tecla, por el sencillo procedimiento de pulsarla.

Títulos Musicales

El programa es una pequeña rutina o «truco» (como se le quiera llamar) para imprimir un texto «musicalmente». El texto se introduce en líneas DATA a partir de la línea 100, poniendo una coma entre cada letra, y los espacios y las comas que se quieran imprimir en pantalla deben ir entre comillas. Como pueden ver, el programa es muy sencillo pero vistoso, sobre todo para presentaciones de juegos.

El listado es el siguiente:

```
10 MODE 1
20 RESTORE 100
30 READ 1$
40 IF 1$="*" THEN END
50 S=ASC(1$)
60 PRINT 1$;
70 SOUND 1,S,5,7
80 FOR N=1 TO 100:NEXT
90 GOTO 30
100 DATA E,s,t,e," ",e,s," ",u,n,
",",e,j,e,m,p,l,o," ",d,e,i," ",f,u,
",",n,c,i,d,n,e,m,i,n,e,n,t,o," ",d,e," ",
",",l,a," ",f,u,t,i,n,a," ",",",",p,a,r,
",",e," ",M,I,C,R,O,H,O,D,B,Y," ",",",A,R,
",",S,T,R,A,D,*
```



E

l mérito consiste precisamente en esto, porque, si bien hay formas menos complejas de extraer esa información (mirar el manual, por ejemplo), no cabe duda de que son notoriamente obsoletas y, además, no nos sirven en absoluto para aprender técnicas de programación. Al fin y al cabo, para algo están los ordenadores.

Cedemos la palabra al autor del programa, que se ha tomado la molestia de enviarnos una detallada descripción del mismo, en pantalla y a mano.

Sirve para hallar el código ASCII del teclado. También lo admite con SHIFT o CAPS LOCK.

Línea	Función
10- 50	Identificación del programa.
60- 80	Condiciones del color para papel y pluma.
90-150	Instrucciones del programa.
160-	Aguarda una pulsación para iniciar el programa.
170-180	Cabeceras de resultado.
190-	Aguarda una pulsación para iniciar la ejecución.
200-	Pregunta si el ASCII es igual a 13 (ENTER) o es igual a 9 (TAB), si se pone a blancos el campo del carácter.
210-	Orden de sonido, con relación al código ASCII de la tecla pulsada.
220-	Imprime el carácter pulsado, y a continuación imprime el ASCII.

```
10 REM *****
20 REM ** ASCII **
30 REM ** 1985 POR **
40 REM ** JOSE A. GARCIA **
50 REM *****
60 PAPER 0
70 PEN 9
80 CLS
90 LOCATE 4,6:PRINT "PARA OBTENER E
L CODIGO ASCII DEL"
100 LOCATE 9,8:PRINT "TECLADO CORRE
CTAMENTE."
110 LOCATE 4,10:PRINT "FÍJESE EN EL
CARÁCTER DE REFERENCIA"
120 LOCATE 3,12:PRINT "SI SE QUIERE
LA LETRA EN MAYUSCULAS"
130 LOCATE 3,13:PRINT "Y ESTA EN M
INUSCULA O VICEVERSA PULSE "
140 LOCATE 15,17:PRINT "[CAPS LOCK]
150 LOCATE 4,23:PRINT "<PARA EJECUT
AR PULSE UNA TECLA>"
160 I$=INKEY$:IF I$="" THEN 160 ELS
E CLS
170 LOCATE 7,9:P$=" código ascii d
el teclado ":PRINT UPPER(P$)
180 PRINT:LOCATE 9,11:PRINT " CARAC
TER ":LOCATE 21,11:PRINT " NUMERO"
190 R$=INKEY$:IF R$="" THEN 190
200 IF ASC(R$)=13 OR ASC(R$)=9 THEN
LOCATE 13,13:PRINT CHR$(32);CHR$(3
2);CHR$(32) ELSE 210
210 SOUND 1,RND*ASC(R$),10,15,1
220 LOCATE 13,13:PRINT CHR$(32);R$;
CHR$(32):LOCATE 21,13:PRINT CHR$(32
);ASC(R$);CHR$(32):GOTO 190
```

O SON

SEIKOSHA ...



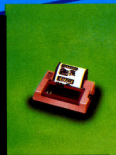
CARNAVAL SITGES



OPERA CHINA



DIOS NEPALI



SEIKOSHA GP



OPERA CHINA



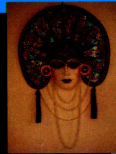
POPULAR HONG KONG



SEIKOSHA SP



POPULAR CHINA



CARNAVAL RIO



SEIKOSHA MP



CERAMICA MANTISES



OPERA JAPONESA



SEIKOSHA BP



SATTIRO



POPULAR JAPON

...O SON

MASCARAS

- GP-50 * La pequeña 40 cps. Papel normal con interface paralelo, serial y Spectrum.....17.990 ptas.
 GP-700 * La de color 50 cps. 7 colores. 80 columnas. Tracción y fricción. Papel de 10 pulgadas64.990 ptas.
 SP-1.000 * La programable 100 cps.24 cps en alta calidad 96 cart. programables en RAM. Introdutor hoja a hoja.♦.....64.990 ptas.
 SP-1.000AS La programable 100 cps.24 cps en alta calidad con interface RS-232. Introdutor hoja a hoja.♦.....59.900 ptas.
 MP-1.300AI La polivalente 300 cps, 60 cps en alta calidad, interface paralelo y RS-232. Introdutor hoja a hoja.♦&.....119.900 ptas.
 BP-5.200 * La de oficina 200 cps, 106 en alta calidad.Buffer 4K.Carro de 15".Tracción y fricción.♦.....199.900 ptas.
 BP-5.420 * La más rápida 420 cps. 106 cps en alta calidad. Buffer de 18K. Paralelo y RS-232.♦.....299.900 ptas.

Interfaces: Serie RS-232C, Spectrum, IBM, COMMODORE, MSX, QL, Apple Macintosh, HP-IB

♦ Disponen de introdutor automático de documentos opcional.

& Dispone de Kit opcional de color

Nota: I.V.A. 12%, no incluido en los precios arriba indicados

* con interface paralelo
 * con interface Spectrum

Avda. Blasco Ibañez, 116

Tel. (96) 372.88.89

Tellex 62220 - 46022 VALENCIA

Muntaner, 80-2.º-4.º

Tel. (93) 323.32.19

08011 BARCELONA

Agustín de Foxá, 25-3.º-A

Tels. (91) 733.57.00-733.56.50

28036 MADRID

DiRac

Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentran en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todas.

Poseo el libro *Soft 158* que describe el firmware que viene con el 464, pero no he encontrado ningún otro libro que facilite más información sobre el —664— (que es el ordenador que yo poseo) más concretamente sobre el firmware que controla la unidad de microfloppy, ya que el resto del firmware es prácticamente idéntico en los dos modelos.

Tampoco he encontrado ninguna publicación sobre el hardware del 664, tema éste que me interesa muchísimo.

Así que, si alguien me sabe orientar en este sentido y darme alguna pista sobre cómo localizar el «CPC 664 Service Manual o algún libro similar» le estaré eternamente agradecido.

Juan Antonio Vázquez (Cádiz)

Como habrás podido observar, el ordenador es capaz de detectar por sí mismo si la unidad de entrada/salida por defecto en un momento dado en la cinta o el disco; es decir, si tú no le das la orden "ITAPE" o similar, cualquier operación de lectura/escritura de ficheros se realizará en el disco.

Esto implica que las mismas rutinas de lectura/escritura que rigen el cassette funcionarán exactamente igual con el disco, siempre que no corrompas la zona de memoria del firmware o actives expresamente el cassette.

Una forma muy sencilla de comprobar lo que te decimos, es llamar a la rutina firmware que produce un directorio respetando los parámetros que deben suministrarsele. Si el disco está operativo, verás en la pantalla una salida equivalente al comando Basic CAT.

A) ¿Las cintas que anunciáis con todos los programas publicados en el mes, la vais a poner a la venta en los quioscos juntamente con la revista una vez al mes, o sólo las vais a vender solicitándola?

B) ¿En caso de enviarnos un programa grabado en cinta como indicáis, devolveréis la cinta? Si lo hicierais sería de agradecer, pues algunas están por las nubes y las baratas son un rosario de Read errors.

C) ¿Pueden utilizarse cintas sin fin de modo de microdrive en el 464 y 664?

D) ¿Qué a pasado con las posibilidades de ampliación de RAM que tan profusamente remarcaba la primera publicidad de Amstrad y que ha caído en el olvido últimamente?

Fernando Urruti (Sestao)

A) *La cinta que contiene los programas publicados en los cuatro números de cada mes no se venderá en quioscos. Tienes que pedirla por correo.*

B) *En principio, las cintas no se devuelven a menos que en la carta que nos mandes envíes los sellos de correos necesarios para tal efecto. Si deseas recogerla, está a tu disposición durante un plazo no superior a dos meses a partir de la fecha en que tu programa llegue a nuestra redacción.*

C) *No existen ese tipo de periféricos para los Amstrad. No son necesarios debido a la unidad de disco.*

D) *Las posibilidades de ampliación de la publicidad realmente se refieren a memoria ROM, no RAM. Por razones muy obvias, se evitó dejar ese punto lo suficientemente claro.*

1.º ¿Son compatibles todos los programas que vienen en las cintas "SERIE ORO" con el CPC 464?

2.º Al hacer el programa de "BIORRITMOS" me da "SYNTAX

ERROR" en las líneas 760-840 y 210. He comprobado dichas líneas y no encuentro ningún error, es decir, están igual que vienen en el listado. ¿Me podríais indicar dónde reside el error?

3.º Existen algunos comandos en el 664 que no lo admite el 464, pero ¿se pueden suprimir éstos por otros que sí los admita el 464?

Juan José Morales (Málaga)

1.º *A menos que se indique explícitamente lo contrario, sí.*

2.º *Este problema, como comentamos en su día en esta misma sección, se debe a la forma en la que el Amstrad evalúa las funciones definidas por el usuario mediante la orden DEF FN.*

Estas funciones son atesoradas en un lugar especial de la memoria, pero el intérprete Basic sólo las estudia cuando la función es invocada desde alguna parte del programa. Si en la definición anteriormente hecha de la función había un error de sintaxis, el ordenador informará de ello, pero dirá que el error ocurre en la línea donde la función es llamada, que normalmente estará perfectamente correcta. Este fallo del intérprete Basic, ha dado en este programa y sin duda dará en otros, lugar a muchas confusiones de este tipo.

Amstrad Ideas

AMSTRAD Semanal comunica a todos sus lectores la apertura de una nueva sección dedicada a recoger las mejores ideas que exploten al máximo las posibilidades del ordenador, materializadas en programas claros y cortos (máximo 25 líneas). Los mejores de entre todos ellos serán publicados con el nombre de su autor en la revista, recibiendo como premio, gratuitamente en su domicilio los cuatro primeros números de nuestra cinta mensual. Los programas enviados deberán incluir:

- Cinta de cassette con el programa o programas grabados.
- Explicación detallada del funcionamiento y propósito del programa, mecanografiado a 2 espacios o con letra clara.

Es imprescindible indicar en el sobre claramente: **AMSTRAD IDEAS.**

La dirección es:

Hobby Press, S. A.

La Granja, s/n.

Polígono Industrial de Alcobendas.
Madrid

¡GUERRA al I.V.A.!

Suscríbete ahora a

**MICROHOBBY
AMSTRAD**

y benefíciate de un

26%

DESCUENTO

(Oferta válida sólo hasta el 28 de febrero de 1986)

Microhobby AMSTRAD te ofrece ahora una oportunidad excepcional. Hasta el 28 de febrero de 1986 podrás suscribirte a nuestra revista sin tener que pagar el recargo correspondiente al impuesto sobre el Valor Añadido.

Microhobby AMSTRAD lo abona por ti. Ahora puedes recibir Microhobby AMSTRAD en tu domicilio durante todo un año por sólo 5.900 ptas., es decir, 2.400 ptas. menos de su valor real.

¡APROVECHA ESTA OPORTUNIDAD!

RECORTA O COPIA ESTE CUPON Y ENVIALO A HOBBY PRESS, S.A. APDO. DE CORREOS 232 ALCOBENDAS (MADRID)

Deseo suscribirme a **Microhobby AMSTRAD** durante un año (50 números) por **sólo 5.900 ptas.**, lo que me supone **un ahorro de 2.100 pesetas.**

El primer número que deseo recibir es el _____

NOMBRE _____ EDAD _____

APELLIDOS _____

DOMICILIO _____

CIUDAD _____ PROVINCIA _____

C. POSTAL _____ TELEFONO _____ PROFESION _____

Marco con una (x) en el casillero correspondiente la forma de pago que más me conviene.

Talón bancario adjunto a nombre de HOBBY PRESS, S. A.

Giro Postal a nombre de HOBBY PRESS, S. A., N.º _____

Contra reembolso del primer envío

VISA N.º _____

Fecha de caducidad de la tarjeta _____

Domiciliación bancaria (50 NUMEROS MAS 1 DE REGALO)

Banco _____ Sucursal _____ Localidad _____

N.º de cuenta _____

Si lo prefiere puede
suscribirse por teléfono:
(91) 654 28 98

Firma y fecha

Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMÚN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMÚN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A.**
AMSTRAD SEMANAL.

Apartado de correos 54.062
28080 MADRID

¡ABSTENERSE PIRATAS!

Amstrad 664/6128 desearía intercambio de programas en disco o cinta, utilidades o juegos. Manuel Díaz Fernández. C/ Foncalada, 11 - 1.º dcha. 33002 Oviedo. Tel. 21 14 17.

Vendo un cassette o cambio por otro. Cambio «El almirante Graff Spee» por «combat Linx» o «Pole position» o «One on one» o «Kong Strikes Back». Mandar a Ricardo Garcías. C/ Cala Estancia, 5 Bajos. 07007 Palma de Mallorca (Balears).

Vendo ordenador **Amstrad 464** con pantalla de fósforo verde. Tengo más de 50 programas y varios libros. Regalo joystick y modulador. Intercambio programas del **Amstrad**. Preguntar por José Luis Soriano. C/ Reus, 6-1. 46009 Valencia. Tel. 349 95 48.

Por necesidades económicas vendo ordenador **Amstrad CPC-664** con unidad disco 3" incorp., monitor Color 14" y progr. CP/M, Logo, Base Datos, Proces. Textos, Taspint, Tascopy y 6 progrs. más, todos en discos. Comprado el 5-10-85 con garantía oficial. Precio mínimo 98.000 pts. al contado (aceptaré la mejor oferta). Urge venta. A. Moreno. C/ Riera Alta, 43, at. 3.º 08001 Barcelona.

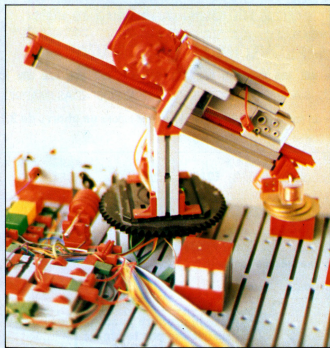
Me gustaría intercambiar programas, juegos, ideas, etc., con usuarios del CPC 464 que vivan en Burgos. Llamar por la tarde al Tel. 26 06 89. Preguntar por Javier.

NL New Line
GABINETE DE INFORMÁTICA

- **Clases de Informática sobre AMSTRAD**
En grupos o individuales
- **Ordenadores AMSTRAD y periféricos**
Los mejores precios
- **Software a la medida**

ZURBANO, 4 ☎ 410 47 63
28010 MADRID

MASTER COMPUTER



Si no lo encuentras en tu tienda habitual, llámanos y te lo enviaremos directamente contra reembolso.

Tenemos todos los modelos de Amstrad, periféricos, software y libros.

Disco o/y cassette Amstrad
Commodore
Apple

Robot
Fischertechnik

Distribuidor para España
precio incluyendo caja de construcción

software
interface
adaptador Todo.— **34.990** ptas.

Centro Comercial, Local 15
Ciudad Sto. Domingo
Carretera de Burgos, Km 28
ALGETE - MADRID
Tel. 622 12 89

SACALE EL JUGO A TU ORDENADOR. DISEÑA TUS PROPIAS PANTALLAS Y DIVIERTETE JUGANDO CON...

Este mes:

YOUR COMPUTER

Te ofrece algo realmente sabroso:

MUSICA

Este magnífico programa escrito en código máquina te permitirá manejar el sonido y la música en tu Amstrad desde Basic, mediante un nuevo juego de comandos creados especialmente para ello.

CROSS

Tienes cuatro revólveres para destruir a tus enemigos en el mínimo tiempo posible. Necesitarás toda tu habilidad, rapidez de reflejos y suerte, mucha suerte.

JUMPER

Debes alcanzar la cima del Valle de las Cintas Deslizantes. Tienes que saltar por los huecos de las vallas, que se desplazarán a derecha e izquierda con una rapidez de vértigo.

MAGGOT

Te encuentras en la amable tierra de las setas gigantes. Tu misión es guardarla del ataque y la invasión de una peligrosísima serpiente polimórfica que las ataca sin piedad.

TIMEBOMB

Una organización terrorista de Oriente Medio ha colocado una bomba de tiempo en el laberinto de defensa del Laboratorio de investigación bacteriológica de Lexington.

RSX

Your Computer ha pensado en los usuarios del Amstrad CPC464 y ha creado un nuevo juego de comandos completo para tu ordenador, de forma que el Basic así ampliado no tenga nada que envidiar al de los otros modelos de la serie.

SINTAX, S.A.

Si no lo encontrara en su kiosco, puede solicitarlo directamente a nuestra editorial:
Paseo de la Castellana, 268. Tel.: (91) 733 25 99. 28046 Madrid.

2 **YOUR COMPUTER**
EL CORAZÓN DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES

AMSTRAD
464-664-6128

La mejor selección de programas de juegos y utilidades, publicados en la revista de mayor difusión de Europa en ordenadores. Ahora reproducidos en cassette, en auténtica exclusiva mundial.

695.-
PTAS.

2 **YOUR COMPUTER**
EL CORAZÓN DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES
AMSTRAD
464-664-6128

1 Cross
2 Jumper
3 Maggot
4 Timehomb
5 A lósis
6 RSX

Las cifras de Your Computer se editan mensualmente para Amstrad, S.A. en su ordenador Commodore.

ESTA A VENTA EN TU KIOSCO!

¡GANA UN 128 K!
Total garantía de carga

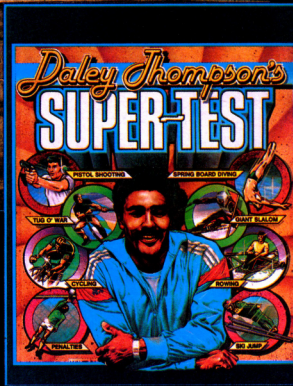
SI BUSCAS LO MEJOR **ERBE** Software LO TIENE

TODO EL DEPORTE EN TU ORDENADOR

NATACION, TIRO AL PLATO, SALTO DE POTRO,
LEVANTAMIENTO DE PESO,
TIRO AL ARCO, TRIPLE SALTO



**PENALTIES, CICLISMO, TIRO DE PISTOLA,
REMO, SALTO DE ESQUI, SLALOM GIGANTE
TIRO DE CUERDA, SALTO DE TRAMPOLIN**



DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE, SANTA ENGRACIA, 17. Tel. 447 34 10.
DELEGACION BARCELONA, Avd. MISTRAL, 10. Tel. (93) 432 07 31