

MICROHOBBY**AMSTRAD***Semanal*

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO II N.º 36

160 Ptas.

Canarias 165 pts.

EL ORDEN DE
LAS COSAS ES DBASEII.
UNA SUPERBASE
DE DATOS

**ATENCION
A TUS
PRIMEROS
PASOS EN
GRAFICOS**

COPYCAT:

Cinta-disco,
disco-cinta
en lenguaje
máquina

SOFTWARE

Salva a Sweevo
de un mundo de
Fantasía Infernal



LO NUESTRO ES HACER BUENAS GESTIONES



PARA
AMSTRAD, PC Y COMPATIBLES

¡ No estamos para juegos !

DEMOSTRACIONES DE NUESTRO SOFTWARE
COMERCIAL Y DE GESTION
EN NUESTRO STAND



les esperamos en

PRECIOS ESPECIALES 1.ª FERIA AMSTRAD

FACTURACION - Sólo teclee un código y salen todos los datos del cliente. Numeración correlativa automática. Admite 30 productos distintos por factura. Admite 30 productos distintos por factura. Automáticos, descuentos, cargos, IVA. Proporciona 5 totales por factura. (PVP 15.300 incl. IVA)

PRESUPUESTOS - Guarda en memoria los presupuestos y extiende las facturas. Conceptos de 200 caracteres cada uno (3 renglones de escritura) (PVP 18.300 incl. IVA)

CUENTAS - PROVEEDORES, BANCOS, CLIENTES - 3 ficheros separados. Resúmenes totales, unitarios o parciales. El mejor auxiliar de CONTABILIDAD al día. (PVP 8.600 incl. IVA)

CONTROL DE ALMACEN IVA - Código de 9 dígitos alfanuméricos. 25 dígitos denominación. Una sola pantalla entradas y salidas, con visión de asientos anteriores. Stocks máximo, mínimo y aviso para reposicionamiento. Totales entradas y salidas cada pantalla (PVP 15.300 incl. IVA)

CLIENTES (con etiquetas) - 11 campos distintos para localización. Etiquetas 4 modelos distintos en salida de dos. El más fiel auxiliar ahorrador de tiempo. (PVP 8.600 incl. IVA)

RECIBOS - Resuelve el problema interminable a asociaciones, comunidades, colegios. Fijos los campos del normalizado y 12 campos libres (4 numéricos con cálculos automáticos). Liquidaciones bancos. (PVP 18.300 incl. IVA) Con numeración automática (21.200 incl. IVA).

RESTAURANTES - Tratamiento de minuto y facturas. Resúmenes por grupos. Mesas abiertas permanentemente, correcciones, cambios, etc. hasta emisión fro. final. (PVP 35.000 incl. IVA)

IVA POR ALMACEN - Rellena liquidaciones Hacienda. Introduce cuentas IVA gastos. (PVP 18.900 incl. IVA)

URBANIZACIONES - Lectura y tratamiento de contadores consumos. (agua, gas, luz, etc) Extensión recibos y totalizaciones bancos. Emisión etiquetas.

LIBROS DEL IVA - Controles de repercutado y soporte orden numérico. Resúmenes estudios comparativos. Rellena liquidación Hacienda. (PVP 16.800 incl. IVA)

ADMINISTRACION DE FINCAS - Gestión completa profesionales. Sencillo manejo cualquier persona (PVP 40.000 incl. IVA)

FACTURACION Y ALMACEN - Gestión unida. Ficheros clientes, producto, descuentos y cargos. Todos los resúmenes. (PVP 18.900 incl. IVA)

1 AÑO DE GARANTIA

NUESTRO EQUIPO PROFESIONAL
PARA CUALQUIER MODIFICACION QUE
UD. INDIQUE EN LOS PROGRAMAS, A
UN PRECIO MODICO

Llamar o contactar con Juan Luis Ruiz

PEDIDOS, TELEFONO, CARTA O TELEX
REEMBOLSO SIN GASTOS.

ESPECIAL A COLABORADORES
RESTO DE ESPAÑA



**informática
GROTUR, S.A.**

C/ JAIME EL CONQUISTADOR, 27
28045 MADRID Tno. 474 55 00
474 55 32
Télex: IGSA 48452

AMSTRAD

sumario

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.º Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

José Flores

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sapuerta

Robert Chatwin

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Portada

M. Borco

Ilustradores

J. Igual, J. Pons, F. L. Frontón,

J. Septien, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

María Andriño

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

Marta García

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortés

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

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.º Rosa González

M.º del Mar Calzada

Redacción, Administración**y Publicidad**

La Granja, 39

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Paulino Blanco

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDIC, S. A. Crta. de Irún,

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solano, 16

Depósito 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

América 1.532. Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

Año II • Número 36 • 6 al 12 de Mayo de 1986
160 ptas. (incluido I.V.A.)
Camarias, 155 ptas. + 10 ptas. sobretasa aérea.
Ceuta y Melilla, 155 ptas.

5 Primera plana

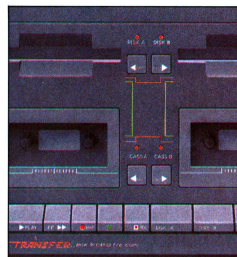
Ha salido el especial de Micromanía.

8 Para... PCW

Probamos Dbase II, la base de datos más legendaria del mundo, y probablemente la mejor que existe hoy para Amstrad.

Código máquina 18

Una increíble utilidad para pasar programas entre disco y cinta de casi todas las maneras posibles.



24 Mr. Joystick

La aventura tridimensional, pixel a pixel, del infeliz Sweeve.

Primeros pasos 26

Cómo se define y representa gráficamente un carácter ASCII creado por el usuario.

32 Serie Oro

Todo lo que quieras saber y poder hacer con los números complejos, a partir de ahora tu Amstrad te lo pone fácil.

ProgramAcción 40

Último artículo de la serie del manejo de sonido en Basic que cierra con «broche de oro» un capítulo más de ProgramAcción.

AVISO A NUESTROS LECTORES

Comunicamos a todos nuestros lectores que desde el 1 de mayo nuestra dirección y teléfonos son los siguientes:
HOBBY PRESS, S.A.
Ctra. de Irún km 12,400 (Fuencarral). 28049 MADRID
Teléfonos:
Suscripciones: 734 65 00
Redacción: 734 70 12

MICRO-1

el IVA lo paga
MICRO-1

C/ Duque de Sesto, 50. 28009 Madrid

Tel.: (91) 275 96 16/274 53 80

(Metro O'Donnell o Goya)

Aparcamiento gratuito en Felipe II

SOFTWARE: ¡¡2 PROGRAMAS POR EL PRECIO DE 1!!

Y además, completamente gratis, un magnifico reloj de cuarzo. Increíble ¿verdad?

	Ptas.
PING PONG	2.295
SABOTEUR	2.295
RAMBO	2.295
YIEAR KUNG FU	2.295
WORLD SERIES BASEBALL	2.095
MAPGAME	2.750
RAID	2.295
HYPERSPORTS	2.295
HIGHWAY ENCOUNTER	1.750
HIGHWAY ENCOUNTER DISCO	3.300
ALIEN B	1.750

	Ptas.
DYNAMITE DAN	2.100
SABRE WULF	1.650
THEY SOLD A MILLION	2.500
FIGHTER PILOT	1.975
MASTER OF T. LAMP	1.950
NIGHT SHADE	1.950
HACKER	1.950
SUPER TEST	2.300
TORNADO LOW LEVEL DISCO	3.300
TORNADO LOW LEVEL	1.750
KNIGHT LORE	1.750

SOFTWARE DE REGALO: ¡¡OFERTA 2 x 1!!

Beach Head Decathlon Dummy Run Beach Head Southern Belle

Fabulosos
precios para tu Amstrad
CPC-464 CPC6128
PCW-8256 y
PCW-512

SOFTWARE DE GESTION PROFESIONAL

DBA II	17.800	DR. GRAPH	15.100
CBASIC	15.100	CONTABILIDAD	
DR DRAW	15.100	Y VTOS.	16.600

IMPRESORAS ¡¡20% DTO. SOBRE P.V.P.!!

COMPATIBLE IBM PC-XT 256 K
Y DOS DISKETTES DE 360 K
229.900 PTAS.

UNIDAD DE DISCO 5¼"
PARA AMSTRAD
34.900 PTAS.

LAPIZ OPTICO+INTERFACE
3.495 PTAS.

CINTA VIRGEN ESPECIAL ORDENADORES
69 PTAS.

SINTETIZADOR DE VOZ EN
CASTELLANO
15% DTO.
CASSETTE ESPECIAL ORDENADOR
5.295 PTAS.

JOSTICK QUICK SHOT II
1.995 PTAS.
JOYSTICK QUICK SHOT V
2.295 PTAS.
con la compra de un joystick
¡¡GRATIS 1 RELOJ DE CUARZO!!

DISKETTE 5¼"
295 PTAS.

DISKETTE 3"
990 PTAS.

«MARK II LIGHT PEN»

Tenemos en el mercado otro lápiz óptico, que viene en un paquete que incluye el periférico propiamente dicho junto con el software necesario para hacerlo funcionar.

El aparato, creado por «Electric Studio Products», está preparado para correr en todos los Amstrad CPC.

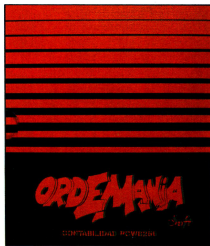
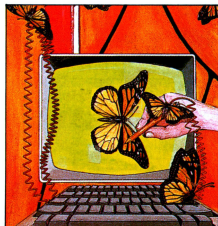
Existe, por tanto, en versiones de disco y cassette a los siguientes precios:

CPC 464 CASSETTE	4.900 ptas
CPC 464-664 DISCO	6.900 ptas
CPC 6128 DISCO	6.900 ptas

A primera vista, y a la espera de un veredicto más riguroso proveniente de un Banco de pruebas, el paquete, lápiz y soft, tiene bastante buena pinta. Parece ser aceptablemente sensible y su manejo es muy sencillo y amigable, regido por un menú de iconos. Sus capacidades gráficas son, en principio, muy buenas, según lo que los miembros de la redacción de **AMSTRAD** han podido observar.

La distribución exclusiva en España de los productos de la casa inglesa la realiza Ofites Informática, cuya dirección es la siguiente:

Avda. Isabel II, 16-B
Tel. 45 55 44 - 45 55 33
20011 San Sebastián.



Más y más para el PCW 8256

Una nueva compañía de Software, Ordemanía Soft, se ha incorporado al cada vez más competitivo mundo de los programas de gestión para el PCW 8256, con un programa de contabilidad adaptado al plan Contable Español, contemplando el asunto del IVA.

Se trata de una creación muy completa que esperamos tener muy pronto oportunidad de analizar exhaustivamente en la sección «Para... PCW».

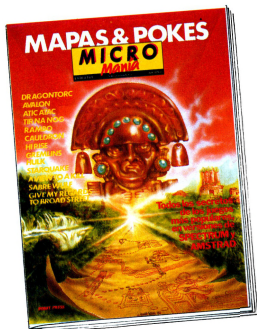
Aprovechamos la coyuntura para agradecer desde aquí a nuestros lectores la favorable opinión que «Para... PCW» les merece, y que han tenido el detalle de hacérselo saber por carta o teléfono.

Ordemanía Soft está en:
C/ Torres Quevedo, 34
02003 Albacete
Tel. (967) - 22 79 44



Corta y pega este cupón en la casilla correspondiente de la página 16 del número 31 de **AMSTRAD Semanal**, una vez completada la página, envíanosla junto con tus datos. ¡SUERTE!

Primera PLANA



En el mes de mayo

MICROMANIA LANZA UN NUMERO «MUY ESPECIAL»

MICROMANIA, nuestra hermana mayor que como sabéis se dedica mes a mes a analizar los mejores juegos para todos los ordenadores, va a lanzar el próximo mes de mayo un especial, al precio de 400 ptas., dedicado por entero a una de las secciones que más le caracterizan: **Patas Arriba**.

Pokes y **mapas** a todo color de los juegos más populares, **Rambo**, **Starquake**, **Cauldron**, **Atic Atac**, **Tir Na Nog**... y así hasta un total de 13, se desarrollan a lo largo de 84 páginas con un gran despliegue de fotos e instrucciones para resolverlos más fácilmente.

Ofites

Presenta: el universo del software,

DELTA
+

La más moderna base de datos DELTA, superándose a sí misma, "DELTA +", desarrollada para CP/M por COMPSOFT con todo en español.

Diseña sus propios ficheros; desde un simple fichero de nombres y direcciones hasta su propio sistema contable. El formato estándar DIF permite intercambiar datos en DELTA, desde las hojas de cálculo CRACKER II, etc... y viceversa. Intercambio de datos con la mayoría de los tratamientos de texto como NEWWORD para MAILING.

Incluye un sencillo y funcional sistema de impresión de etiquetas con: hasta 5 columnas de etiquetas, 65 caracteres por etiquetas, 20 líneas con 3 campos cada una.

- PROGRAMABLE Y RELACIONAL.
- FICHEROS INDEXADOS.
- HASTA 90 CAMPOS ó 2.000 CARACTERES.
- MÚLTIPLES SISTEMAS DE BUSQUEDA, 8 CLAVES.
- FICHEROS DE HASTA 8 Mb.
- 8 GRUPOS DE TRANSACCION POR REGISTRO.

BASE
DE DATOS

17.850 pts.

NEWWORD

Programa de tratamiento de textos mejorando todo lo anterior. Manual y programa en español, que le enseñarán con facilidad y rapidez lo más avanzado en procesadores de textos. Compatibilidad funcional con WORDSTAR incluyendo muchas capacidades adicionales.

Tiene un potente MAIL-MERGE con opción de selección de destinatarios por criterios base de datos, creación de documentos, impresión de etiquetas. Utiliza todo el espacio de disco. Ensamblaje de textos, sustitución, etc., de la forma más fácil: autohace copias de seguridad. ¡NUNCA PERDERA UN TEXTO!

- Ñ, ACENTOS, DIERESIS, ETC...
- PRESENTACION EXACTA EN PANTALLA DEL FUTURO DOCUMENTO IMPRESO.
- INTERCAMBIOS DE FICHEROS CON CRACKER.
- VARIABLES SUSTITUIBLES EN IMPRESORA.
- POTENTE CALCULADORA.
- COMPROBADOR ORTOGRAFICO Y GRAN DICCIONARIO (45.000 TERMINOS AMPLIABLES).
- POSIBILIDAD DE LECTURA DE FICHEROS DE DELTA, CARD BOX, SUPERCALC, DBASE II, ETC...

TRATAMIENTO
DE TEXTOS

17.850 pts.

CRACKER II

El CRACK de las hojas de cálculo, la que deja detrás al resto. Funciones nunca vistas, formato de fechas, salvaguardia continua sobre un fichero. Realiza automáticamente copias de seguridad. Además de las tradicionales funciones, CRACKER II posee funciones lógicas, estadísticas y de alta matemática. Interchangea datos con NEWWORD, bases de datos y la mayoría de las hojas de cálculo.

- CELDAS PROGRAMABLES.
- FUNCIONES ESPECIALES: Fecha, días; desde y hasta la fecha de la semana, del año, lapso de tiempo, retraso, beep entrada, saludo usuario.
- SISTEMA DE AYUDA ON-LINE.
- SUMA CONDICIONAL.
- TOMAR DECISIONES EN LA HOJA.
- 18 MODOS GRAFICOS DISTINTOS.
- TRADICIONALES FUNCIONES MATEMATICAS Y AMPLIACION, FUNCIONES ESTADISTICAS Y LOGICAS.
- GENERA GRAFICOS EN BASE A LOS DATOS.

HOJA
DE CALCULO

17.850 pts.

EDITOR Y DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA

IVA
no
incluido

TOTALMENTE
EN
ESPAÑOL

Informática

estas son sus estrellas.

NUCLEUS

NUCLEUS más que una estrella una constelación; tres ESTRELLAS en un SUPERPROGRAMA, la solución a cualquier aplicación por compleja que sea, NUCLEUS es GENERADOR DE PROGRAMAS, BASE DE DATOS Y GENERADOR DE INFORMES.

Toda la información es multi-intercambiable y de libre acceso por cualquiera de los demás programas. Así los datos de la base los condicionamos y utilizamos en el generador de programas y los imprimimos a través del generador de informes.

- GENERADOR DE PROGRAMAS EN MALLARD BASIC.
- CREACION DE BASES DE DATOS RELACIONALES.
- GENERADOR DE INFORMES.
- DISEÑADOR DE FORMATOS.
- DISEÑADOR DE PANTALLAS.
- CODIGO FUENTE DE LIBRE ACCESO Y LIBRE DE ERROR.
- DISEÑA SU PROPIO SISTEMA.
- MAILMERGE.

GENERADOR
DE PROGRAMAS

26.780 pts.

BRAINSTORM

La revolución del pensamiento, BRAINSTORM es un programa que piensa con Vd.

El compañero ideal para el empresario, director o cualquier persona que tenga que planificarse o tomar decisiones.

BRAINSTORM es la ayuda necesaria para su organización. El programa que se ha standarizado en Inglaterra, tan necesario, útil y popular como una base de datos o un tratamiento de textos.

- ORGANIZA POR RANGOS.
- ACCESO DESCENDENTE PROMEMORIZADO.
- PLANIFICACION A NIVEL DIA.
- DECISIONES A LARGO PLAZO.
- REVISION DE PROBLEMAS.
- SIMULTANEIZACION DE TAREAS.
- PROCESO TOP/DOWN.

ORGANIZADOR
DE IDEAS

17.850 pts.

STARCOM

Piii... su ordenador le comunica:

La revolución de las comunicaciones, de la mano de OFITES INFORMÁTICA, llega a España. El nuevo mundo de las comunicaciones digitales lo tiene a su disposición, las redes de transmisión electrónica digitalizada, con su PCW 8256 o PCW 8512a través de un interface RS 232-C con otros ordenadores, redes de transmisión de datos, etc..., Vd. podrá enviar o recibir ficheros de texto o de datos, ASCII, etc..., creados por NEWWORD y otros...

- TRANSICIONES DIRECTAS EN RED.
- COMPATIBILIDAD CON NEWWORD.
- POSIBILIDADES DE TRANSMISIONES VIA MODEM, RED TELEFONICA.
- COMUNICACION INSTANTANEA.

COMUNICACIONES

17.850 pts.

DE VENTA EN LOS MEJORES COMERCIOS DE INFORMÁTICA

Si Vd. tiene alguna dificultad para obtener los programas, puede dirigirse a:



Avda. Isabel II, 16 - 8º
Tels. 455544 - 455533
Telex 36698
20011 SAN SEBASTIAN

CONDICIONES ESPECIALES PARA DISTRIBUIDORES

DATA BASE II

Autor: **Fco. Javier Barceló T**

En el número 33 de esta revista, se hizo referencia a una base de datos para los CPC. Quedó explicado lo que era una base de datos, y cómo manejaba todo tipo de ficheros con una gran facilidad. Al aumentar la memoria de los ordenadores, las bases de datos se han ido haciendo más complejas, y han ido incorporando más y más funciones, hasta llegar al programa que hoy analiza MICROHOBBY AMSTRAD.



Data Base II, es otro programa clásico en los ordenadores personales. Pero éste no es un simple programa de base de datos. Más que eso, se podría decir que es un pequeño lenguaje de programación, con una finalidad única: desarrollar programas que gestionen uno o más ficheros de datos.

La primera impresión que da DBASE II es realmente buena. El libro de instrucciones es una voluminosa carpeta, estructurada en dos partes, y, ¡SORPRESA!, en español. La primera parte de este libro es una introducción al manejo del programa, que explica poco a poco, las principales características del programa. La segunda parte, explica el uso avanzado del programa, y da un glosario de todas las posibilidades del mismo. No obstante, y éste quizá sea el único «pero» que se pueda poner al paquete, el manual deja poco claras algunas de sus características. La creación y manejo de una base de datos es fácil, y ofrece pocos problemas, pero la combinación de esto con las otras herramientas que proporciona ya complica el asunto un tanto, siendo necesarias varias horas de práctica para alcanzar un dominio aceptable de las posibilidades que tiene. Que no son pocas...

A la hora de examinar el programa, se pueden dividir las facultades de DBASE II en dos partes: La primera consistirá en la creación y el manejo de una base de datos o fichero. La segunda, examinar el resto de las facultades que proporciona el programa y el resultado de combinarlas con las anteriores. Esto es, un programa.

MANEJO DE UNA BASE DE DATOS

Cuando se carga el programa, lo primero que hace es pedir la fecha. Este dato es opcional, pero es conveniente darlo. Al hacerlo, cada vez que se modifica el fichero, el programa graba la fecha dada, con lo que permite saber en todo momento el último día que éste se modificó. Pero además, al hacer los listados saldrá también la fecha.

Después de la fecha, aparece un punto a la izquierda de la primera línea libre. Esto es la línea de comandos. DBASE II reconoce los comandos a través de las cuatro primeras letras tecladas, de modo que no hace falta escribirlos enteros.

Si el programa detecta un error de sintaxis, lo advierte y pregunta si se desea corregir la orden. En caso afirmativo, pregunta:

— CHANGE FROM.: Teclar la parte incorrecta.

— CHANGE TO...: Teclar correctamente. En el cuadro de control del cursor, aparece



cómo salir de lo que se esté haciendo, a la línea de comandos. Para salir del programa, se usa el comando QUIT.

1. Creación de una base de datos

El primer mandato para crear un fichero es CREATE. Este crea la base de datos, en base a las siguientes preguntas:

- Nombre del fichero.
- Estructura.

La definición de la estructura es de especial importancia, dado que aunque se pueda modificar mediante el comando MODIFY, esto borra todos los registros que haya grabado anteriormente.

En la estructura, hay que introducir primero el nombre del campo, luego su tipo, su longitud y número de decimales. Admite tres tipos de campos: alfanumérico, numérico y lógico. Este último es para aceptar expresiones lógicas de tipo: Verdadero o Falso. Si un campo, aunque sea numérico, no va a exigir ningún tipo de operación matemática, es preferible ponerlo como alfanumérico, debido a ciertas operaciones que son posibles con los campos numéricos, como totalizar los campos numéricos de los registros. (Así evitamos que añada al total de ellos el número del D.N.I. o el C.I.F. etc.). En los campos numéricos se pueden definir las posiciones decimales, pero siempre teniendo en cuenta que hay que contar entre ellas el punto decimal.

El nombre puede constar de hasta 10 caracteres, incluyendo el signo (:), sin espacios intermedios. En caso de error, DBASE II lo detecta y pide que se introduzca el nombre otra vez. Después, el programa va numerando los campos automáticamente, hasta un máximo de 32 campos por registro. El máximo de caracteres que pueden sumar todos los campos de un registro es de 1.000, y la longitud máxima de cada campo es de 254 caracteres. Como se puede ver, admite una capacidad más que aceptable. Al pulsar INTRO en un campo vacío, el programa interpreta que se ha acabado de definir la estructura, y pregunta si se quiere empezar a introducir datos en el fichero. Se puede empezar en ese momento, u optar por hacerlo posteriormente.

Si se dispone de varios ficheros distintos, para seleccionar el fichero con el que se va a ope-

rar, se utiliza el comando USE seguido del nombre del fichero. Si se está usando un fichero, y se quiere cambiar a otro, al volver a utilizar este comando, cierra el fichero antiguo antes de poner a disposición del programa el nuevo fichero.

2. Introducción de datos

El mandato APPEND permite introducir datos en una base ya creada. Al seleccionarlo, se coloca en el primer registro libre del fichero, presentando en la pantalla el número de registro, y los nombres de los campos definidos previamente, quedando preparado para aceptar los nuevos datos. El espacio del que se dispone para introducir estos datos está delimitado entre signos (:) y en video inverso, de manera que éste no puede exceder de la capacidad definida previamente en la estructura. Para acabar, basta con pulsar INTRO en el primer campo vacío del registro, y el programa retorna a la línea de comandos. Además, se pueden intercalar registros entre otros a través del comando INSERT. Previamente habrá que «activar» el registro entre el que se quiere intercalar. La manera de «activar» un registro se ve un poco más adelante.

3. Consulta, modificación y borrado de datos

Para facilitar la búsqueda, consulta y modificación de los datos del fichero, DBASE II dispone de dos comandos: EDIT y BROWSE.

EDIT pregunta el número del registro, presentándolo a continuación en el mismo formato que APPEND. Es decir, un campo en cada línea. Una vez en pantalla, el registro se puede modificar o borrar. Borrar el registro consiste en colocar una marca en el mismo. En la parte superior de la pantalla aparece la palabra DELETED (borrado), y en los listados aparece con un asterisco en la posición anterior al primer campo. A partir de ese momento, y a no ser que se borre dicha marca, no podrá ser copiado, añadido o clasificado. No obstante éste no será físicamente eliminado del fichero hasta que se emplee el comando PACK. La modificación se efectúa simplemente situan-



Para... PCW



do el cursor encima del dato a modificar, y escribiendo encima.

BROWSE es similar a EDIT, excepto en la presentación en pantalla. Esta, la realiza una línea por registro, presentando en cada una tantos campos como quepan en la pantalla. Se puede desplazar la línea a izquierda o derecha de manera que queden otros campos a la vista, o llamar al comando incluyendo una lista de campos. En este caso, sólo saldrán por pantalla los campos pedidos. La ventaja de este mandato está en que para consultas o modificaciones, disponemos de varios registros en pantalla simultáneamente (todos los que quepan), aunque a veces no dispongamos de todos los campos de cada registro.

Para manejar los registros, el programa emplea un puntero, que señala al último registro que haya sido utilizado. Este es el registro «activo». Para presentar dicho registro en la pantalla, se puede utilizar el mandato DISPLAY. Este, presentará el registro «activo» en ese momento, y volverá a la línea de comandos. DISPLAY ALL listará todos los registros, y si se especifica un número de registro, listará dicho registro y éste se convertirá en el registro «activo».

Para listar todos los registros de un fichero, se tecldea LIST. Si hay más registros de los que caben en la pantalla, se puede detener el listado con (CTRL S), y éste continuará al pulsar cualquier otra tecla.

Se puede listar parcialmente el fichero ampliando el comando usando códigos u operadores.

Esto es, se puede pedir un listado en el que los teléfonos empiecen por un determinado número, o en los que el distrito postal sea mayor o menor que un número, etc... Esto se hace con LIST FOR (expresión). LIST OFF lista los registros sin su número de registro. Además, LIST STRUCTURE presenta en pantalla los datos introducidos al crear un fichero, y la última fecha en que éste se puso al día.

Otros mandatos para combinar con los anteriores y poder moverse por el fichero, son

GO SKIP y (?). Su funcionamiento es similar a los anteriores, excepto (?) que además se puede utilizar para realizar operaciones matemáticas y presentar el resultado en pantalla. (Similar a PRINT en Basic).

4. Borrado total de datos y de ficheros

Como se ha dicho anteriormente, colocar la marca de borrado en un registro no significa borrarlo físicamente. Aunque el registro «marcado» no se pueda usar, este borrado no es irreversible. Se puede recuperar de dos formas distintas. La primera es volver a editar el registro, y desactivar el código. La segunda es a través del comando RECALL. Este comando, quitó el asterisco del registro o registros que estén marcados.

Para borrar un fichero del disco, se utiliza el comando DELETE FILE. Pero una vez borrado el fichero, éste no se puede volver a recuperar.

El comando PACK borra definitivamente los registros marcados. Además, en caso de que los registros sean indexados, rehace el índice de manera que corresponda con el fichero de datos.

5. Almacenamiento en la memoria virtual y operadores

Otra característica interesante de DBASE II es que podemos almacenar constantes o variables en la memoria. Esto es, sin escribirlas en disco, y que se borran al apagar el ordenador. El programa reserva espacio en la memoria para 64 variables, de una longitud máxima de 254 caracteres cada una, y con una longitud total máxima de 1.536 caracteres. Su utilidad está en almacenar programa. (Por ejemplo, la fecha).

Los comandos para este cometido son STORE, que guarda la expresión y su contenido en memoria, y RELEASE que la elimina de la memoria. Su uso posterior se hace a través del nombre, como si fuera otra variable.

DBASE II dispone de los operadores aritméticos, relacionales y lógicos comunes (+, -, *, /, ...), pero además incluye uno muy útil. Es (\$). Este operador localiza el contenido de una cadena, o de parte de ella, dentro de otra cadena. Esto es especialmente útil para búsquedas alfanuméricas.

6. Otros comandos

COPY copia (curioso ¿no?) el contenido de un fichero en otro.

REPLACE y CHANGE permiten cambiar el contenido de gran número de campos automáticamente.



SORT clasifica el fichero a través de un campo determinado.

INDEX crea un fichero de claves que se actualiza automáticamente, y permite localizar cualquier registro a través del campo clave de manera casi instantánea.

FIND utiliza el fichero de claves para localizar rápidamente el dato que se pide.

LOCATE localiza el dato pedido, en un fichero no indexado.

REPORT permite organizar la salida de datos de una manera determinada. Esto es, para listar informes, etc... con un formato concreto. Permite poner encabezamientos, fechas...

COUNT da el número de registros que cumplen una condición dada.

SUM acumula el valor de campos numéricos de registros que cumplan una determinada condición.

TOTAL acumula todos los campos numéricos de un fichero, con la particularidad de que los resultados los guarda en otro fichero.

HERRAMIENTAS DE PROGRAMACION

Los programas de bases de datos menos evolucionados, acabarían aquí. Habría que comentar su capacidad, potencia y flexibilidad. Pero éste no. En éste, una vez visto esto, resulta que queda mucho camino por explorar, y que las posibilidades restantes permiten muchas más cosas.

Programar con DBASE II no es más (ni menos...) que crear un fichero, que en vez de datos, almacena comandos y órdenes.

Con un vocabulario más que aceptable, permite crear menús, poner mensajes en pantalla, actualizar un fichero en base a datos de otro, y la mayor parte de las necesidades que se le puedan ocurrir a alguien dentro del campo de la gestión de ficheros.

El aprendizaje de esta manera de programar no es fácil, ni rápido. Pero tampoco es demasiado difícil. Lo primero que hay que dominar, es la parte primera.

Es necesario dominar el manejo de los comandos anteriores. Después de eso, habrá que empezar poco a poco, a dominar esta parte. Desde luego, hace falta tomarse su tiempo. Esto puede ser una mala noticia para los ansiosos, pero si se emplea la paciencia y se es metódico, los resultados pueden ser muy profesionales.

Al explorar las distintas instrucciones, y dado que muchas son similares en uso al Basic, profundizaremos más en las que sean exclusivas o diferentes.

Y por supuesto, los comandos que se verán ahora no sólo se puede sino que es necesario combinarlos con los que ahora siguen para que el programa sea una realidad.

1. El principio

Como se ha dicho anteriormente, un programa en DBASE II es un fichero compuesto de comandos u órdenes de DBASE II exclusivamente. Para su creación y modificación se utiliza el comando MODIFY COMMAND más el nombre que queramos dar al programa. Una vez creado, para ejecutarlo habrá que teclear DO nombre del programa, y éste empezará a ejecutarse. Al crear el fichero, entra en funcionamiento un editor de pantalla, tan simple como fácil de utilizar. La manera de moverse por el editor está descrita en el cuadro de comandos.

Una vez creado el fichero-programa, éste se pone en marcha con el mandato DO más el nombre del fichero.

Desde un programa, a modo de menú, se puede llamar a otros programas con sucesivos mandatos DO. Todos estos programas pueden estar «anidados», de manera que determinada acción en uno de ellos, ponga en marcha otro... y así hasta 16 programas. Pero el retorno debe hacerse siguiendo el camino inverso.

Es decir, que del último hay que pasar al penúltimo... etc. (Esto en caso, repito, de que estén anidados). Para finalizar estos programas hay que utilizar la instrucción RETURN o EOF.

2. Bucles

Para la toma de decisiones condicionadas, DBASE II dispone de varias instrucciones.

IF, ELSE permite los mismos bucles que en Basic, incluyendo expresiones AND, OR, < + y =. No hay que poner Then. La mayor particularidad de este bucle es que la rutina a ejecutar si se cumple la condición, incluye todas las líneas que siguen, a IF, hasta la línea que contenga ENDFIF. Esto hace que, en caso de varios bucles, se necesite un ENDFIF por cada uno. Estos bucles deben necesariamente, estar incluidos uno dentro totalmente de otro.

DO WHILE ejecuta una rutina hasta que se cumpla la condición que se le dé. Esta rutina se acaba con ENDDO, y al igual que en la anterior, debe hacer uno por cada bucle.

3. Datos del teclado

Para pedir contestación por el teclado, se pueden utilizar tres instrucciones:

WAIT interrumpe el programa hasta que se

CONTROL DEL CURSOR

X...:	Pulsando CTRL +
F...:	Cursor hacia abajo
	Idem
A...:	Cursor hacia arriba
	Idem
D...:	Mueve el cursor un carácter hacia la derecha
S...:	Mueve el cursor un carácter hacia la izquierda
V...:	Activa y desactiva el modo de inserción
G...:	Borra el carácter de debajo del cursor
Borr...:	Idem
-Borr...:	Borra el carácter a la izquierda del cursor
W...:	Graba los cambios, y retorna a la línea de comando
Q...:	Abandona la tarea en curso y retorna sin grabar los cambios
R...:	En APPEND, graba un registro y pasa al siguiente
*C...:	En EDIT, graba un registro y avanza al siguiente
*U...:	Activa y desactiva la marca de borrado de registro
B...:	En BROWSE, desplaza el registro un campo hacia la derecha
Z...:	En BROWSE, desplaza el registro un campo hacia la izquierda
ESC...:	Retorno a la línea de mandatos

5. Uso de varios ficheros

Para posibilitar el uso simultáneo de dos ficheros, están los comandos SELECT PRIMARY y SELECT SECONDARY. De esta manera se tienen abiertos y conservando el puntero en el registro que interese. Recordemos que USE cierra el fichero que haya abierto antes de abrir otro.

El modo de funcionamiento es el siguiente. Con SELECT PRIMARY o SELECT SECONDARY se selecciona en todo momento el fichero en uso. Si usando el fichero primario hay que referirse a una variable del fichero secundario, hay que antepone (S.) al nombre de la variable, y si es al revés, se antepone (P.).

Además, se pueden combinar varios o todos los campos de los dos ficheros para crear un tercero. Esto se usa con el comando JOIN. Su funcionamiento es el siguiente. Lee el primer registro de la primera base de datos, y busca los registros de la segunda base para comprobar cuáles cumplen la condición impuesta. Estos son copiados en la base final, y entonces pasa al segundo registro de la primera base, repitiendo el ciclo. Este orden, según la longitud de los ficheros, puede tardar un buen rato en completarse.

UPDATE permite sumar o reemplazar datos de dos bases de datos si ambas bases están clasificadas por la misma clave.

6. Otros comandos

CLEAR reinicializa DBASE II, borrando todas las variables y cerrando los ficheros. RESET indica al Sistema Operativo CP/M un cambio de disco.

* Igual al comando REM de Basic. REMARK equivale a la sentencia Print, presentando en pantalla el mensaje.

RENAME cambia de nombre un fichero. Dicho fichero no debe estar abierto.

QUIT TO... sale de DBASE II, ejecuta lo a las órdenes de CP/M que sigan, y vuelve a retornar a DBASE II.

7. Funciones

DBASE II también dispone de algunas funciones similares a las de Basic. Los parámetros que incorporemos a las mismas deben estar entre paréntesis. Estas son las más interesantes.

INT proporciona la parte entera de un número.

VAL convierte una cadena alfanumérica compuesto por números en una variable numérica.

STR convierte una variable numérica en cadena alfanumérica.

LEN da la longitud de una cadena.

FILE retorna el valor verdadero si el fichero que se indique está en el disco. En caso contrario retorna el valor falso.

TRIM elimina los espacios en blanco de una cadena.

RANK da el valor decimal del primer carácter de una cadena.

CHR permite utilizar el código ASCII.

! convierte las letras minúsculas en mayúsculas.

\$ crea una cadena alfanumérica compuesta por parte de otra cadena.

& función de macro sustitución. Utilizada ante el nombre de una variable, reemplaza su nombre por su valor. La variable tiene que ser alfanumérica. Evita trabajo si dicha variable debe ser utilizada con frecuencia.

similar a INSTR en Basic, devuelve la posición donde está contenido el primer carácter de una cadena secundaria dentro de una cadena primaria. En caso de no estar contenida en ella, devuelve el valor 0.

8. Ayuda y programación

Esto es algo que se agradece. Desde la línea de comandos, se puede usar el comando HELP con el nombre de cualquier comando o función, y en la pantalla saldrá una breve explicación en inglés de su cometido y manera de operar. Si hacemos HELP HELP, nos saldrá una lista ordenada alfabéticamente de todos los comandos.

TECLAS DEL CURSOR

Dcha...: Avanza al siguiente campo, o registro si es el último

Izda...: Retrocede al campo o registro anterior

Los controles marcados con * no se deben utilizar en el mandato APPEND

pulse cualquier tecla. La tecla pulsada puede ser utilizada en alguna variable.

INPUT presenta un mensaje en pantalla, y espera respuesta.

Las variables alfanuméricas deben escribirse entre comillas, porque en caso contrario da error. Los números se teclean sin comillas.

ACCEPT admite sólo variables alfanuméricas, por lo que no es necesario escribirlas entre comillas.

READ admite la corrección de variables presentadas en pantalla por el comando GET.

4. Presentación en pantalla

El carácter () se utiliza para situar el cursor en las coordenadas de pantalla deseadas. Es similar al LOCATE del Basic de los CPC. SAY presenta un mensaje en pantalla, a partir de la posición del cursor.

GET sirve para presentar en pantalla el valor de campos de un registro. Cada 64 veces que se use este mandato, hay que hacer CLEAR GETS para liberar la memoria destinada a esto, y poder leer nuevos datos.

ERASE borra la pantalla de texto.

de codificar el manejo de los ficheros, su clasificación o indexación y su puesta al día con una línea de comandos, se pueden hacer cosas que en Basic, en el mejor de los casos, llevarían muchas líneas. No es que se pueda hacer LO MISMO con este programa que con el Basic. Pero lo que este programa hace, lo hace más fácilmente que el Basic. Y si se tarda en aprender a usarlo, el BASIC tampoco se aprende en diez minutos, ¿no?

En la cara 2 del disco del programa, han puesto un gran ejemplo de programa hecho con DBASE II. Un control de stocks. Esto puede ser de gran ayuda a la hora de empezar a programar. Es más que conveniente estudiarlo junto con las instrucciones. Da una idea muy aproximada de las posibilidades que tiene este programa.

CONCLUSIONES FINALES

Antes de nada, decir que no se han descrito todos los comandos y funciones de DBASE II. Esto alargaría el artículo demasiado. Se han descrito las posibilidades más destacables, para que se puedan dar una idea de qué es y cómo funciona este programa.

La primera conclusión es que estamos ante algo más que un programa de base de datos.



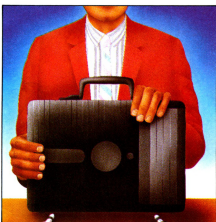
La tarea que se ha hecho es digna de elogio, al menos en cuanto a programación. Sus posibilidades reales hacen que el rendimiento sacado tanto al CPC 6128 como al PCW 8256 sea realmente magnífico.

Puestos a sacar algún defectillo, la implantación realizada en el PCW resulta un tanto peculiar, pero esto es algo a lo que los dueños deben estar acostumbrados. Desde luego, por lo menos el funcionamiento de las teclas de cursor resulta curioso. Y en ambos ordenadores, CPC y PCW, el manejo combinado de CTRL y las teclas no resulta demasiado cómodo al principio, sobre todo teniendo en cuenta que, a veces, la misma combinación de teclas en distintos partes del programa produce resultados diferentes. El cuadro que se acompaña son las funciones habituales, pero no las únicas. El manual se hace totalmente

imprescindible para lograr algo medianamente decente.

Y ya que hablamos del manual, el hecho de que haya sido traducido no tiene precio. Desde luego que se puede mejorar, sobre todo algunas explicaciones que no quedan demasiado claras, pero el resultado conjunto es bueno. Además, el distribuidor ha sacado una publicación bimensual para sus usuarios, dedicada a informar sobre éste y otros productos, lo que puede resultar de una gran ayuda.

En resumen, otro programa que ayuda a Amstrad a mantener esa imagen de ordenador profesional, con la mejor relación calidad-precio del mercado informático español.



DATA BECKER PUESTA FUERTE POR AMSTRAD



Ofrece una colección muy interesante de sugerencias, ideas y soluciones para la programación y utilización de su CPC-664. Desde la estructura del hardware, sistema de funcionamiento Tokens Basic, dibujo que el joystick, aplicaciones de ventanas en pantalla y otros muchos interesantes programas como el procesamiento de datos, editor de sonido, generador de caracteres, monitor de código máquina hasta listados de interesantes juegos.
CPC-664 Consejos y Trucos. 263 págs. P.V.P. 2.200,- ptas.



Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje. Escritos de una forma muy completa y fácil de comprender. Teorema de Pitágoras, progresiones geométricas, escritura circular, crecimiento exponencial, verbos irregulares, guadasdas cuadradas, movimiento pendular, estructura de moléculas, cálculo de interés y muchas cosas más.
CPC-464 El libro del colegio. 300 págs. P.V.P. 2.200,- ptas.



PEKS, POKES Y CALLS se utilizan para introducir al lector de una forma fácilmente accesible al sistema operativo y al lenguaje máquina del CPC. Proporciona además muchas e interesantes posibilidades de aplicación y programación de su CPC.
PEKS Y POKES del CPC 664/6128. 180 págs. P.V.P. 1.800,- ptas.



La técnica y programación del Procesador Z80 son los temas de este libro. Es un libro de estudio y de consulta imprescindible para todos aquellos que poseen un Commodore 128, CPC, MSX u otros ordenadores que trabajan con el Procesador Z80 y desean programar en lenguaje máquina.
El Procesador Z80. 580 págs. P.V.P. 3.800,- ptas.



EL LIBRO DEL FLOPPY del CPC lo explica todo sobre la programación con discos y la gestión de ficheros mediante el floppy DD1-1 y la unidad de discos incorporada del CPC 664/6128. La presente obra, un auténtico estándar, representa una ayuda imprescindible tanto para el que desee incursionar en la programación con discos como para el más curtido programador de ensamblados. Especialmente interesante resulta el listado exhaustivamente comentado del DOS y los muchos ejemplos de ejemplo, entre los que se incluye un completo paquete de gestión de ficheros.
El Libro del Floppy del CPC. 353 págs. P.V.P. 2.800,- ptas.



¡Domina CP/M por fin! Desde explicaciones básicas para almacenar números, la protección contra la escritura, o ASCII, hasta la aplicación de programas auxiliares de CP/M así como CP/M interno para avanzados, cada capítulo del CPC rápidamente encontrará las ayudas e informaciones necesarias, para el trabajo con CP/M. Este libro tiene en cuenta las versiones CPC 2.2 así como CP/M Plus (3.0), para el AMSTRAD CPC 464, CPC 664 y CPC 6128.
CP/M. El libro de ejercicios para CPC. 260 págs. P.V.P. 2.800,- ptas.



TEXTOMAT 8.800 ptas.

¡El procesador de textos más vendido en Alemania, ahora también disponible para

AMSTRAD

BOLETIN DE PEDIDO

FERRE - MORET S.R.L.

Tuete n.º 2, av. 2ª, Tel. 218 02 93
BARCELONA 08006

Deves adonar

Gastos envío

300 ptas.

Adjunto cheque

Reembolso más gastos del mismo

NOMBRE

DIRECCION

ERBE *dades*

SABOTEUR

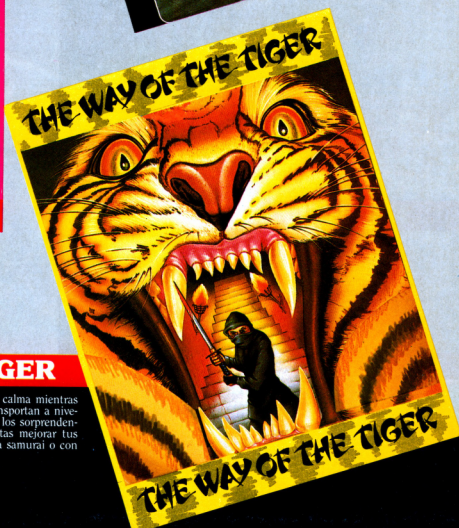
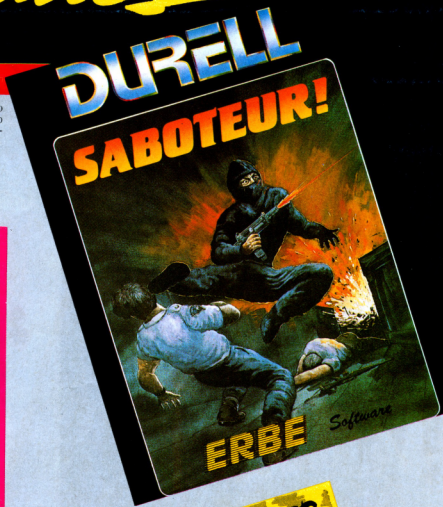
Como experimentado mercenario cuidadosamente entrenado en las artes marciales, debes cumplir la misión que te ha sido encomendada: robar el disco que con la información de los rebeldes tiene el Gran Dictador.



El primer juego de lucha libre hecho para ordenador. Más de 25 movimientos diferentes te permitirán hacer todo tipo de llaves: desde la sujeción de espaldas hasta la voltereta de hombros, pasando por los mismos programadores del legendario "Exploding Fist".

THE WAY OF THE TIGER

Entra en el mudo de los samurais. Mantén la calma mientras el movimiento y las rutinas de combate te transportan a niveles que nunca pensaste posibles. Experimenta los sorprendentes efectos del "Triple Scroll" mientras intentas mejorar tus técnicas de lucha cuerpo a cuerpo, con espada samurai o con mil posibilidades más.



Para que juegues **ERBE**

PING-PONG

La gran sorpresa. Gráficos increíbles, movimiento super-rápido, podrás efectuar las mismas jugadas que si tuvieras la paleta en tu mano.

Botes, rebotes, efectos, dejadas, saques, cortadas, mates... todo es posible con esta maravilla llamada "Ping-Pong"



KNIGHT GAMES

Un desafío medieval en tu ordenador. Transportate a la Edad Media y conviértete en caballero de la Mesa Redonda demostrando tus habilidades en el torneo. Lucha a espada, ballesta, lanza, mazas, arco, hachas y con todas las armas propias de aquella fantástica época.

FRANKIE GOES TO HOLLYWOOD

El juego de los juegos. El que fuera número 1 indiscutible en Spectrum y Commodore, ahora disponible para Amstrad. Mas de 10 juegos diferentes se encierran en "Frankie goes to Hollywood", el programa más original que hayas visto.



DISTRIBUIDOR EXCLUSIVO

ERBE SOFTWARE

C/ STA. ENGRACIADA

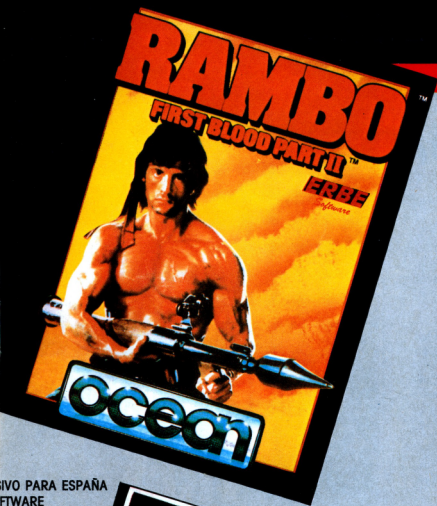
28010 MADRID

TFNO. (91) 400 00 00

DELEGACION DE VENTAS

AVDA. MISTRIAL, 10

TFNO. (93) 400 00 00

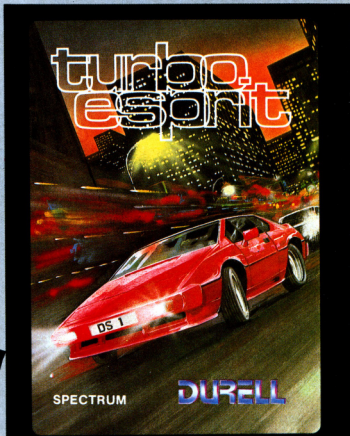
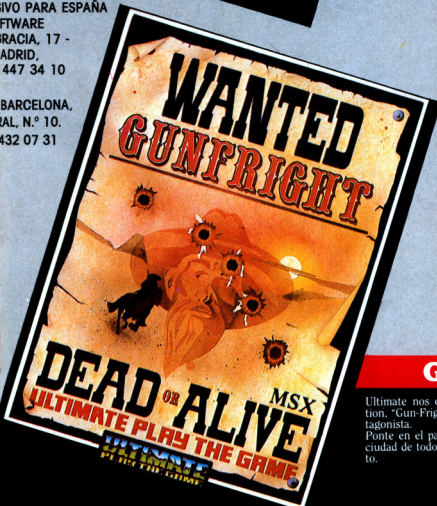


RAMBO

Toda la emoción del film, en tu ordenador. Siéntete como John Rambo en la jungla vietnamita e intenta salvar a tus compañeros prisioneros en el campo de concentración.

DISTRIBUIDOR PARA ESPAÑA
 SOFTWARE
 BARCELONA, N.º 10.
 TEL. 432 07 31

BARCELONA,
 TEL. N.º 10.
 432 07 31



TURBO ESPRIT

Tu misión: vigilar y cuidar el cumplimiento de la ley que se ve amenazada por una terrible banda de delincuentes que han hecho del tráfico de narcóticos su negocio más rentable.

Tus medios: un Lotus Turbo Spirit dotado de uno de los máximos adelantos técnicos y con el que deberás patrullar por calles y avenidas.

GUN-FRIGHT

Ultimate nos ofrece para Amstrad, usando su técnica Filmation, "Gun-Fright", el juego en el que el lejano oeste es el protagonista.

Ponte en el papel de Quikdraw, el sheriff que piensa librar la ciudad de todos los pistoleros a lomos de su buen caballo Panto.

NUMEROS SIN REPETICION

Realizado por el lector
Rafael Sánchez Alcón

El programa, una pequeña subrutina, consiste en seleccionar N números al azar y sin repetición, de un margen de números dados por C.

Las líneas 10-60 identifican el programa.

La línea 80 pide el número mayor de entre los que saldrán los seleccionados. (por ejemplo, en la Lotería Primitiva (49), en el Bingo (90)).

La línea 90 pide cuántos números se quieren seleccionar. En el caso anterior 6 (para la Lotería Primitiva).

La línea 100 dimensiona la Matriz de los números a seleccionar.

Las líneas 120-250 ejecutan la selección y la imprimen.

Hay dos Bucles para generar sonidos. Líneas 170-210 y 220-230.

```

10 *#####
20 *## SELECCION DE NUMEROS ##
30 *## SIN REPETICION ##
40 *## ##
50 *## 1986 POR RAFA S. ALCON ##
60 *#####
70 MODE 1
80 INPUT "introduzca la cantidad de
numeros que va a emplear ",c
90 LOCATE 1,8:INPUT"introduzca la c
antidad de numeros que va a selecci
onar ",n
100 DIM a(n)
110 CLS
120 FOR x=0 TO n-1
130 a(x)=INT(RND*c)+1
140 FOR m=1 TO x
150 IF a(x)<>a(m-1) THEN NEXT m ELS
E 130
170 FOR q=1 TO 50
180 ca=INT(RND*3)+1
190 so=INT(RND*500)+1:du=INT(RND*10
)+1
200 SOUND ca,so,du
210 NEXT q
220 FOR q=60 TO 10 STEP-1
230 SOUND 1,q,1:NEXT q
240 PRINT x+1;"NUMERO GENERADO=>",a
(x)
250 NEXT x

```

LE CONCEDEMOS TRES DESEOS



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!

Vendo programa de Contabilidad General para **Amstrad** con disco. Totalmente original.

Prestaciones profesionales. Totalmente modificable. Adaptable a cualquier impresora paralelo. 450 cuentas auxiliares, 99 cuentas mayores, 6.900 apuntes por disco. Sin límite de apuntes. José Luis Muñoz Traperero. Las Torres. Plaza Cejilla, 2 - 2.º D. Jerez de la Frontera (Cádiz).
Tel (956) 32 68 95 - 31 24 58.

Programador y usuario de un **Amstrad**, desearía contactar con usuarios de cualquier modelo **Amstrad**, para intercambio de cualquier tipo de programas. Realizo programas a medida por encargo (Programas de gestión). Ponerse en contacto con: Miguel Angel Fanega López de la Franca. C/ Reyes Católicos, 1 - 1.º A. Daimiel (C. Real) Tel. (926) 85 09 08.

Se intercambian juegos para el **Amstrad** CPC-464 con usuarios de Barcelona o provincia. Llamar de 15,30 a 17 h o de 21 a 22 h al Tel. 212 42 84. Preguntar por Iván.

Cambio ordenador **Amstrad** CPC-464 monitor color en perfecto estado, más 30 programas comerciales, por **Amstrad** CPC-664 monitor color pagando diferencia justa. Llamar al Tel. (93) 212 42 84 de Barcelona. Preguntar por Iván. (Llamar de 21,30 a 22,00 h).

Desearía contactar con usuarios del **Amstrad** CPC-6128, en Cádiz y su provincia. Interesados escribir a: Silvia Rodríguez Fernández. Avda. Ramón de Carranza, 22 - P. 6.º D. 11006 Cádiz. Tel. 28 72 80.

4 297318

A usted, minorista, y con sólo marcar este teléfono, le concedemos lo que siempre ha esperado de su mayorista informático.

- 1 Todas las marcas**, Amstrad, Spectrum, Commodore... para que con una sola llamada, usted tenga todo lo que necesita.
- 2 Rapidez en el servicio.** Le entregamos su pedido en 24 horas, sin demoras y en cualquier punto de España.
- 3 Trato directo.** Mantenemos un contacto continuo con usted, nos preocupamos por sus problemas y le ayudamos a solucionarlos. Queremos que usted sea algo más que un cliente.

Si es esto lo que pide a su mayorista, LLAMENOS

**DISTRIBUCION
INTROLINE, S.A.**

CUMPLIMOS SUS DESEOS

COPYCAT

Dado que la gran mayoría de usuarios de ordenadores «Amstrad» poseen unidad de disco, hemos creído conveniente realizar un programa coprador que permita traspasar los programas de cinta a disco. Además de ésta, también posee las opciones de copiar de cinta a cinta y de disco a cinta.



El programa está compuesto por un bloque de Basic y otro de código máquina. El primero se encarga de presentar el menú de opciones en pantalla y abrir las salidas para disco o cassette. El bloque en código máquina, es el que realiza la función de traspasar los programas.

Este último está compuesto por las rutinas de **INPUT**, de **LOAD** y **SAVE**. La primera de ellas es la encargada de pedir los nombres de los programas que queremos copiar y el nombre con el que queremos salvar la copia de dicho programa. Las otras dos rutinas realizan la función que su nombre indica.

A continuación, explicaremos detenidamente cómo manejar cada una de las opciones de que consta el programa.

Cuando ejecutemos el programa aparecerá el menú de opciones en pantalla, para elegir cualquiera de ellas deberemos pulsar el número correspondiente a dicha opción.

Si pulsamos la tecla «1», es decir si deseamos pasar un programa de cinta a cinta, en primer lugar, se nos pedirá el nombre del programa que deseamos copiar, en caso de que deseamos pasar el primer programa que se encuentre en la cinta, pulsaremos la tecla «ENTER».

Seguidamente nos pedirá el nombre con el cual deseamos salvar dicho programa, por lo que deberemos indicarle el nombre con el que deseamos almacenarlo en la cinta.

Una vez hecho esto, aparecerá en pantalla el mensaje siguiente:

Press PLAY then any key

por lo que debemos colocar en el cassette la cinta con el programa original, y a continuación pulsar cualquier tecla.

Una vez que se hayan cargado los primeros bloques de programa, aparecerá el mensaje:

Press REC and PLAY then any key

por lo que deberemos sacar la cinta original del cassette y colocar en él la cinta en la que deseamos copiar dicho programa. Esta operación deberemos realizarla tantas veces como nos pida el programa.

Una vez finalizada la copia, se borrará la ventana de información, y aparecerá en pantalla únicamente el menú principal, con lo cual podremos elegir de nuevo cualquiera de las opciones que el programa nos ofrece.

Cargar en cinta y salvar en disco

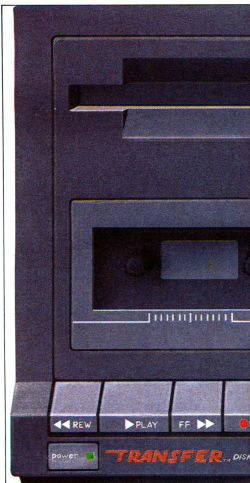
Si deseamos elegir la segunda opción, es decir, cargar de cinta y salvar en disco, se nos pedirá también el nombre del programa que queremos cargar y el nombre con el que lo queremos copiar.

A continuación aparecerá el siguiente mensaje:

Inserta el disco destino y pulsa una tecla.

Con lo que deberemos colocar en la unidad de disco, el diskette en el cual queremos copiar el programa. Cuando hayamos realizado esta operación, deberemos pulsar cualquier tecla.

Seguidamente aparecerá el mensaje de carga del cassette con lo cual



deberemos colocar en el cassette la cinta original y a continuación pulsar una tecla.

A partir de este momento, podremos dejar el disco destino en la unidad, y la cinta en el cassette y pulsar cualquier tecla cada vez que el programa nos lo indique.

La copia finalizará cuando se borre la pantalla inferior de información, y aparezca en pantalla únicamente el menú de opciones.

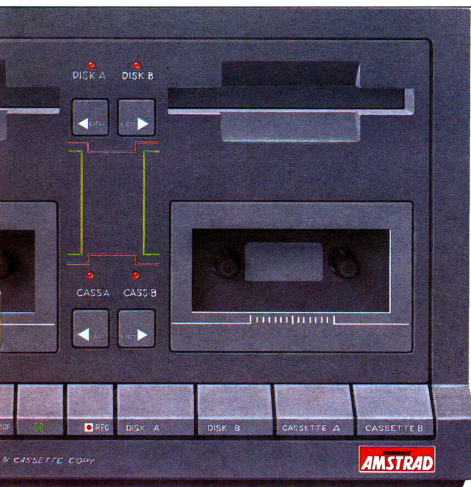
Si elegimos la tercera opción, es decir, si deseamos copiar de disco a cinta, deberemos colocar el disco original en la unidad e insertar la cinta donde deseamos obtener la copia en el cassette. Una vez hecho esto indicaremos el nombre del programa que queremos copiar y el nombre con el que queremos salvar la copia.

Seguidamente aparecerá en pantalla el siguiente mensaje:

INSERTA DISCO FUENTE Y PULSA UNA TECLA

con lo que si hemos realizado las operaciones indicadas anteriormente, deberemos pulsar una tecla cuando el programa nos lo indique.

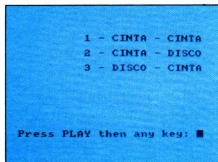
Una vez finalizada la copia, se borrará la ventana de información y aparecerá únicamente el menú prin-



principal con lo que podremos seleccionar cualquier otra operación.

Dicho esto, ya estamos en condición de utilizar el programa y hacer todas las copias de seguridad que deseemos.

Explicaremos a continuación cómo debe procederse para cargar correctamente dicho programa.



En primer lugar teclaremos el listado número 1, que es el programa Basic que controla el funcionamiento de la rutina en código máquina, y que es el encargado de presentar el menú en pantalla.

Una vez hecho esto y comprobado que no existe ningún error, lo grabaremos en cinta o disco de la siguiente forma:

SAVE" COPIADOR

Seguidamente teclaremos el programa cargador en Basic de la rutina de código máquina.

Una vez teclado lo ejecutaremos con la orden «RUN», y una vez finalizado si observamos en pantalla el mensaje siguiente:

ERROR EN DATAS

procederemos a revisar las líneas «DATA» del programa hasta que dicho mensaje no aparezca al ejecutarlo.

Una vez teclado correctamente el programa procederemos a salvarlo en cinta o disco, a continuación del programa Basic salvado anteriormente.

Para salvar correctamente la rutina en código máquina creada por el programa anterior, deberemos hacer lo siguiente:

SAVE" COPY", B,&A000, &25E

con lo cual tendremos cargado en cinta o disco el programa copiador.

Para poder ejecutarlo únicamente tendremos que hacer lo siguiente:

RUN" COPIADOR

y una vez cargado estará listo para efectuar cuantas copias deseemos.

Código MAQUINA

PROGRAMA CARGADOR

```

10 FOR N=0000 TO 00250
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<62371 THEN PRINT "ERROR
EN DATAS"
60 DATA 285,3,187,285,188,187,285
70 DATA 13,168,285,252,168,281,33
80 DATA 239,161,285,184,161,33,1
90 DATA 16,285,74,168,17,223,161
100 DATA 285,217,168,58,235,168,58
110 DATA 222,161,285,196,168,285,18
8
120 DATA 187,33,254,161,285,184,161
130 DATA 33,116,285,74,168,17
140 DATA 286,161,285,217,168,58,235
150 DATA 168,58,285,161,285,196,168
160 DATA 285,188,187,281,175,58,235
170 DATA 168,17,236,168,285,117,187
180 DATA 285,226,168,62,95,285,98
190 DATA 187,285,24,187,71,254,13
200 DATA 288,254,127,48,41,254,31
210 DATA 56,241,254,123,48,237,58
220 DATA 235,168,254,48,40,238,62
230 DATA 8,285,98,187,128,285,98
240 DATA 187,62,95,285,98,187,58
250 DATA 235,168,68,58,235,168,128
260 DATA 18,19,24,284,58,235,168
270 DATA 254,48,187,61,58,235
280 DATA 168,27,62,32,18,62,8
290 DATA 285,98,187,62,32,285,98
300 DATA 187,62,8,285,98,187,62
310 DATA 8,285,98,187,62,32,285
320 DATA 98,187,62,8,285,98,187
330 DATA 62,95,285,98,187,24,152
340 DATA 62,8,285,98,187,62,32
350 DATA 285,98,187,6,16,33,234
360 DATA 168,54,32,35,16,251,281
370 DATA 33,236,168,116,8,237
380 DATA 176,281,11,136,19,11,128
390 DATA 177,32,251,281,8,8,8
400 DATA 8,8,8,8,8,8,8
410 DATA 8,8,8,8,8,8,8
420 DATA 58,92,162,254,1,32,9
430 DATA 33,15,8,285,188,161,285
440 DATA 24,187,33,285,161,78,35
450 DATA 17,159,81,285,148,188,282
460 DATA 166,161,218,171,161,229,58
470 DATA 93,16,251,281,24,285
480 DATA 33,162,285,184,61,285,24
490 DATA 187,33,222,161,78,35,17
500 DATA 159,49,285,119,188,32,3
510 DATA 225,24,184,56,3,225,24
520 DATA 189,229,221,225,253,225,22
1
530 DATA 126,18,253,119,18,221,126
540 DATA 21,253,119,21,221,126,22
550 DATA 253,119,22,221,126,253
560 DATA 119,24,221,126,25,253,119
570 DATA 25,221,126,26,253,119,26
580 DATA 221,126,27,253,119,27,285
590 DATA 137,188,48,67,48,16,285
600 DATA 128,188,48,48,48,9,285
610 DATA 149,188,48,38,48,46,24
620 DATA 233,58,92,162,254,1,32
630 DATA 9,33,159,161,24,33,159
640 DATA 285,24,187,285,143,188,48
650 DATA 8,48,11,285,122,188,48
660 DATA 11,281,33,199,161,24,13
670 DATA 33,199,161,24,33,199
680 DATA 161,24,33,199,161,229
690 DATA 285,188,187,225,78,35,126
700 DATA 35,285,98,187,164,249,21
710 DATA 5,69,82,32,78,82,8
720 DATA 8,8,8,8,8,8,8
730 DATA 8,8,8,8,8,8,8
740 DATA 8,8,8,8,8,8,8
750 DATA 8,8,8,8,8,8,8
760 DATA 8,8,8,8,8,14,78
770 DATA 79,77,66,82,69,32,48
780 DATA 76,79,65,68,41,58,14
790 DATA 78,79,77,66,82,69,32
800 DATA 48,83,65,86,69,41,58
810 DATA 39,73,78,83,69,82,84
820 DATA 65,32,68,73,83,67,79
830 DATA 32,68,69,83,84,78,78
840 DATA 79,32,87,32,88,85,74
850 DATA 83,65,32,85,78,65,32
860 DATA 84,69,67,76,65,38,73
870 DATA 78,83,69,82,84,65,32
880 DATA 68,79,83,67,79,78,78
890 DATA 85,69,78,84,69,32,89
900 DATA 32,88,85,76,83,65,32
910 DATA 85,78,65,32,84,69,67
920 DATA 76,65,8,8,8,8,8

```

PROGRAMA COPION

```

10 MODE 1:INK 0,13:INK 1,0:BORDER 1
31CLS
20 LOAD"COPY",A0000
30 LOCATE 10,7:PRINT "1 - CINTA - C
INTA"
40 LOCATE 10,9:PRINT "2 - CINTA - D
1SDC"
50 LOCATE 10,11:PRINT "3 - DISCO -
CINTA"
60 IF INKEY(64)=0 THEN ITAPE:GOTO 1
00
70 IF INKEY(65)=0 THEN ITAPE,IN:IDI
SC,OUT:IGOSUB 150:GOTO 100
80 IF INKEY(57)=0 THEN IDISC,IN:ITA
PE,OUT:IGOSUB 170:GOTO 100
90 GOTO 60
100 WINDOW #0,1,00,20,25
110 CALL A0000
120 GOSUB 190
130 CLS #0
140 GOTO 60
150 POKE A&25C,1
160 RETURN
170 POKE A&25D,1
180 RETURN
190 POKE A&25C,0:POKE A&25D,0:RETRU
N
    
```

LISTADO DESENSAMBLADO

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

0000 20 MOD0:   EQU  M0000
0C0E 20 HODO:   EQU  M0C0E
8C30 30 BORDER: EQU  M8C30
8C32 40 IN$S:  EQU  M8C32
8B4C 50 CLS:   EQU  M8B4C
8B75 60 CORD:  EQU  M8B75
8B76 70 CLDI:  EQU  M8B76
8B10 80 UNKEY: EQU  M8B10
0000 C00000  90 CALL M8B00
0000 C00000 100 CALL CLS
0000 C00000 110 CALL NAMEX
0000 C0FCA0 120 CALL INIC
0000 C0F 120 RET
0000 21EFA1 140 NAMEX: LD HL,TXTL
0000 C00B01 150 CALL P_ERRO
0000 21B118 160 LD HL,#0001
0000 C00A00 170 CALL INPUT
0000 11DFAA 180 LD DE,NAMEX+1
0000 C00900 190 CALL TRASP
0000 31E0A0 200 LD A,(LONG)
0000 320E01 210 LD (NAMEX),A
0000 C0C000 220 CALL BORRA
0000 C0C000 230 CALL CLS
0000 21FEFA 240 LD HL,TXTS
0000 C00000 250 CALL P_ERRO
0000 21B118 260 LD HL,#0001
0000 C00A00 270 CALL INPUT
0000 31CE01 280 LD A,(NAMEX+1)
0000 C00900 290 CALL TRASP
0000 30E0A0 300 LD A,(LONG)
0000 320001 310 LD (NAMEX),A
0000 C0C040 320 CALL BORRA
0000 C0C000 330 CALL CLS
0000 C00000 340 RET
0000 31E0A0 350 INPUT: XOR A
0000 32E0A0 350 LD (LONG),A
0000 11E0A0 370 LD DE,NAME
0000 C07500 380 CALL COORD
0000 32F0 390 CALL PAUSA
0000 32F0 400 RET
0000 C05000 410 CALL IMPRI
0000 C01000 420 I_BUC: CALL WKY
0000 47 430 LD A,_
0000 FE00 440 CP 13
0000 C0 450 LD A,_
0000 62F7 460 CP 127
0000 2029 470 JR 2,DELET
0000 38F1 480 CALL P_ERRO
0000 38F1 490 JR C,I_BUC
0000 FE70 500 CP 123
0000 38ED 510 JR NC,I_BUC
0000 38E9 520 LD A,(LONG)
0000 FE20 530 CP 40
0000 20E4 540 LD A,_BUC
0000 38E0 550 LD A,_
0000 C05000 560 CALL IMPRI
    
```

Hisoft GENA3.1 Assembler. Page 2.

```

0070 70 570 LD A,_B
007C C05000 580 CALL IMPRI
007F 32F0 590 LD A,_
    
```

```

0001 C05000 600 CALL IMPRI
0004 3AEB00 610 LD A,(LONG)
0008 3C 620 INC
000B 32E0A0 630 LD (LONG),A
000B 70 640 LD A,_B
000C 12 650 LD A,_B
000D 13 660 INC DE
000E 18CC 670 JR I_BUC
000F 30 680 LD A,(LONG)
000F FE00 690 CP 8
000F 20C5 700 JR 2,I_BUC
0010 30 710 DEC
0010 32E0A0 720 LD (LONG),A
0010 10 730 DEC DE
0010 3E20 740 LD A,_32
0010 1E 750 LD (0E),A
0010 3E00 760 LD A,_8
0010 3E00 770 CALL IMPRI
0010 3E20 780 LD A,_32
0010 C05000 790 CALL IMPRI
0010 3E00 800 LD A,_8
0010 C05000 810 CALL IMPRI
0010 3E00 820 LD A,_8
0010 C05000 830 CALL IMPRI
0010 3E20 840 LD A,_32
0010 C05000 850 CALL IMPRI
0010 3E00 860 LD A,_8
0010 C05000 870 CALL IMPRI
0010 3E20 880 LD A,_
0010 3E00 890 LD A,_
0010 C05000 900 CALL IMPRI
0010 3E00 910 BORRA: LD A,_8
0010 3E00 920 CALL IMPRI
0010 3E20 930 LD A,_32
0010 C05000 940 CALL IMPRI
0010 819 950 BORR: LD HL,NAME
0010 21E0A0 960 LD HL,BUC
0010 3E20 970 LD HL,_32
0010 3E 980 INC
0010 18F0 990 D_NZ,BUC
0010 3E 1000 RET
0010 21E0A0 1010 TRASP: LD HL,NAME
0010 810000 1020 LD BC,16
0010 E000 1030 LDIR
0010 C0 1040 OR
0010 810013 1050 PAUSA: LD BC,5000
0010 00 1060 PAUSA: DEC A
0010 810000 1070 LD A,_B
0010 20F0 1070 JR NZ,PAUS
0010 C0 1100 OR C
0010 110 LONG: DEFS 1
0010 3A5C02 1120 NAME: DEFS 16
0010 FE01 1140 CP 1
    
```

Hisoft GENA3.1 Assembler. Page 3.

```

0010 2000 1150 JR NZ,I_PASA
0010 2100A2 1160 LD HL,TXT1
0010 C00001 1170 CALL ERRO
0010 C01000 1180 CALL WKY
0010 21C041 1190 LD HL,NAME
0010 46 1200 LD A,_
0010 21 1210 INC HL
0010 119F51 1220 LD DE,#19F
0010 C00000 1230 CALL M8C0E
0010 C0A0A1 1240 JR 2,ERRO1
0010 D00001 1250 JP NC,ERRO2
0010 1260 PUSH HL
0010 3A00A2 1270 LD A,(FLAG)
0010 1280 CP 128
0010 2120 1290 JR NZ,I_PASG
0010 C00000 1300 LD HL,TXT2
0010 C01000 1310 CALL P_ERRO
0010 C00000 1320 CALL WKY
0010 2100E1 1330 LD HL,NAME
0010 1340 LD (HL),A
0010 1350 INC HL
0010 0E319F 1360 LD DE,#319F
0010 C07700 1370 CALL COT700
0010 2120 1380 JR NZ,BIEN
0010 1E 1390 POP HL
0010 1E00 1400 LD A,_
0010 3003 1410 BIEN: JR C,PASAR
0010 1E 1420 POP HL
0010 1E00 1430 LD A,_
0010 00E1 1440 PASAR: PUSH HL
0010 00E1 1450 POP IX
0010 00E1 1460 POP I
0010 00E12 1470 LD A,(IX+10)
0010 F0712 1480 LD (IX+10),A
0010 00E15 1490 LD (IX+21),A
0010 F0715 1500 LD (IX+21),A
0010 00E16 1510 LD A,(IX+24),A
0010 00E19 1520 LD (IX+24),A
0010 00E18 1530 LD A,(IX+20)
0010 00E1A 1540 LD (IX+20),A
0010 00E1B 1550 LD A,(IX+25),A
0010 00E1C 1560 LD (IX+25),A
0010 00E1D 1570 LD A,(IX+26),A
0010 00E1E 1580 LD (IX+26),A
0010 00E1F 1590 LD A,(IX+27),A
0010 00E20 1600 LD (IX+27),A
0010 00E21 1610 TRABA: CALL ERRO
0010 202F 1620 JR 2,ERRO1
0010 3010 1630 JR NC,ERRO4
0010 C00000 1640 CALL ERRO1
0010 202E 1650 JR 2,ERRO1
0010 3000 1660 JR NC,ERRO0
0010 C00000 1670 CALL M8C0E
    
```

```

0010 2021 1680 JR 2,ERRO1
0010 302E 1690 JR NC,ERRO4
0010 1800 1700 JR TRABA
0010 3A00A2 1710 ERRO0: LD (FLAG),A
0010 FE01 1720 CP 1
    
```

Hisoft GENA3.1 Assembler. Page 4.

```

0010 2000 1730 JR NZ,PASO1
0010 2100A2 1740 LD HL,TXT1
0010 C00001 1750 CALL P_ERRO1
0010 C01000 1760 CALL WKY
0010 21C041 1770 PASE1: CALL M8C0E
0010 3E00 1780 JR 2,ERRO1
0010 3E00 1790 JR NC,ERRO0
0010 C00000 1800 CALL M8C0E
0010 3A00A0 1810 JR NC,ERRO3
0010 C05 1820 RET
0010 1800 1830 LD HL,TXT2
0010 1800 1840 JR P_ERRO
0010 21C7A1 1850 ERRO1: LD HL,TXT2
0010 21C7A1 1860 ERRO2: LD HL,TXT2
0010 21C7A1 1870 ERRO3: LD HL,TXT2
0010 1800 1880 JR P_ERRO
0010 1800 1890 LD HL,TXT2
0010 1800 1900 JR P_ERRO
0010 21C7A1 1890 ERRO4: LD HL,TXT2
0010 1800 1910 LD HL,TXT2
0010 1800 1920 LD HL,TXT2
0010 1800 1930 LD HL,TXT2
0010 1800 1940 LD HL,TXT2
0010 1800 1950 LD HL,TXT2
0010 1800 1960 LD HL,TXT2
0010 1800 1970 LD HL,TXT2
0010 1800 1980 LD HL,TXT2
0010 1800 1990 LD HL,TXT2
0010 1800 2000 LD HL,TXT2
0010 1800 2010 LD HL,TXT2
0010 1800 2020 LD HL,TXT2
0010 1800 2030 LD HL,TXT2
0010 1800 2040 LD HL,TXT2
0010 1800 2050 LD HL,TXT2
0010 1800 2060 LD HL,TXT2
0010 1800 2070 LD HL,TXT2
0010 1800 2080 LD HL,TXT2
0010 1800 2090 LD HL,TXT2
0010 1800 2100 LD HL,TXT2
0010 1800 2110 LD HL,TXT2
0010 1800 2120 LD HL,TXT2
0010 1800 2130 LD HL,TXT2
0010 1800 2140 LD HL,TXT2
0010 1800 2150 LD HL,TXT2
0010 1800 2160 LD HL,TXT2
0010 1800 2170 LD HL,TXT2
0010 1800 2180 LD HL,TXT2
0010 1800 2190 LD HL,TXT2
0010 1800 2200 LD HL,TXT2
    
```

Hisoft GENA3.1 Assembler. Page 5.

```

0010 21B118 2210 LD HL,TXT1
0010 21B118 2220 LD HL,TXT1
0010 21B118 2230 LD HL,TXT1
0010 21B118 2240 LD HL,TXT1
0010 21B118 2250 LD HL,TXT1
0010 21B118 2260 LD HL,TXT1
0010 21B118 2270 LD HL,TXT1
0010 21B118 2280 LD HL,TXT1
0010 21B118 2290 LD HL,TXT1
0010 21B118 2300 LD HL,TXT1
0010 21B118 2310 LD HL,TXT1
0010 21B118 2320 LD HL,TXT1
0010 21B118 2330 LD HL,TXT1
0010 21B118 2340 LD HL,TXT1
0010 21B118 2350 LD HL,TXT1
0010 21B118 2360 LD HL,TXT1
0010 21B118 2370 LD HL,TXT1
0010 21B118 2380 LD HL,TXT1
0010 21B118 2390 LD HL,TXT1
0010 21B118 2400 LD HL,TXT1
0010 21B118 2410 LD HL,TXT1
0010 21B118 2420 LD HL,TXT1
0010 21B118 2430 LD HL,TXT1
0010 21B118 2440 LD HL,TXT1
0010 21B118 2450 LD HL,TXT1
0010 21B118 2460 LD HL,TXT1
0010 21B118 2470 LD HL,TXT1
0010 21B118 2480 LD HL,TXT1
0010 21B118 2490 LD HL,TXT1
0010 21B118 2500 LD HL,TXT1
    
```

Table used: 506 from 1000



p ara que los datos se realicen al trabajo entre M.H. ANS
 TRAD to have a look. Toda las indicaciones de trabajo
 en el lenguaje se encuentran a la disposición en un cas-
 sete mensual. calcitrónico.

SOFTWARE de muchos rombos, para mayores

TOTALMENTE EN ESPAÑOL

C Compilador C

Versión completa del famoso C-Hisoft para CP/M. Capacidades de E/S, ficheros aleatorios y modos de acceso binario y ASCII. Incluye editor ED 80 compatible WORDSTAR.

15.000 ptas.

PASCAL 80 Compilador Pascal

Especial para Z-80. Deja el programa fuente en un programa directamente ejecutable. Incluye ED 80, editor compatible con WORDSTAR.

15.000 ptas.

KNIFE Editor sectores

Permite trabajo directo sobre disco, bien en hexadecimal o ASCII, recuperar ficheros perdidos o borrados, alterar y/o proteger directorios, todo bajo AMSDOS y CP/M.

7.900 ptas.

DEVPAK 80 Ensamblador/des

ED 80: Editor Configurable GEN 80: Macros, inclusión en disco, ensamblador condicional, manipulación bit a bit. MON 80: Monitor y debugger, puntos de ruptura y presentación de memoria.

15.000 ptas.

MODULA-2 Comp. Modula-2

Implementación total del lenguaje MODULA-2 para CP/M. Compilador en un único paso, listo para ser linkado.

19.900 ptas.

TORCH Tutor de CP/M

Diseñado específicamente para AMSTRAD. Incluye THE WAND, creador de menús de programas.

7.900 ptas.

POLYPRINT Multitipos

Transforme su impresora en una imprenta. Permite la impresión en 8 tipos distintos de letras; configurable para cualquier impresora.

*** 11.900 ptas.**

POLY TYPEFACES Multitipos

Añade a la potencia del programa POLYPRINT 8 juegos adicionales de impresión a los ya existentes.

*** 9.900 ptas.**

WRITE HAND MAN Sidekick en CP/M

Residente en memoria, sin interferir en su programa principal le ofrece: Calculadora (Hex-Dec), Block de notas y teléfonos, Calendario, Directorios, etc...

11.900 ptas.

POLYPLOT Impresora/Plotter

Permite realizar gráficos sofisticados en su impresora. Gráficos de pastel, histogramas comparativos, gráficos de líneas, Imágenes de 980 PIXELS de densidad.

*** 11.900 ptas.**

POLYMAIL Mailing

Sencillo sistema de MAIL-MERGE. Idóneo para producir circulares. Incluye editor. Permite la realización de etiquetas autoadhesivas.

*** 10.900 ptas.**

CATALOG Clasificador

Asigna a cada disco un número de serie y además indexa y cataloga los ficheros en ese disco.

8.900 ptas.

MULTI-TEXT Módulo de textos

Módulo de textos, preparado para ser empleado con nuestro lápiz óptico ESP o con las teclas de cursor.

6.900 ptas.

FIRST STEPS Tutor de Newword

Explore las enormes capacidades del procesador de textos NEWWORD; guiado desde los fundamentos del proceso de textos.

7.000 ptas.

MASTER LOCOSCRIPT

Dos cintas audio con instrucciones claras para aprendizaje y apoyo al manual del tratamiento de textos LOSOSCRIPT.

3.000 ptas.

DRAUGHTS- MAN II

Nueva versión mejorada y compatible con nuestra tableta GRAFPAD II: Gran capacidad en gráficos.

6.200 ptas.

*** los 4 juntos 23.800 ptas.**

TYPING CRASH COURSE Inicia a teclear

Curso de iniciación a los teclados, recomendado para personas no acostumbradas a su uso.

9.900 ptas.

TWO FINGERS Curso mecanográfico

Conozca a fondo las posibilidades del teclado, escribiendo con sus diez dedos en lugar de sólo dos.

9.900 ptas.

IVA no incluido



DE VENTA EN LOS MEJORES COMERCIOS DE INFORMATICA
Si Vd. tiene alguna dificultad para obtener los programas, puede dirigirse a:

Ofites
Informática

Avda. Isabel II, 16 - 8º
Tels. 455544 - 455533
Télex 36698
20011 SAN SEBASTIAN

CONDICIONES ESPECIALES PARA DISTRIBUIDORES
EDITOR Y DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA

AMSTRAD CPC - 464

AMSTRAD



ORDENADOR

SERIE CPC

UNIDAD CENTRAL MEMORIAS

- Microprocesador 280A - 64K RAM ampliables - 32K ROM ampliables
- **TECLADO** • Teclado profesional con 74 teclas en 3 bloques - Hasta 32 teclas programables - Teclado redefinible
- **PANTALLA** • Monitor RGB verde (12" o color (14"))

	Normal	Alta Res.	Multicolor
Col x líneas	40 x 25	80 x 25	20 x 25
Colores	4 de 27	2 de 27	16 de 27
Puntos	320 x 200	640 x 200	160 x 2

- Se pueden definir hasta 8 ventanas de texto y 1 de gráficos • **SONIDO**
- 3 canales de 8 octavas moduladas independientemente - Altavoz interno regulable - Salida estéreo • **BASIC**
- Locomotive BASIC ampliado en ROM - Incluye los comandos AFTER y EVERY para control de interrupciones

AMSTRAD CPC 464

- **CASSETTE** • Cassette incorporada con velocidad de grabación (1 ó 2 Kbaudios) controlada desde Basic • **CONECTORES**
- Bus PCB multiuso, Unidad de Disco exterior, paralelo Centronics, salida estéreo, joystick, lápiz óptico, etc.
- **SUMINISTRO** • Ordenador con monitor verde o color - 8 cassettes con programas - Libro "Guía de Referencia BASIC para el programador" - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA.

TODO POR 59.900 Pts. (monitor verde)
90.900 Pts. (monitor color)

AMSTRAD CPC 6128

- **UNIDAD DE DISCO** • Unidad incorporada para disco de 3" con 180K por cara • **SISTEMAS OPERATIVOS**
- AMSDOS, CP M 2.2, CP M Plus (3.0)
- **CONECTORES** • Bus PCB multiuso, paralelo Centronics, cassette exterior, 2ª Unidad de Disco, salida estéreo, joysticks, lápiz óptico, etc.
- **SUMINISTRO** • Ordenador con monitor verde o color - Disco con CP M 2.2 y lenguaje DR LOGO - Disco con CP M Plus y utilidades - Disco con 6 programas de obsequio - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA.

TODO POR 84.900 Pts. (monitor verde)
119.900 Pts. (monitor color)

PCW - 8256

AMSTRAD CPC - 6128



ES AMSTRAD

¡¡Increíble!!

AMSTRAD PCW 8256

UNIDAD CENTRAL. MEMORIAS

- Microprocesador Z80A - 256K RAM de las que 112K se utilizan como disco RAM
- **TECLADO** • Teclado profesional en castellano (ñ, acento...) de 82 teclas

- **PANTALLA** • Monitor verde de alta resolución - 90 columnas x 32 líneas de texto • **UNIDAD DE DISCO** • Disco de 3" y 173K por cara - Opcionalmente, 2ª Unidad de Disco de 1 Mbyte integrable

- **SISTEMA OPERATIVO** • CP M Plus de Digital Research • **IMPRESORA** • Alta calidad (NLQ) a 20 c.p.s. - Calidad estándar a 90 c.p.s. - Papel continuo u hojas sueltas - Alineación automática del papel - Caracteres normales, comprimidos, expandidos, control del paso de letra (normal, cursiva, negrita, subíndices, superíndices, subrayado, etc).

- **OPCIONES** • Kit de Ampliación a 512K RAM y 2ª Unidad de Disco - Interface Serie RS 232C y paralelo

Centronics • **SUMINISTRO** • Ordenador completo con teclado, pantalla, Unidad de Disco e Impresora - Discos con el procesador de Texto LocoScript, CP M Plus, Mallard, BASIC, DR, LOGO y diversas utilidades - Manuales en castellano - Garantía Oficial AMSTRAD ESPAÑA.

TODOS POR 129.900 Pts.



Los más prestigiosos paquetes de Software Profesional, en formato AMSTRAD... a "precios AMSTRAD"

Existe también la versión **PCW 8512** con **512K RAM** y la 2ª Unidad de Disco de 1 Mbyte incorporada **PVP. 169.900 Pts.**
* El **PCW 8256** puede utilizarse como terminal y en comunicaciones.

El I.V.A. no está incluido en los precios.

NOTA: Es muy importante verificar la garantía del aparato ya que sólo **AMSTRAD ESPAÑA** puede garantizarle la ordenada reparación y sobre todo materiales de repuesto oficiales (Monitor, ordenador, cassette o unidades de discos).

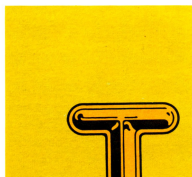
AMSTRAD ESPAÑA

Avda. del Mediterraneo, 9. Tels. 433 45 48 - 433 48 76.
28001 MADRID

Delegación Cataluña: Tarragona, 110 - Tel. 325 10 58.
08015 BARCELONA

SWEEVO'S WORLD

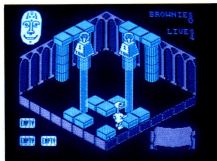
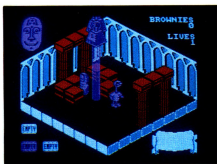
El mundo de Sweevo se encuentra emplazado en las profundidades de un templo sagrado, en el cual los peligros y extraños fenómenos que nos aguardan, sólo pueden ser descubiertos por un intrépido explorador.



Todo comenzó cuando Sweevo, un apuesto joven de alta alcurnia, decidió emprender su última y más ambiciosa expedición al antiguo Egipto.

Dada su gran afición a las expediciones, no pudo resistir la oportunidad de visitar las milenarias construcciones de Abu Simbel.

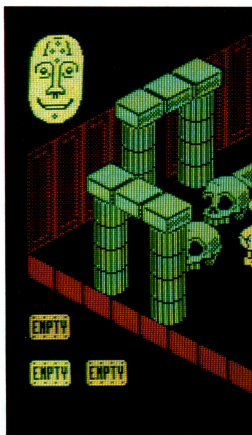
En su empeño por descubrir nuevas vías y cámaras secretas, tiene la fortuna de encontrar una estancia oculta, hasta la cual no había conseguido llegar ningún explorador anterior.



El aspecto de la cámara es de lo más sencillo, no conteniendo tesoros ni piezas de gran valor, la sala está totalmente vacía y aparentemente, aunque inexplorada, no contiene nada interesante.

Tras un reconocimiento más profundo y fijándose especialmente en una de las paredes laterales, descubre unas inscripciones jeroglíficas que desentrañan los secretos de la cámara en la que nos encontramos.

La interpretación de las inscripciones no es fácil, pero nuestro hombre, como aficionado arqueólogo que es, consigue descifrarlas no sin esfuerzo.



El mensaje es el siguiente:

En esta cámara, se encuentra la entrada al mundo extracorpóreo del emperador Ramses III, quien consiga entrar en él podrá encontrar riquezas que colmarían la ambición de cualquier mortal. La entrada de este mundo, se encuentra bajo una losa de esta cámara, situada a veinte pasos al este del lugar señalado en esta inscripción: entrar tiene un precio.

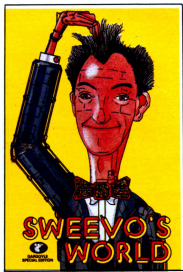
Intrigado por la parte final del mensaje y llevado por su espíritu aventurero, Sweevo no lo duda un instante y se pone a la búsqueda de la losa, la cual no tarda en encontrar, levantándola con gran esfuerzo.

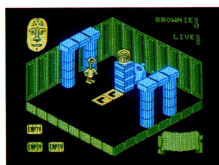
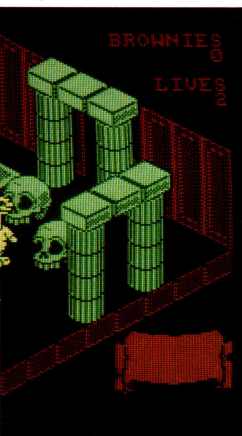
Una luz con vida propia sale de la porción de suelo que cubría la losa, ahora todo está en las manos de Sweevo, la inscripción ha conseguido intrigarle, pero a la vez el miedo a lo desconocido le aconseja prudencia.

Solamente unos minutos de contemplación de la extraña luz que sale del suelo, le hacen dirigirse irremediablemente hacia el hueco y dejar-se caer por él.

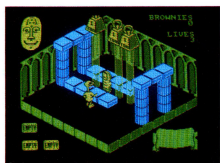
La caída es rápida y Sweevo se ve conducido a las profundidades del templo sagrado, en este mundo de ultratumba, extraños moradores y visiones aterradoras nos aguardan.

Compatible: CPC/464, CPC/664 y CPC/6128





Mister JOYSTICK



Al principio del juego podemos elegir la zona de salida, ofreciéndonos cuatro posibilidades; The first cut, Dead centre, Waiting room y Fruit salad.

En cada una de las pantallas que recorremos nos aguardan trampas y extraños fenómenos; dedos gigantes que surgen del suelo, lasas móviles, cráneos de enorme tamaño, lápidas, seres hostiles y frutas de mostruosas dimensiones.

En determinadas habitaciones podemos recoger objetos que nos serán de utilidad en otras pantallas, elevadores y demás artilugios nos permitirán llegar a los que no podamos alcanzar por nuestros propios medios.

Existen pasos que nos permiten pasar de una pantalla a otra de distinto nivel; ciertas baldosas del suelo al ser pisadas nos catapultan hacia niveles superiores, mientras que en otras zonas hay huecos en el suelo que nos permiten pasar al nivel inferior. Nuestro objetivo en el juego es recorrer el mayor número de cámaras posible, consiguiendo en ellas los objetos que hacen aumentar nuestra puntuación. Al final de cada partida aparecerá una relación del tanto por ciento de la aventura resuelto y de los distintos objetos y elementos hostiles eliminados.

Como parte importante del juego, debemos destacar la realización de los gráficos.

La construcción de los mismos se ha hecho, buscando una perspectiva plana tridimensional (del tipo Knight Lore), en la cual nuestro hombre se mueve en las tres direcciones

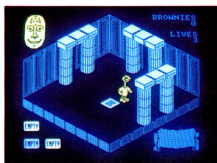
espaciales posibles, permitiendo cualquier movimiento.

El dibujo de los distintos elementos que forman cada pantalla, está realizado con una técnica impecable, consiguiendo unos buenos efectos visuales; cabe destacar los gráficos de las frutas gigantes y las barras de energía que sustentan pesas de acero de 1.000 kg.

El juego está realizado en el modo de pantalla 1, que si bien no permite una gran riqueza de colorido, solamente se pueden tener en pantalla cuatro colores simultáneamente, si permite una definición mayor, al manejar unos puntos más pequeños.

Cosa que ayuda en gran manera a conseguir unos gráficos más definidos y precisos que los que tienen los juegos realizados en el modo 0 de dieciséis colores.

Una aventura de larga duración con sorpresas y peligros en cada pantalla, para la cual los mapas y el afán de entrar en nuevas cámaras son pieza clave del éxito.



Recorrer el número desconocido de cámaras que lo integran, es sin duda la aventura más excitante y peligrosa para cualquier mortal. Sweet's world es un programa de los de «mapa y a buscar».

Cada nueva pantalla tiene su especial significación y mediante ella podemos pasar a otras diferentes en las cuales encontraremos toda clase de cosas.



DESCUBRAMOS ALGUNOS CARACTERES REALES

Hemos visto, por el momento, que hay dos formas de hacer que aparezca en la pantalla cualquier tipo de información. Utilizamos el cursor de textos para escribir caracteres alfanuméricos y el de gráficos para dibujar y trazar líneas.



Utilizar cualquiera de estos métodos para escribir algo en la pantalla obtenemos un resultado final que concierne al **hardware** del ordenador: el cambio de color de un número de puntos o pixels dentro de la pantalla.

Dependiendo de la combinación particular de los puntos seleccionados, esperamos encontrar una forma reconocible, si escribimos un carácter tal como la letra A o dibujamos una figura con el cursor de gráficos, semejante a un triángulo o un círculo.

En este artículo volveremos a hablar sobre el uso del cursor de textos y examinaremos alguno de los muchos caracteres que podemos escribir en la pantalla. Además vamos a intentar ver la manera de construirlos.

Empecemos inicializando **—resetean—** el ordenador. Ahora le tenemos ya dispuesto en modo comando, y si pulsamos una tecla aparecerá en la pantalla un determinado carácter, colocado en el curso de textos.

Pulsamos la letra A y, por supuesto, aparecerá una A. Pero si miramos el carácter con atención veremos claramente que está formado por una combinación de puntos. Podríamos reproducir la letra A utilizando el comando **PLOT**, siempre y cuando sepan dar con la combinación correcta de puntos a dibujar.

El programa I muestra cómo puede trazarse el carácter punto a punto. Observe que primero escribimos la letra A con la línea 40 y luego dibujamos una serie de puntos de, también, dicha letra.

La línea 90 lee las coordenadas x e y de las instrucciones DATA de las líneas 120 a 180. Por medio de la instrucción **PLOT** conseguimos ir haciendo un trazado relativo dependiendo de las coordenadas del punto dibujado anteriormente. Los caracteres producidos son idénticos, pero es mucho más fácil escribirlo que representarlo punto a punto.

¡Imagine lo que puede ocurrir si por alguna circunstancia le toca trazar con puntos cada una de las letras, números o signos de puntuación! El hecho de reproducir un pequeño

mensaje supondría una considerable cantidad de trabajo, y la programación se convertiría en una empresa tediosa y aburrida.

Afortunadamente no tendremos necesidad de llegar a esto ya que todos pueden escribirse con el comando **PRINT**. Escondida en algún profundo lugar de la memoria del **Amstrad** está toda la información necesaria para generar la «trama» de puntos de los caracteres más comúnmente utilizados y además un buen número de otros que son menos cotidianos. Ha hecho el trabajo por nosotros.

Caracteres y bytes

El juego de puntos requerido para cada carácter se conoce como «patrón» o «trama» de bit. El ordenador almacena el patrón de bit de cada carácter y cuando se le pide que escriba uno en particular, busca la trama apropiada y envía el dato a la pantalla.

Todo ello se realiza por medio de rutinas en código máquina que trabajan muy rápidamente y dan como resultado el que aparezca el carácter en la pantalla incluso antes de darnos tiempo a soltar la tecla.

Cada carácter se diseña o dibuja en una red de 8 por 8 puntos. La figura I ilustra cómo se da forma a la letra A. Es la trama de bit particular que hemos utilizado en el programa I para trazar el carácter.

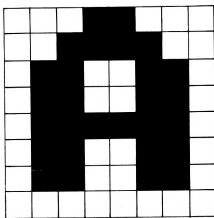
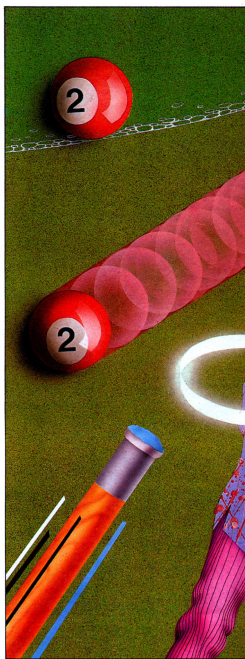


Figura I: Patrón de Bit para «A».



El **Amstrad** tiene una capacidad de almacenar la información relativa a 256 caracteres.

Si desea examinar los patrones de bit de cualquier otro, puede encontrar todos en el manual del usuario.

A cada uno de ellos se le asigna un número comprendido entre 0 y 255 para que el ordenador pueda identificarlo y así escribir en cada momento el carácter correcto.

Por ejemplo, la letra A se corresponde con el número 65 y podemos usar este número de código para escribirla.

En lugar de introducir: **PRINT «A»** podemos utilizar esta otra alternativa:

```
PRINT CHR$(65)
```

Ambas sacarán una letra A en la pantalla. En efecto, la segunda dice: «Escribe el carácter cuyo número de código es 65».



No podremos ver este carácter, pero ¡lo escucharemos! Origina un corto pitido que se genera a través del altavoz interno del ordenador. De aquí el término de caracteres de control, ejerzan control sobre los distintos órganos del micro.

Hay un buen número de ellos. Los números 8, 9, 10 y 11 controlan el movimiento del cursor de textos, de modo que:

PRINT CHR\$(10)

bajará el cursor de textos una línea, mientras que:

PRINT CHR\$(11)

subirá el cursor una línea.

Algunos caracteres de control tienen equivalencia con instrucciones Basic. Por ejemplo, el carácter 29 pondrá el borde de la pantalla del color que queramos, aunque tendrá que ir seguido de otros dos números que determinarán el color elegido. **La instrucción:**

PRINT CHR\$(29);CHR\$(3);CHR\$(3)

produce exactamente el mismo efecto que:

BORDER 3

y coloreará el borde de la pantalla en rojo. Observe que es necesario que introduzcamos dos números de color aunque sean iguales. Con números distintos conseguiremos un borde parpadeante.

Del mismo modo el carácter número 4 es equivalente al comando **MODE**, y la instrucción siguiente seleccionará el Modo 2:

PRINT CHR\$(4);CHR\$(2)

Normalmente no necesitaremos utilizar los caracteres de control en la programación Basic, pero pueden sernos muy útiles cuando forman parte de una larga cadena de caracteres, como veremos la semana que viene.

Aunque hemos dicho que los caracteres 0 a 31 no escriban nada en la pantalla, es posible imprimir el signo representativo del carácter. Para conseguirlo pulsamos la tecla CTRL al mismo tiempo que otra cualquiera. Por ejemplo, pulsando CTRL y G (la séptima letra del alfabeto) al mismo tiempo aparecerá en la pantalla un símbolo que representa una campanita.

Códigos y caracteres

Podemos examinar todos los caracteres manteniendo pulsada CTRL y presionando cada tecla de la A a la Z por turno. Algunos símbolos representan claramente el carácter que les corresponde. Los caracteres 8 a 11 de movimiento del cursor (CTRL más una letra de la H a la K) están representados por flechas que indican la dirección del movimiento del cursor. Sin embargo, algunos símbolos dejan un poco oscuro el significado del carácter del control que representan.

Primeros PASOS

Fijese que si introducimos un comando:

PRINT CHR\$(7)

la función que realiza el carácter de control se ejecuta realmente. En este caso escucharemos un pitido. Pero si pulsamos CTRL y G simultáneamente, aparecerá el símbolo en la pantalla, pero la función no se ejecutará y oiremos «un silencio».

Podemos utilizar la tecla CTRL para introducir caracteres de control en el listado del programa y conseguir que cumpla con su trabajo, siempre que lo pongamos entre comillas dentro de la instrucción apropiada. Por lo tanto, en vez de utilizar una línea parecida a esta:

10 PRINT CHR\$(7)

conseguiremos el mismo resultado utilizando el siguiente comando:

10 PRINT ""

el platillo volante es el símbolo que aparece al pulsar juntos CTRL y G. Ahora cuando ejecutemos el programa la función de control se realizará exactamente cuando se ejecute la línea 10, pero el símbolo no aparecerá en la pantalla.

El segundo grupo de caracteres son los numerados entre 32 y 126. Representan todas las letras, números y signos de puntuación que aparecen en el teclado. El programa 11 nos muestra todos estos caracteres junto con sus respectivos números de código.

Una característica interesante de los caracteres es que forman un juego standard que podemos encontrar en la mayor parte de los ordenadores.

Como ya hemos visto, si ordenamos al **Amstrad** para que escriba el carácter número 65 aparecerá en la pantalla la letra A. Si tenemos acceso a otro ordenador podemos intentar escribir con el mismo número de carácter.

Nos encontraremos con que también aparece la letra A, aunque no esté definida de la misma forma. De cualquier modo la letra A seguirá siendo legible. Podemos deducir de todo esto que hay una lista standard de códigos que se corresponden con cada una de las letras.

Este conjunto de caracteres se conoce como código Ascii. El nombre proviene de la abreviatura de «American Standard Code for Information Interchange». Como el propio título implica, se trata de un código creado en América, pero que ha sido generalmente aceptado como código standard para todos los caracteres alfanuméricos en la mayoría de los ordenadores.

Podríamos intentar utilizar la mencionada instrucción para sacar algún otro sustituyendo el 65 por cualquier otro número.

Por el momento vamos a evitar dar valores que están por debajo de 32 o por encima de 126. Dentro de este rango encontraremos las letras, números y signos de puntuación fácilmente reconocibles —los más usados.

Los 256 caracteres incorporados pueden ser divididos en tres grupos distintos. Los que están comprendidos entre 0 y 31 son un juego especial de caracteres de control.

Nos estamos refiriendo a los caracteres que no son imprimibles y que generalmente no hacen aparecer ningún signo en la pantalla.

Un ejemplo útil de uno de estos caracteres de control es el que tiene el código número 7. Puese introducir en el ordenador:

PRINT CHR\$(7)

De un país a otro encontraremos variaciones sin importancia en algunos símbolos de los menos utilizados y signos de puntuación. Siendo un código americano, no aparecen, por ejemplo, el símbolo de la libra esterlina o el de la peseta en este juego standard.

En Basic hay una palabra clave para representar el código **AscII** de un determinado carácter. Si desea encontrar el número de código que tiene la letra A, por ejemplo, puede utilizar el comando:

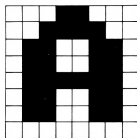
```
PRINT ASC ("A")
```

Nos devolverá el valor 65, que como ya sabemos es el número de código de la letra A. El programa III es una variante del programa II, pero en este caso trabaja de un modo menos directo. Saca los valores AscII de cada una de las letras de la A a la J. Están contenidas en la instrucción DATA de la línea 120.

Si desea que el programa imprima todos los códigos AscII, tendría que ampliar la instrucción DATA para incluir todos los caracteres alfanuméricos.

A la vez podemos ir utilizando el programa IV, que espera a que pulsemos una tecla y entonces saca el número del código AscII correspondiente a esa tecla en particular. Cuando use este programa tenga en cuenta si está puesta o quitada la tecla que fija los mayúsculas (CAPS LOCK). También podemos pulsar la tecla SHIFT (MAYS) para conseguir letras mayúsculas.

128 64 32 16 8 4 2 1



16 + 8 = 24
32 + 16 + 8 + 4 = 60
64 + 32 + 4 + 2 = 102
64 + 32 + 4 + 2 = 102
64 + 32 + 16 + 8 + 4 + 2 = 126
64 + 32 + 4 + 2 = 102
64 + 32 + 4 + 2 = 102

Fuera de los AscII

Finalmente hay un tercer grupo de caracteres, los numerados entre 127 y 255.

Se trata de un grupo de símbolos especiales cuya definición no es standard.

Es muy poco probable que podamos encontrar un juego de caracteres idéntico al del **Amstrad** en algún otro ordenador.

Como ve, los diseñadores han sacado el máximo rendimiento al hecho de que su **Amstrad** pueda almacenar hasta un total de 256 caracteres y han creado un interesante surtido de símbolos poco comunes para que podamos utilizarlos en nuestros programas.

Podemos examinar de nuevo el manual del usuario para ver todos los caracteres que posee, o si no cambie la línea 40 del programa II por la siguiente:

```
40 FOR carácter = 127 TO 255
```

Comprobará que hay toda clase de caracteres desde símbolos matemáticos a musicales, de noipes a hombrecitos. Si observa cuidadosamente quizá hasta podría ver un ocasional «marcianito» (¿Qué micro puede ser más completo?).

Pero no podemos acceder directamente desde el teclado a algunos de estos caracteres, así que, para sacarlos en la pantalla, es necesario normalmente teclear:

```
PRINT CHR$(n)
```

donde «n» es el número de código del carácter. Observe, sin embargo, que al símbolo de la libra si podemos acceder directamente desde el teclado, está a la izquierda de CLR precisamente.

Con todos estos caracteres a nuestra disposición, podríamos pensar que ya tenemos todos los símbolos que necesitamos usar en nuestros programas. Pero no es así. Más tarde o más temprano necesitaremos utilizar uno o varios símbolos que ya no existen en el juego de caracteres incorporados.

Nuestros propios caracteres

Para solucionar este problema, en **Locomotive** nos han dado a los programadores la oportunidad de poder diseñar nuestro propio juego de caracteres e incorporarle después al interno del ordenador. En efecto, cada uno de los caracteres que nos han dado con el **Amstrad** puede ser rediseñado en parte o totalmente y convertirlo en un símbolo diferente.

Antes de empezar a dibujar nuestros propios caracteres necesitaremos saber cuáles son los que ya están incorporados.

¿Recuerda que le comentamos que cada carácter ocupa una red de 8 por 8 puntos? Vamos a examinar atentamente cómo se dibuja la letra A. Conocemos su trazado por la figura I, que se repite en la figura II, pero en esta ocasión hemos añadido unos números al lado de la trama.

Cruzando la parte superior de la figura, cada columna lleva asignado un número. Partiendo de la columna de la derecha, que tiene el número 1, a cada columna de la izquierda le hemos puesto un número que es el doble del de la columna anterior. Estas cantidades representan los ocho bits de un número binario y cada una de sus combinaciones se corresponde con un decimal, comprendido entre 0 y 255.

En la línea superior de la figura II, los cuadros de las columnas 16 y 8 están en negro. Si sumamos los números de estas dos columnas obtendremos 24. Y la representación del número 34 en binario es:

```
00011000
```

Podemos ver que los números 0 y 1 ocupan exactamente el mismo lugar que los cuadros blancos y negros de la fila superior de la figura II.

En otras palabras, los 0 y 1 del número binario reflejan la estructura de la línea. Un 1 representa que la columna está rellena (en negro), y un 0 que permanece vacía (en blanco).

Podemos aplicar la misma norma a cada una de las ocho líneas de la figura II.

Sumamos los valores de cada columna en la que hay un cuadro negro dentro de una fila y el resultado nos da el número que aparece al lado derecho de la figura.

Para comprobar que ha entendido todo lo anterior intente calcular los valores de cualquier otro carácter.

La figura III (la letra I) es fácil y apropiado para ir empezando, pero la figura IV (nuestro amigo el invasor espacial) es un poco más complicado.

Necesitaremos encontrar un sistema para calcular estos valores si queremos diseñar nuestros propios caracteres. Las tablas I y II nos dan la respuesta.

El **Amstrad** ya tiene incorporados los 256 caracteres máximos, así que, ¿cómo podemos sustituir uno de los ya existentes por el nuevo que acabamos de crear? Bueno, en precisamente los números entre 240 y 255. Si rediseñamos el carácter 240, el nuevo símbolo sencillamente se escribe encima del almacenado anteriormente con el número 240.

Pasemos a la práctica. Vamos a intentar crear un carácter nuevo. Antes de nada metamos el comando:

```
PRINT CHR$(240)
```

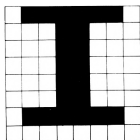
El ordenador nos va a sacar en la pantalla el carácter original con número de código 240, que se trata de la flecha ascendente.

Para definir el nuevo carácter, usaremos la instrucción **SYMBOL** seguida del número de código.

A continuación, y por orden, irán las ocho cantidades que representarán los valores de cada una de las líneas del nuevo símbolo.

Problemos meter en el ordenador:

```
SYMBOL 240,255,255,255,255,  
255,255,255,255
```

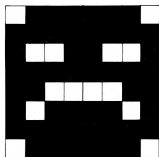


64 + 32 + 16 + 8 + 4 + 2 = 126 01111110
16 + 8 = 24 00011000
16 + 8 = 24 00011000
16 + 8 = 24 00011000
16 + 8 = 24 00011000
16 + 8 = 24 00011000
16 + 8 = 24 00011000
64 + 32 + 16 + 8 + 4 + 2 = 126 01111110
0 + 0 + 0 + 0 + 0 + 0 + 0 = 0 00000000

seguido de: PRINT CHR\$(240)

Ahora en lugar de la flecha ascendente nos ha salido un cuadrado relleno.

Hemos utilizado el valor máximo, 255, en cada una de las líneas del carácter lo que nos da como resultado que los 64 cuadros de la red 8 por 8 estén en negro.



```

0+64+32+16+8+4+2+0=126 01111110
128+64+32+16+8+4+2+0=255 11111111
128+ 0+ 0+16+8+0+0+0+1=153 10011001
128+64+32+16+8+4+2+1=255 11111111
128+64+ 0+ 0+0+0+2+1=195 11000011
128+ 0+32+16+8+4+0+1=189 10111101
128+64+32+16+8+4+2+1=255 11111111
0+64+32+16+8+4+2+0=126 01111110

```

Intente diseñar su propio carácter. Recuerde que puede utilizar cualquiera que tenga número de código comprendido entre 240 y 255.

Pruebe el comando **SYMBOL** seguido de 8 números, entre 0 y 255, elegidos completamente al azar y colocados a continuación del número del carácter.

A la vez intente escoger una secuencia de números que den forma a un símbolo que podamos reconocer.

Si tiene una hoja de papel cuadrículado utilícela como ayuda a la hora de dibujar sus propios caracteres.

El programa V le permite introducir 8 números variables que se utilizan para dibujar un símbolo nuevo.

En él podrá ver en la pantalla cada línea del símbolo en el momento que hemos metido todos los datos en el ordenador.

Diseñar un carácter

El comando **SYMBOL** funciona siempre de la misma manera independientemente del modo de pantalla que deseamos usar.

En Modo 0 aparece el carácter ocupando una superficie mayor que la que ocuparía el Modo 1 y, por lo tanto, utilizaría más píxeles, sin embargo, el símbolo se crea usando el mismo esquema de numeración.

Intente cambiar la línea 20 del programa V al Modo 0 y verá que funciona igual de bien, pero desde luego los caracteres aparecen mucho mayores que antes.

Como dijimos más arriba, el **Amstrad** permite definir 16 nuevos caracteres, pero sólo si no le decimos lo contrario.

Imagine que 16 no son suficientes y, por lo tanto, tenemos que decir al ordenador que necesitamos definir más. Para hacerlo utilice la instrucción:

SYMBOL AFTER n

donde «n» es el número desde donde desea comenzar a redefinir sus propios caracteres. Podemos, en efecto, volver a crear cada uno de los símbolos comprendidos entre 32 y 255.

Es muy probable que queramos redefinirlos todos, pero vamos a imaginar que necesitamos, por ejemplo, 64 caracteres nuevos.

Vamos a dibujar de nuevo los numerados del 192 al 255. Para hacerlo usaremos el comando:

SYMBOL AFTER 192

Ahora ya podemos comenzar a definir cualquier carácter a partir del 192 en la forma que lo hacemos habitualmente y que sería parecido a:

SYMBOL 192, 1, 2, 3, 4, 5, 6, 7, 8

Solamente para comprobar que puede hacerse, rediseñaremos todo el juego de caracteres con el programa VI. Pruebe a ejecutarlo y después intente usar el teclado.

¿Puede adivinar qué ha ocurrido?

Y al final...

Es todo por esta semana. Como ve, redefinir algún carácter puede serle muy útil aunque solamente desee una serie de simpáticos alienígenas mucho más completa que la que tenemos.

Dibujar una colección de nuevos caracteres puede resultar un trabajo bastante laborioso que puede hacerse mucho más fácilmente usando el programa especial de utilidades llamado **«generador de caracteres»**.

Lo usaremos en el próximo artículo, y además también veremos la manera de juntar varios de éstos y formar grandes caracteres gigantes.

PROGRAMAS

```

10 REM PROGRAMA I
20 MODE 1
30 LOCATE 1,5
40 PRINT "Esta letra esta escrita "
50 " :a"
50 LOCATE 1,12
60 PRINT "Esta letra esta hecha con
plots"
70 FOR pixel=1 TO 28
80 MOVE 512,208
90 READ x,y
100 PLOT x,y
110 NEXT
120 DATA 2,2,4,2,10,2,12,2
130 DATA 2,4,4,4,10,4,12,4
140 DATA 2,6,4,6,6,6,8,6,10,6,12,6
150 DATA 2,8,4,8,10,8,12,8
160 DATA 2,10,4,10,10,10,12,10
170 DATA 4,12,6,12,8,12,10,12
180 DATA 6,14,8,14

```

```

10 REM PROGRAMA II
20 MODE 1
30 ==1:y=1
40 FOR character=32 TO 126
50 LOCATE x,y
60 PEN 1
70 PRINT character;
80 PEN 2
90 PRINT CHR$(character)
100 yy=y+1
110 IF y=25 THEN y=1:x=x+6
120 NEXT
130 WHILE INKEY="" :WEND

```

```

10 REM PROGRAMA III
20 MODE 1
30 FOR letra=1 TO 10
40 READ letra$
50 PEN 1
60 PRINT "El codigo Ascii de ";
70 PRINT letra$;
80 PRINT " es ";
90 PEN 2
100 PRINT ASC(letra$)
110 NEXT letra
120 DATA A,B,C,D,E,F,G,H,I,J

```

```

10 REM PROGRAMA IV
20 MODE 1
30 PRINT "Pulse una tecla - seguida
de ENTER"
40 PRINT
50 PEN 1
60 INPUT "Tecla ";tecla$
70 PEN 3
80 PRINT "Su codigo Ascii es ";ASC(
tecla$)
90 GOTO 40

```

```

10 REM PROGRAMA V
20 MODE 1
30 SYMBOL 240,0,0,0,0,0,0,0,0
40 FOR numero=1 TO 8
50 LOCATE 1,numero
60 INPUT "Numero ";n(numero)
70 SYMBOL 240,n(1),n(2),n(3),n(4),n
(5),n(6),n(7),n(8)
80 NEXT numero
90 LOCATE 20,5
100 PRINT CHR$(240)
110 LOCATE 1,20
120 INPUT "Intentamos otro 'si$
130 IF UPPER$(si$)="S" THEN GOTO 20

```

```

10 REM PROGRAMA VI
20 MODE 1
25 PRINT "Espera. Vas a oír un soni
do!"
30 SYMBOL AFTER 32
40 FOR character=32 TO 255
50 SYMBOL character,255,255,255,255,
255,255,255,255
60 NEXT character
70 PRINT CHR$(7)

```

GANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

Porque pretendemos que **AMSTRAD SEMANAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis los encargados de realizar estas páginas, en las que podéis aportar ideas y programas interesantes para otros lectores.

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.
- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:
 - Descripción general del programa.
 - Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.
 - Instrucciones de manejo.
- Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.
- No se admitirán programas que contengan caracteres de control, debido a que no son correctamente interpretados por las impresoras.
- En una sola cinta puede introducirse más de un programa.
- Una vez publicado, **AMSTRAD SEMANAL** abonará al autor del programa de **15.000 a 100.000** pesetas, en concepto de derechos de autor.
- Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.
- **AMSTRAD SEMANAL** se reserva el derecho de publicación o no del programa.
- Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANAL**.
- Los programas sospechosos de plagio serán eliminados inmediatamente.

¡ENVIANOS TU PROGRAMA!

Adjuntando los siguientes datos:
**Nombre y apellidos,
dirección y teléfono.**
Indicando claramente en el sobre:

AMSTRAD SEMANAL
a HOBBY PRESS, S. A. La Granja, 39
Pol. Ind. Alcobendas (Madrid)

MASTER COMPUTER

Centro Comercial Guadalupe
Ctra. Canillas, 136-1.ª planta
Tel. 200 80 85 MADRID

Centro Comercial El Bulevar
La Moraleja
Tel. 654 16 12 MADRID

También abierto domingos de 10 a 2
Centro Comercial Ciudad Sto. Domingo
Ctra. de Burgos, km 28.
Tel.: (91) 622 12 89
Algete (Madrid) (Central)

MICRO DEALER AMSTRAD CENTER

MAYORISTAS DE INFORMATICA
AMSTRAD PCW 8512
AMSTRAD PCW 8256
AMSTRAD CPC 6128 Fósforo verde y color
AMSTRAD CPC 472 Fósforo verde y color
Impresoras. Interfaces. Joysticks
Diskettes de 3 puigadas. cable. etc.
Sinclair. Commodore. New Print.
Spectravideo. Compatibles IBM
C. Comandante Zorita. 13. 28020 Madrid
Telfs. 233 07 81 - 233 07 35

FUNDAS PARA TU «AMSTRAD»

464-472-664 y 6128 2.262
8256 3.250
Joystick Quickshot II 1.975

Pago reembolso, más 250 ptas. de gastos de envío.
Indicar modelo y monitor (verde o color).

Pedidos a: **BAZAR POPULAR**
Apartado 27.040
08080 BARCELONA

MASTER HARD

Servicio Técnico
Para **AMSTRAD**
en Galicia,
León
y Asturias.
C/ Magdalena, 213
El Ferrol
Tel.: (981) 35 84 32

OPERACION CAMBIO

Pásate a monitor color por
25.000 ptas.

Voramos:
Tu Amstrad 464 en 50.000 ptas.
Amstrad 664 en 60.000 ptas.
En la compra de un Amstrad
CPC 6128, PCW8256, PCW8512

Consulte para monitor color
☎ (91) 270 34 97 de 4,30 a 8,30

MECA-SCRIB

El Curso de Mecanografía para el AMSTRAD PCW 8256.

¡¡IMPORTANTE PARA ACADEMIAS!!

- Gestión de alumnos.
- Capacidad para 60 alumnos.
en un solo diskette.

Pedidos a:
EDUCOMP, S.A.
C/ Molina de Aragón, 1.
Tel. (91) 22 32 12
19003 GUADALAJARA

PUBLICIDAD

NL New Line

GABINETE DE INFORMATICA

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

ZURBANO, 4 ☎ 410 47 63
28010 MADRID

RUN "TASWORD"

© 1984 TASMAN SOFTWARE LTD.

TASMAN
SOFTWARE

Tasman

SOFTWARE

por fin en España, software a precios británicos

TASWORD

¿Se imagina su ordenador convertido en una máquina de escribir? TASWORD es la mejor relación calidad-precio en tratamiento de texto profesional.

Totalmente en castellano, permitiendo realizar MAIL MERGE, trabajar en bloques sin ninguna interrupción incrementando su velocidad, etc... (en versión 6128 aprovecha las 128 K creando un disco virtual de 64 K).

- Acentos, Ñ, ü, ¿, etc...
- Compatible Productos TASMÁN.
- Adaptación impresoras.
- Configuración propia por usuario.
- Ensamblaje de textos.



9.900 pts.

AMSTRAD
COMMODORE
EINSTEIN
MSX



7.900 pts.

AMSTRAD
COMMODORE
MSX
SPECTRUM



7.900 pts.
SPECTRUM

TAS-SPELL

Primer auxiliar que corregirá la ortografía de sus escritos y pondrá los acentos olvidados no dando margen a ningún error. Contiene un potente diccionario con más de 20.000 vocablos pudiendo Vd. ampliarlos. Complemento ideal para su TASWORD con disco.



Próximamente
en versión PCW 8256
8512

7.600 pts.

AMSTRAD

TAS-PRINT

Con TAS-PRINT la escritura elevada a arte. Utiliza las grandes posibilidades gráficas de su ordenador. Las posibilidades tipográficas las explota al máximo al dar una doble pasada optimizando la calidad.

Los tipos de escritura son: COMPACTA MEDIAN DATA-AUG
LECTURA LIGHT POLICE SCRIPT



7.600 pts.

AMSTRAD
EINSTEIN



5.900 pts.

AMSTRAD
SPECTRUM



6.900 pts.

QL
SPECTRUM

TASCOPY

Sin necesidad de un PLOTTER podrá obtener sus gráficos de pantalla a través de la impresora. Un increíble ZOOM le permite realizar sus gráficos en 4 hojas formando un póster de gran tamaño.



7.600 pts.

AMSTRAD



5.900 pts.

AMSTRAD
SPECTRUM



6.900 pts.

QL
SPECTRUM

GRAFMAN

Programa de E.G. Computer Graphics especialmente diseñado para trabajar conjuntamente con TASCOPY representando las funciones matemáticas en desarrollo de diagramas por coordenadas, permitiendo su efecto "ZOOM" ampliar sectores de dichos diagramas.



5.600 pts.

SOLO AMSTRAD



6.200 pts.

• IVA NO INCLUIDO

TOTALMENTE
EN
ESPAÑOL



DE VENTA EN LOS MEJORES COMERCIOS DE INFORMÁTICA
Si Vd. tiene alguna dificultad para obtener los programas, puede dirigirse a:

Ofites Informática

CONDICIONES ESPECIALES PARA DISTRIBUIDORES
EDITOR Y DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA Y PORTUGAL

Avda. Isabel II, 16 - 8º
Tels. 455544 - 455533
Télex 36698
20011 SAN SEBASTIAN

CALCOMPLEX

Autores: **Daniel Palamo Ortega** y **Angel Ripoll Rodenas**

Este programa ayudará a trabajar con números complejos. Con él podrás realizar numéricamente operaciones que realizas de manera analítica. Además podrás representar gráficamente los números con lo que podrás hacerte una idea de cómo transforma cada operación un número del plano complejo.



En una representación gráfica de este tipo, empleamos un sistema de ejes cartesianos. El eje horizontal es el de las partes reales y el vertical el de las imaginarias. Cada número complejo viene representado por un vector (en la pantalla una línea) que va desde el original, punto 0 0, hasta el punto de coordenadas (real, imaginario) que lo determinan.

Las operaciones que puedes realizar son:

- Operaciones básicas.
- Logaritmos.
- Funciones trigonométricas.
- Módulo y argumento.
- Raíz y potencia.
- Representación gráfica.

Operaciones básicas

En esta opción puedes sumar, restar, multiplicar y dividir los números complejos. Elegida la operación se pedirá el segundo número que interviene en la misma. Tras realizarla aparecerá el resultado y la opción de almacenarlo para posteriores cálculos.

Logaritmo

Permite calcular el logaritmo del número complejo. Tienes posibilidad de elegir la base del mismo (sólo bases reales positivas). Al extraer el logaritmo de un complejo surge el problema de su indeterminación. La función logaritmo de un complejo está multivaluada, es decir, al logaritmo

de un número complejo le corresponden infinitos valores distanciados $2 \cdot \pi \cdot i$ según la siguiente fórmula:
$$\log Z = \log R + i \cdot \arg + 2 \cdot \pi \cdot i \cdot k$$

Funciones trigonométricas

En esta opción se realizan los cálculos con el número principal.

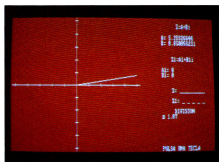
Es posible realizar las siguientes funciones: seno, coseno, tangente, secante, cosecante, cotangente. Y sus inversas, es decir, arcoseno, arccoseno, arcotangente, etc.

No es aconsejable escoger números cuyas partes real o imaginaria sea mayor de 88 radianes ya que el número e elevado a esa potencia está en los límites de rebosamiento de la máquina.

En las funciones inversas se hace uso del logaritmo por lo que estamos sujetos a las mismas restricciones que en su caso.

Potenciación y radicación

Puedes llevar el número principal a cualquier número. En cuanto a raíces puedes calcularlas con índice real o imaginario. Si el índice es entero positivo obtendrás tantas raíces como las que indique el índice, por ejemplo, si es una raíz cúbica aparecerán tres resultados en pantalla, éstos se encuentran almacenados en las matrices RE e IM.



Hay un problema cuando el índice no es entero. Para ser exactos los problemas se presentan cuando el índice es irracional. Hemos considerado que introducirás las aproximaciones racionales al número. La raíz de un número complejo está valuada de forma infinita. Nosotros hemos escogido la determinación principal ($n=0$) esa es la razón de que sólo aparezca una de las posibles raíces.

No se aconseja elevar a un número muy alto por la razones expuestas anteriormente.

Módulo y argumento

Un número complejo puede expresarse en su forma más clásica: parte real e imaginaria o bien dando su módulo (lo que mide el vector que lo representa) y su argumento (el ángulo que forma el vector con el semieje positivo). Puedes calcular el módulo y argumento introduciendo el número como parte real e imaginaria o al revés. Cuando introduces módulo y argumento este último puedes introducirlo en ángulos o radianes. El programa trabaja siempre con radianes.

Cuando un número está muy cerca del número máximo que permite el ordenador volver a repetir la operación fijándose en la ventana central para comprobar que el ordenador no avisa de error (overflow) ya que este error no afecta a la ejecución del programa.

VARIABLES DE CONTROL

FNRD
FNANG
FNREAL
FNIMAG
FNDIVR
FNDIVI
FNLRL
FNLRI
FNCH
FNCH

Calcula el módulo
Calcula el argumento
Calcula parte real
Calcula parte imaginaria
Parte real de división
Parte imaginaria de división
Parte real de potencia
Parte imaginaria de potencia
Calcula coseno hiperbólico
Calcula seno hiperbólico

LISTA DE VARIABLES

RE,IM:	Matrices de almacenamiento de las raíces enteras
Z:	Variable de control de subrutinas
A,B	Parte real e imaginaria del número principal
A1,B1:	Parte real e imaginaria del número secundario
RA,AN:	Módulo y argumento del número principal
RA1,AN1:	Módulo y argumento del número secundario
V1,V2:	Variables transitorias para almacenar número
BA:	Base del logaritmo
MAI,MA,MA1:	Números máximos en gráficos
D1:	Valor de cada división en gráficos
SR,RR,ELR,MR I	
LOR,SER,COR I	
TGR,CSR,SCR I	Variables de salida real de datos
CTR,ARSR,ARCR I	
ARTR,ARCTR I	
ARCCR,ARCSR I	
SI,RI,ELI,MI I	
LOI,SEI,COI I	
TGI,CSI,SCI I	Variables de salida imaginaria de datos
CTI,ARSI,ARCI I	
ARTI,ARCTI I	
ARCCI,ARCSI I	
W?:	Variables de control

TABLA DE SUBROUTINAS

LINEAS	FUNCION
10-24	Cabeceras
25-37	Definición de funciones
38-41	Programa principal
65-79	Raíz entera: realiza raíces con índice entero
80-87	Raíz general: realiza raíz con cualquier índice
88-96	Suma
97-105	Resta
106-114	Exponenciación
115-122	Abandonar programa
123-129	Presenta número en las ventanas inferiores
130-138	División
139-147	Multiplicación
148-158	Introducción número principal
159-169	Introducción número secundario
170-173	Espera pulsación de tecla
174-187	Halla el módulo de un número
188-196	Presentación
197-216	Menú operaciones elementales
217-234	Menú raíz y potencia
235-245	Logaritmo
246-273	Menú trigonometría
274-280	Seno
281-287	Coseno
288-296	Tangente
297-304	Cosecante
305-312	Secante
313-320	Cotangente
321-330	Arcoseno
331-340	Arcocoseno
341-350	Arcotangente
351-359	Arcocotangente
360-368	Arcocosecante
369-377	Arcosecante
378-393	Menú módulo y argumento
394-412	Halla un número a partir del módulo y argumento
413-447	Representación gráfica
459-467	Definición de ventanas
468-482	Almacena número en Z o Z1

Serie
ORO

```

10 REM *****
11 REM *****
12 REM **
13 REM ** COMPLEJOS
14 REM **
15 REM ** por
16 REM **
17 REM ** Angel Ripoll
18 REM **
19 REM ** Y
20 REM **
21 REM ** Daniel Palomo
22 REM **
23 REM *****
24 REM *****
25 GOSUB 194
26 GOSUB 494
27 DEFREAL A-Z
28 CLS:DIM RE(1),IM(1)
29 DEF FNRO(A,B)=MAX(ABS(A),ABS(B))
30 DEF FNRI(A,B)=MIN(ABS(A),ABS(B))/MAX(ABS(A),ABS(B))^2
31 DEF FNRA(A,B)=ATN(B/A)
32 DEF FNRM(A,B)=ARCS(C)
33 DEF FNIM(A,B)=ASIN(B)
34 DEF FNID(A1,A1,B,B1)=(A#A1+B#B1)/(A1#A1+B1#B1)
35 DEF FNLI(RA,AN,A1,B,B1)=(A#A1+B1#LDG(RA)
36 DEF FNLR(RA,AN,A1,B1)=(RA#A1)#E
XP(-B1#AN)
37 DEF FNRA(A)=(EXP(A)-EXP(-A))/2
38 DEF FNRI(A)=(EXP(A)+EXP(-A))/2
39 GOSUB 128
40 GOSUB 46
41 ON Z GOSUB 227,244,156,119,251,2
06,418,452
42 Z=0:GOTO 39
43 :
44 REM ELECCION DE OPCIONES
45 :
46 CLS:CLS #1:LOCATE #1,16,2:PRINT
#1,"OPCIONES
47 LOCATE #1,5,4:PRINT #1,"ABANDONA
R":STRING$(16,"-");"A"
48 LOCATE #1,5,6:PRINT #1,"LOGARITM
O":STRING$(16,"-");"L"
49 LOCATE #1,"-":B:PRINT #1,"TRIGONOH
ETRIA":STRING$(14,"-");"T"
50 LOCATE #1,5,10:PRINT #1,"RAIZ Y
POTENCIA":STRING$(12,"-");"R"
51 LOCATE #1,5,12:PRINT #1,"INTRODU
CIR NUMERO":STRING$(10,"-");"I"
52 LOCATE #1,5,14:PRINT #1,"MÓDULO
Y ARGUMENTO":STRING$(9,"-");"M"
53 LOCATE #1,5,16:PRINT #1,"OPERACI
ONES BASICAS":STRING$(8,"-");"O"
54 LOCATE #1,5,18:PRINT #1,"REFRESE
NTACION GRAFICA":STRING$(5,"-");"G"
55 LOCATE 14,2:PRINT"ELIGE OPCION"
56 Y$=INKEY:IF Y$="" THEN 56
57 IF Y$="R" OR Y$="r" THEN Z=1
58 IF Y$="L" OR Y$="l" THEN Z=2
59 IF Y$="I" OR Y$="i" THEN Z=3
60 IF Y$="A" OR Y$="a" THEN Z=4

```

```

61 IF Y$="T" OR Y$="L" THEN Z=5
62 IF Y$="O" OR Y$="o" THEN Z=6
63 IF Y$="H" OR Y$="h" THEN Z=7
64 IF Y$="M" OR Y$="m" THEN Z=8
65 IF Z=0 THEN G=ELSE WHILE INKEY$
<>:WEND:RETURN
66 :
67 REM RAIZ ENTERA
68 :
69 C=2:CLEAR INPUT:CLS:CLS #1
70 CLEAR INPUT:CLS:CLS #1
71 LOCATE #1,18,2:PRINT #1,"RAIZ"IGO
SUB 165
72 IF RA=0 THEN GOSUB B4:RETURN
73 IF INT(A1)=A1 AND B1=0 THEN 74 E
LSE GOSUB B4:RETURN
74 IF A1>0 THEN ERASE RE,IM:DIM RE(
A1-1),IM(A1-1) ELSE B5
75 FOR N=0 TO A1-1
76 RE(N)=RA*(1/A1):IM(N)=(AN/A1)+C(2
#N1)/A1
77 IF W=1 THEN RETURN
78 IF (N+1) MOD 5=0 THEN GOSUB 179:
CLS #1:CLS:LOCATE #1,18,2:PRINT #1,"
RAIZ"IGO
79 C=C+3:LOCATE #1,3,C:PRINT #1," 1
/2":PRINT #1," Z = ":(C#FNREAL(
RE(N),IM(N))):"+":("FNIAG(RE(
N),IM(N))):")":NEXT
80 W=1:GOSUB 503:GOSUB 179:RETURN
81 :
82 REM RAIZ GENERAL
83 :
84 IF RA=0 THEN IF RA1<>0 THEN RA=
O:RA1=OGOTO B9 ELSE LOCATE 1,1000,1
50,15:LOCATE #1,3,11:PRINT #1,"1/2
":PRINT #1,"Z1 = INDETERMINADO"IGO
SUB 179:RETURN
85 IF RA1<>0 THEN A2=FNDRV(1,A1,O,B
1):B2=FNDRV(1,A1,O,B1) ELSE CLS:C
L#1: SOUND 1,1000,150,1:LOCATE #1,
3,11:PRINT #1,"1/2":PRINT #1,"Z1 =
INFINITO"IGOSUB 179:RETURN
87 A3=FNEL(RA,AN,A2,B2):B3=FNEL(R
A,AN,A2,B2)
88 RA=F#REAL(A3,B3):RA1=FNIMAG(A3,
B3)
89 LOCATE #1,3,11:PRINT #1,"1/2":P
RINT #1," Z = ":(C#RAR):"+":("
FNIAG(A3,B3):")":NEXT
90 W=0:V1=RA:V2=RA1:GOSUB 503:GOS
UB 179:RETURN
91 :
92 REM SUMA
93 :
94 CLS #1:LOCATE #1,18,6:PRINT #1,"
SUMA
95 GOSUB 165
96 SR=A#A1:SI=B#B1
97 LOCATE #1,1,12:PRINT #1," Z+Z1=
"
98 PRINT #1,"(S;R1)": "+":("S;I
1")
99 V1=SR+V2#B1:GOSUB 503:GOSUB 179:
RETURN
100 :
101 REM RESTA
102 :
103 CLS #1:LOCATE #1,18,6:PRINT #1,"
RESTA
104 GOSUB 165
105 RR=A-A1:R1=B-B1
106 LOCATE #1,1,12:PRINT #1," Z-1=
"
107 PRINT #1,"(S;R1)": "+":("S;I
1")
108 V1=RR:V2=R1:GOSUB 503:GOSUB 179
:RETURN
109 :
110 REM ELEVAR
111 :
112 CLS:CLS #1:LOCATE #1,15,6:PRINT
#1,"POTENCIA"IGOSUB 165
113 IF RA=0 THEN IF RA1<>0 THEN ELR
=O:EL=1:GOTO 115 ELSE SOUND 1,1000
,150,15:LOCATE #1,1,12:PRINT #1," Z
-1= INDETERMINADO"IGOSUB 179:RETUR
N
114 ELR=FNEL(RA,AN,A1,B1):ELI=FNEL
I(RA,AN,A1,B1)
115 LOCATE #1,1,12:PRINT #1," Z-1=
"
116 PRINT #1,"(F#REAL(ELR,ELI))"
117 "+":("FNIAG(ELR,ELI))":1"
118 V1=F#REAL(ELR,ELI):V2=FNIMAG(EL

```

```

R,ELI):GOSUB 503:GOSUB 179:RETURN
119 :
119 REM ABANDONAR
120 :
121 WHILE INKEY$<>:WEND
122 CLS #1:SOUND 1,1000,150,15
123 LOCATE #1,10,11:PRINT #1,"*ABAN
DONAR*":
124 LOCATE #1,9,12:PRINT #1,"CONFIRM
A PULSANDO S
125 Y$=INKEY$:IF Y$="" THEN 125
126 IF Y$="S" OR Y$="s" THEN CALL O
127 RETURN
128 :
129 REM NUMERO
130 :
131 CLS #2:CLS#3
132 LOCATE #2,7,1:PRINT #2,"Z=A+B"
:PRINT #2,"A=":PRINT #2,"B=":B
133 LOCATE #3,5,1:PRINT #3,"Z1=A+B
1"
:PRINT #3,"A1=":A1:PRINT #3,"B1=
"
134 RETURN
135 :
136 REM DIVISION
137 :
138 CLS #1:LOCATE #1,16,6:PRINT #1,"
DIVISION
139 GOSUB 165
140 IF A1=0 AND B1=0 THEN SOUND 1,1
140,9000,12:CLS #2:CLS #3:CL
S:LOCATE 9,2:PRINTING SE DIVIDIR PO
R CERD:CALL MRB:GOSUB 129,0:GOSUB
B 131:RETURN
141 LOCATE #1,1,12:PRINT #1," Z/1=
"
142 PRINT #1,"(F#FNDRV(A,A1,B,B1))"
: "+":("F#FNDRV(A,A1,B,B1))":1"
143 V1=FNDRV(A,A1,B,B1):V2=FNDRV(
A,A1,B,B1):GOSUB 503:GOSUB 179:RETU
RN
144 :
145 REM MULTIPLICACION
146 :
147 CLS #1:LOCATE #1,13,6:PRINT #1,"
MULTIPLICACION
148 GOSUB 165
149 MR=A#A1-B#B1:MI=A1#B1+B#B1
150 LOCATE #1,1,12:PRINT #1," Z#Z1=
"
151 PRINT #1,"(S;MR)": "+":("S;MI
1")
152 V1=MR:V2=MI:GOSUB 503:GOSUB 179
:RETURN
153 :
154 REM INTRODUCCION
155 :
156 CLS:LOCATE 15,1:PRINT"INTRODUCE
"
157 LOCATE 17,2:INPUT "A=",A:GOSUB
128
158 CLS:LOCATE 15,1:PRINT"INTRODUCE
"
:LOCATE 17,2:INPUT "B=",B:GOSUB 128
159 IF A=0 AND B=0 THEN RA=O:AN=O:G
OTO 164
160 RA=FNDR(A,B)
161 IF A=0 THEN IF B>0 THEN AN=1/2
:GOTO 164 ELSE AN=3#1/2:GOTO 164
162 AN=FNAG(A,B):IF A<0 THEN AN=AN
+1
163 IF A<0 THEN AN=AN+2#1
164 CLS:RETURN
165 :
166 REM OTRO NUMERO
167 :
168 CLS:LOCATE 15,1:PRINT"INTRODUCE
"
169 LOCATE 17,2:INPUT "A1=",A1:GOSUB
B 128
170 CLS:LOCATE 15,1:PRINT"INTRODUCE
"
:LOCATE 17,2:INPUT "B1=",B1:GOSUB
128
171 IF A1=0 AND B1=0 THEN RA1=O:AN1
=O:GOTO 175
172 RA1=FNDR(A1,B1)
173 IF A1=0 THEN IF B1>0 THEN AN1=1
/2:GOTO 175 ELSE AN1=3#1/2:GOTO 175
174 AN1=FNAG(A1,B1):IF A1<0 THEN A
1=AN1+1
175 CLS:RETURN
176 :
177 REM PAUSA
178 :

```

```

179 CLS:LOCATE 12,2:PRINT"FULSA UNA
TELA":CALL &B0:RETURN
180 :
181 REM MODULO Y ARGUMENTO
182 :
183 CLS:CLS #1:LOCATE #1,14,2:PRINT
#1,"VE INICIAR (243) #MOD/ARG"
184 CLEAR INPUT:CLS:LOCATE 9,2:PRIN
T "GRADOS O RADIANES (G/R)
185 Y$=INKEY$:IF Y$="" THEN 185
186 IF Y$="G" OR Y$="g" THEN WHILE
INKEY$<>:WEND:MI=1:AN=AN+30:PI=LO
CATE #1,17,4:PRINT #1,"GRADOS"IGOTO
189
187 IF Y$="R" OR Y$="r" THEN WHILE
INKEY$<>:WEND:MI=1:LOCATE #1,16,4:PRIN
T #1,"RADIANES"IGOTO 189
188 GOTO 185
189 CLS:LOCATE #1,10,9:PRINT #1,"MOD
ULO":RA
190 IF AN=0 THEN LOCATE #1,10,13:PR
INT #1,"ARGUMENTO":AN
191 IF AN<0 THEN IF W=1 THEN LOCATE
#1,10,13:PRINT #1,"ARGUMENTO":AN+3
:GOTO 190 ELSE LOCATE #1,10,13:PRINT #1,"AN
GUMENTO":AN+2#1
192 IF W=1 THEN AN=AN#PI/180
193 RAD=MI*V1+RA:V2=AN:GOSUB 503:G
OSUB 179:RETURN
194 :
195 REM PRESENTACION
196 :
197 MODE O:TAG: BORDER O:INK 0,3:CLS
:INK 7,0:INK 9,7:GRAPHICS PAPER 7:G
RAPHICS PEN 9
198 FOR S=1 TO 205 STEP 2
199 MOVE 50,S:PRINT"NUMEROS COMPLEJ
OS":I#FRAME
200 NEXT
201 FOR P=1 TO 3000:NEXT
202 RA:RETURN
203 :
204 REM MENU OPERACIONES ELEMENTALES
205 :
206 CLS:CLS #1:LOCATE #1,8,2:PRINT
#1,"OPERACIONES ELEMENTALES"
207 LOCATE #1,16,4:PRINT #1,"OPCIONE
S"
208 LOCATE #1,10,8:PRINT #1,"SUMA"
:STRINGS(16,"")S
209 LOCATE #1,10,10:PRINT #1,"RESTA"
:STRINGS(15,"")R
210 LOCATE #1,10,12:PRINT #1,"DIVISI
ON"
:STRINGS(12,"")D
211 LOCATE #1,10,14:PRINT #1,"MULTI
PLICACION"
:STRINGS(6,"")M
212 LOCATE #1,10,16:PRINT #1,"RETOR
O MENU"
:STRINGS(8,"")X
213 Z=0:LOCATE 14,2:PRINT"ELIGE OPC
ION"
214 Y$=INKEY$:IF Y$="" THEN 214
215 IF Y$="S" OR Y$="s" THEN Z=1
216 IF Y$="R" OR Y$="r" THEN Z=2
217 IF Y$="D" OR Y$="d" THEN Z=3
218 IF Y$="M" OR Y$="m" THEN Z=4
219 IF Y$="X" OR Y$="x" THEN RETURN
220 IF Z=0 THEN 214
221 ON Z GOSUB 92,101,145,156
222 Z=0:GOTO 205
223 :
224 :
225 REM MENU RAIZ Y POTENCIA
226 :
227 CLS:CLS #1:LOCATE #1,12,2:PRINT
#1,"RAIZ Y POTENCIA"
228 LOCATE #1,16,4:PRINT #1,"OPCIONE
S"
229 LOCATE #1,10,9:PRINT #1,"RAIZ"
:STRINGS(13,"")R
230 LOCATE #1,10,13:PRINT #1,"POTENC
IA"
:STRINGS(9,"")P
231 LOCATE #1,10,17:PRINT #1,"RETOR
O MENU"
:STRINGS(5,"")X
232 Z=0:LOCATE 14,2:PRINT"ELIGE OPC
ION"
233 WHILE INKEY$<>:WEND
234 Y$=INKEY$:IF Y$="" THEN 234
235 IF Y$="R" OR Y$="r" THEN Z=1
236 IF Y$="P" OR Y$="p" THEN Z=2
237 IF Y$="X" OR Y$="x" THEN RETURN
238 IF Z=0 THEN 234
239 ON Z GOSUB 69,112
240 GOTO 227
241 :

```

```

242 REM LOGARITMO
243
244 CLS:CLS #1:LOCATE #1,15,2:PRINT
#1,"LOGARITMO"
245 LOCATE 6,2:PRINT "INTRODUCE BAS
E DEL LOGARITMO":LOCATE B,3:INPUT
"PARA EL NÚMERO(A) ",BA:IF BA
=> THEN BA=1 ELSE BA=LOG(BA)
246 IF RA=0 THEN LOCATE #1,2,10:PR
INT #1,"LOG(Z)=-INFINITO":GOSUB 179:
RETURN
247 IF BA<0 THEN LOCATE #1,2,9:PRIN
T #1,"EL NÚMERO(BA) NO ES ADMITIDO
COMO BASE":GOSUB 177:RETURN
248 LOR=LOG(RA)/BA:LI=AN/BA
249 LOCATE #1,2,10:PRINT #1,"LOG Z=
"179:PRINT #1:"179:LI"
250 V1=LOR:V2=L01:GOSUB 503:GOSUB 1
79:RETURN
251
252 REM MENU TRIGONOMETRIA
253
254 CLS:CLS #1:LOCATE #1,14,3:PRINT
#1,"TRIGONOMETRIA"
255 LOCATE #1,5,6:PRINT #1,"SEN":ST
RING(B," "):"":SPACE(5):PRINT #1,
"ARCSEN":STRING$(5," "):"A"
256 LOCATE #1,5,8:PRINT #1,"COS":ST
RING(B," "):"C":SPACE(5):PRINT #1,
"ARCCOS":STRING$(5," "):"R"
257 LOCATE #1,5,10:PRINT #1,"TAN":S
TRING(B," "):"T":SPACE(5):PRINT
#1,"ARCTAN":STRING$(5," "):"N"
258 LOCATE #1,5,12:PRINT #1,"SEC":S
TRING(B," "):"E":SPACE(5):PRINT
#1,"ARCCSEC":STRING$(5," "):"O"
259 LOCATE #1,5,14:PRINT #1,"COSEC
":STRING$(6," "):"D":SPACE(5):PRIN
T #1,"ARCCOSEC":STRING$(3," "):"P"
260 LOCATE #1,5,16:PRINT #1,"COTANG
":STRING$(5," "):"F":SPACE(5):PRIN
T #1,"ARCCOTAN":STRING$(3," "):"M"
261 LOCATE #1,10,18:PRINT #1,"RETOR
N O MENU":STRING$(5," "):"X"
262 LOCATE 14,2:PRINT"ELIGE OPCION"
263 BA=1:Z=0:Y=ENKEY$:IF Y="" THEN
N 263
264 IF Y="S" OR Y="s" THEN Z=1
265 IF Y="A" OR Y="a" THEN Z=2
266 IF Y="C" OR Y="c" THEN Z=3
267 IF Y="R" OR Y="r" THEN Z=4
268 IF Y="T" OR Y="t" THEN Z=5
269 IF Y="N" OR Y="n" THEN Z=6
270 IF Y="E" OR Y="e" THEN Z=7
271 IF Y="D" OR Y="d" THEN Z=8
272 IF Y="O" OR Y="o" THEN Z=9
273 IF Y="V" OR Y="v" THEN Z=10
274 IF Y="X" OR Y="x" THEN RETURN
275 IF Y="F" OR Y="f" THEN Z=11
276 IF Y="M" OR Y="m" THEN Z=12
277 ON Z GOSUB 281,326,288,346,296,
363,313,403,305,385,321,374
278 Z=0:GOTO 251
279
280 REM SEND
281
282 CLS #1:LOCATE #1,10,6:PRINT #1,
"SEND"
283 IF BA=0 GOSUB 531:SER=SEN(A) FNCH(B)
:BA=1:SER=COS(A) FNCH(B):A=P1
284 IF W1<>1 THEN LOCATE #1,12:PR
INT #1,"SEN Z="(SER):"":SEI:"1"
V1=SER:V2=SEI:GOSUB 503:GOSUB 179:
MI=0
285 RETURN
286
287 REM COSENO
288
289 CLS #1:LOCATE #1,17,6:PRINT #1,
"COSENO"
290 P1=A:GOSUB 531:COR=COS(A) FNCH(B)
:CI=SEN(A) FNCH(B):A=P1
291 IF W1<>1 THEN LOCATE #1,12:PR
INT #1,"COS Z="(COR):"":CI:"1"
V1=COR:V2=CI:GOSUB 503:GOSUB 179:
MI=0
292 RETURN
293
294 REM TANGENTE
295
296 CLS #1:LOCATE #1,16,6:PRINT #1,
"TANGENTE"
297 IF BA=0 GOSUB 283:MI=1:GOSUB 290
298 IF COR=0 AND CI=0 THEN LOCATE

```

```

#1,12:PRINT#1,"TG=INFINITO":GOSUB
179:RETURN
299 TGR=FNDIVR(SEN,COR,SEI,CI):I=
FNDIVI(SEN,COR,SEI,CI)
300 LOCATE #1,12:PRINT #1,"TG Z=(
"179:)"+"(:"179:)"
301 V1=TGR:V2=TBI:GOSUB 503:GOSUB 1
79:MI=0:RETURN
302
303 REM COSECANTE
304
305 CLS #1:LOCATE #1,16,6:PRINT #1,
"COSECANTE"
306 M1=1:GOSUB 283:IF SER=0 AND SEI
=> THEN LOCATE #1,12:PRINT#1,"COS
E=INFINITO":GOSUB 179:RETURN
307 CSR=FNDIVR(1,SEN,0,SEI):CSI=FND
I(1,SEI,0,SEI)
308 LOCATE #1,12:PRINT #1,"COSEC
Z="(CSR):"":CSI:"1"
309 V1=CSR:V2=CSI:GOSUB 503:GOSUB 1
79:MI=0:RETURN
310
311 REM SECANTE
312
313 CLS #1:LOCATE #1,17,6:PRINT #1,
"SECANTE"
314 M1=1:GOSUB 290:IF COR=0 AND CI
=> THEN LOCATE #1,2,12:PRINT#1,"SEC=
INFINITO":GOSUB 179:RETURN
315 SCR=FNDIVR(1,COR,0,CI):SCI=FND
I(1,COR,0,CI)
316 LOCATE #1,12:PRINT #1,"SEC Z="(
SCR):"":SCI:"1"
317 V1=SCR:V2=SCI:GOSUB 503:GOSUB 1
79:MI=0:RETURN
318
319 REM COTANGENTE
320
321 CLS #1:LOCATE #1,15,6:PRINT #1,
"COTANGENTE"
322 M1=1:GOSUB 283:MI=1:GOSUB 290:IF
SER=0 AND SEI=0 THEN LOCATE #1,1,
12:PRINT#1,"COTG=INFINITO":GOSUB 17
9:RETURN
323 CTG=FNDIVR(COR,SEI,CI,SEI):CTI
=FNDIVI(COR,SEI,CI,SEI)
324 LOCATE #1,12:PRINT #1,"COTG Z
="(CTI):"":CTI:"1"
325 V1=CTI:V2=CTI:GOSUB 503:GOSUB 1
79:MI=0:RETURN
326
327 REM ARCOSENO
328
329 CLS #1:LOCATE #1,16,6:PRINT #1,
"ARCOSENO"
330 IF RA=0 THEN ARSR=0:ARSI=0:GOTO
341
331 W1=P1:A1=P2:B1=P3:R4=P4:AN=P5:A1
=P6:B1=A1:P1=P1:P2=P2:P3=P3:P4=P4:
1=P2:P1=0:GOSUB 159
332 N=0:GOSUB 761:W=0:A1=P5:B1=P6:A=
P1:B=P2:P1=R4:P2=AN:R4=P3:AN=P4
333 REA=FNREAL(RE(O),IM(O))-B,IMA=FN
FNREAL(RE(O),IM(O))+B:IMA1=FNIMAG
(RE(O),IM(O))-B:IMA1=FNIMAG(RE(O),
IM(O))+B
334 K=FNDR(REA,IMA):K1=FNDR(REA,IM
A1):IF REA=0 THEN IF IMA=0 THEN L=P
1/2 ELSE L=3*PI/2
335 IF REA<0 THEN L=L+FNANG(REA,IMA)
:IF REA<0 THEN L=L+PI
336 IF L<0 THEN L=L+2*PI
337 IF REA=0 THEN IF IMA1=0 THEN L
=L+PI/2 ELSE L=L+3*PI/2
338 IF REA<0 THEN L1=FNANG(REA1,
IMA1):IF REA<0 THEN L1=L1+PI
339 IF L1<0 THEN L1=L1+2*PI
340 ARSR1=L1:ARSI1=-1*LOG(K1):ARSR=
L:ARSI=1*LOG(K)
341 LOCATE #1,1,8:PRINT#1,"SEN"-1 Z
="(ARSR):"":ARSI:"1"
LOCATE #1,1,12:PRINT #1,"SEN"-1 Z="(ARSR):"
+(ARSI):"1"
342 M3=1:V1=ARSR:V2=ARSI:V3=ARSR1:
V4=ARSI1:GOSUB 503:GOSUB 179:RETURN
343
344 REM ARCCOSENO
345
346 CLS #1:LOCATE #1,15,6:PRINT #1,
"ARCCOSENO"
347 IF RA=0 THEN ARCR=1:ARCI=0:GOTO
358
348 W1=P1:A1=P2:B1=P3:R4=P4:AN=P5:A1
=P6:B1=A1:P1=P1:P2=P2:P3=P3:P4=P4:
1=P6:B1=0:GOSUB 159

```

```

349
350 REA=FNREAL(RE(O),IM(O))+B:IMA=FN
FNREAL(RE(O),IM(O))-B:IMA=FNREAL
(RE(O),IM(O))+B:IMA1=FNIMAG(RE(O),
IM(O))+B
351 K=FNDR(REA,IMA):K1=FNDR(REA,IM
A1):IF REA=0 THEN IF IMA=0 THEN L=P
1/2 ELSE L=3*PI/2
352 IF REA=0 THEN IF IMA1=0 THEN L
=L+PI/2 ELSE L=L+3*PI/2
353 IF REA<0 THEN L1=FNANG(REA,IMA)
:IMA1):IF REA<0 THEN L1=L1+PI
354 IF L1<0 THEN L1=L1+2*PI
355 IF REA<0 THEN L1=FNANG(REA,IMA)
:IF REA<0 THEN L1=L1+PI
356 IF L1<0 THEN L1=L1+2*PI
357 ARCR=L1:ARCI=-1*LOG(K):ARCR1=L1:
ARSI1=-1*LOG(K)
358 LOCATE #1,1,8:PRINT#1,"COS"-1
Z="(ARCR):"":ARCI:"1"
LOCATE #1,1,12:PRINT #1,"COS"-1 Z="(ARCR):"
+(ARCI):"1"
359 M3=1:V1=ARCR:V2=ARCI:V3=ARCR1:
V4=ARCI1:GOSUB 503:GOSUB 179:RETURN
360
361 REM ARCCOTANGENTE
362
363 CLS #1:LOCATE #1,14,6:PRINT #1,
"ARCCOTANGENTE"
364 IF RA=0 THEN ARTR=0:ARTI=0:GOTO
369
365 REA=FNDR(1-B,1+B,A,-A):IMA=FN
DIVI(1-B,1+B,A,-A):K=FNDR(REA,IMA):
IF REA=0 THEN IF IMA=0 THEN L=P1/2
ELSE L=3*PI/2
366 IF REA<0 THEN L=FNANG(REA,IMA)
:IF REA<0 THEN L=L+PI
367 IF L<0 THEN L=L+2*PI
368 ARTR=L/2:ARTI=-1*LOG(K)
369 LOCATE #1,8:PRINT#1,"TG"-1 Z=
("ARTR):"":ARTI:"1"
LOCATE #1,1,12:PRINT #1,"TG"-1 Z=
("ARTR+PI):"
("ARTI):"1"
370 M3=1:V1=ARTR:V2=ARTI:V3=ARTR+PI
:V4=ARTI:GOSUB 503:GOSUB 179:RETURN
371
372 REM ARCCOTANGENTE
373
374 CLS #1:LOCATE #1,13,6:PRINT #1,
"ARCCOTANGENTE"
375 IF RA=0 THEN ARCTR=P1/2:ARCTI=0
:GOTO 380
376 REA=FNDR(A,B,1-B,1-B):IMA=FN
DIV(A,B,1-B,1-B):K=FNDR(REA,IMA):IF
REA=0 THEN IF IMA=0 THEN L=P1/2 EL
SE L=3*PI/2
377 IF REA<0 THEN L=FNANG(REA,IMA)
:IF REA<0 THEN L=L+PI
378 IF L<0 THEN L=L+2*PI
379 ARCTR=L/2:ARCTI=-1*LOG(K)
380 ARCTOS="CTG"-1 Z="(STR(ARCTR)
+)"+"(STR(ARCTI)+)"
ARCTI=ARCTR+CT
B+1 Z="(STR(ARCTR+PI)+)"+"(STR
ARCTR):"1"
LOCATE #1,1,8:PRINT#1,
ARCTR:LOCATE #1,1,12:PRINT#1,ARCTR
19
381 M3=1:V1=ARCTR:V2=ARCTI:V3=ARCTR+
PI:V4=ARCTI:GOSUB 503:GOSUB 179:RE
TURN
382
383 REM ARCCOSECANTE
384
385 CLS #1:LOCATE #1,14,6:PRINT #1,
"ARCCOSECANTE"
386 IF RA=0 THEN LOCATE #1,4,12:PRIN
T#1,"NO EXISTE EL ARCCOSECANTE DE
CERO":GOSUB 179:RETURN
387 W1=P1:A1=P2:B1=P3:R4=P4:AN=P5:A1
=P6:B1=A1:P1=P1:P2=P2:P3=P3:P4=P4:
1=P6:B1=0:GOSUB 159

```



LOS MEJORES PROGRAMAS PROFESIONALES DEL MUNDO

¡a precios "AMSTRAD"!

PARA AMSTRAD PCW 8256 Y AMSTRAD CPC 6128

MICROSOFT

MULTIPLAN

Uno de los más prestigiosos y completos "hojas de cálculo" del mundo. Rápido y versátil, ofrece prestaciones, como la de relacionar varias hojas entre sí, que no son frecuentes. La capacidad de ejecutar ordenaciones alfabéticas o numéricas, sus posibilidades en cuanto a formato en pantalla y en impresora, los menús en pantalla y la potencia de cálculo, son características distintivas y destacables de MULTIPLAN.

PVP: 15.100.- Ptas. (+ IVA)

MBASIC INTERPRETER

Reconocido como el estándar mundial de los lenguajes intérpretes para microordenadores. Fácil de aprender y utilizar.

PVP: 15.100.- Ptas. (+ IVA)

MBASIC COMPILER

Totalmente compatible con el MBASIC Interpreter para con una velocidad de ejecución de 3 a 10 veces más rápida. Traduce el código fuente a código objeto y permite una utilización más eficaz del espacio.

PVP: 15.100.- Ptas. (+ IVA)

MS COBOL COMPILER

Lenguaje COBOL según el estándar ANSI, especialmente útil para manejar grandes volúmenes de datos.

PVP: 48.500.- Ptas. (+ IVA)

MS SORT

Flexible programa de ordenación según la técnica de la inserción binaria, utilizable independientemente o incluíble en programas escritos en MS COBOL.

PVP: 15.100.- Ptas. (+ IVA)

MS-FORTRAN COMPILER

El lenguaje más utilizado en aplicaciones científicas y de ingeniería, es una potente implementación del ANSI-FORTRAN X3.9.

PVP: 24.900.- Ptas. (+ IVA)

MS MACRO

Un completo paquete de desarrollo que incluye: MS-MACRO ASSEMBLER, MS-LINK, MS-LIB, MS-CREF y DEBUG.

PVP: 12.000.- Ptas. (+ IVA)



dBASE II

El Generador de Programas por excelencia. Permite crear bases de datos relacionados a partir de comandos sencillos y sin requerir conocimientos de programación. Las aplicaciones de dBASE II son instantáneas y cada usuario puede desarrollar las que mejor se adaptan a sus necesidades: ficheros y molinos, contabilidad, nóminas, control de costos, control de almacén, facturación, etc. Ampliamente acreditado como uno de los programas más útiles y recomendables de cuantos existen para microordenadores. Manual en castellano.

PVP: 17.800.- Ptas. (+ IVA)

DR. DRAW

Programa interactivo para la creación y edición de gráficos y diagramas. Tres elementos básicos — líneas, texto y símbolos — son utilizados para producir gráficos de alta calidad: logos, diagramas de bloques, diagramas de flujo, etc. Los símbolos, tipos de letra y estilos de líneas, pueden alterarse y modificarse a voluntad del usuario.

PVP: 15.100.- Ptas. (+ IVA)

DR. GRAPH

Generador de gráficos — de líneas, barras, columnas y de pastel — de muy sencillo manejo. Permite incluir textos y leyendas con gran flexibilidad de creación y edición.

PVP: 15.100.- Ptas. (+ IVA)

PASCAL-MT+

El más rápido PASCAL existente con implementación completa del estándar ISO. Un compilador de código nativo que genera en formato reutilizable para usar con su monitor de enlaces (linker).

PVP: 15.100.- Ptas. (+ IVA)

CBASIC COMPILER

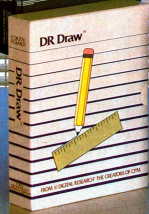
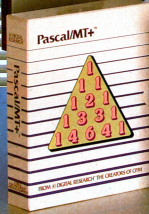
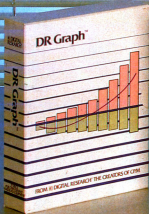
Versión mejorada del clásico lenguaje CBASIC, con mayor velocidad de ejecución y elemento flexible diseñado especialmente para el desarrollo de programas de gestión. Incluye el linker LC80, que cambia la salida del compilador con las rutinas de biblioteca y permite el accionamiento de módulos.

PVP: 15.100.- Ptas. (+ IVA)



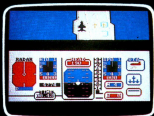
micro **ASHTON-TATE**

P.º CASTELLANA, 179-1.º - 28046 MADRID
Telf. 442 34 33/44



¡DESPEGA CON MICROBYTE!

JUMP JET



JUMP JET

© ANIROG

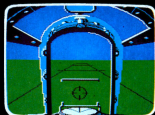
Vive la aventura de pilotar un Harrier. Te encuentras en la cubierta de un portaviones en medio del océano. Tienes que despegar verticalmente, alejarte de la nave y, de repente, te encontrarás solo en el aire, sin ver otra cosa que agua hasta que aparezcan los aviones enemigos dispuestos a destruirte. Defiéndete, lucha por la supervivencia... ¡¡¡ELLOS O TÚ!!!

P.V.P. Cassette, 2.200 Pts. Disco, 2.900 Pts.

SPITFIRE 40

MIRRORSOFT

Trasládate a los años cuarenta, segunda guerra mundial. Ponte a los mandos de un caza Spitfire de la época y siente tú mismo la sensación de volar y combatir en aquellos aparatos donde la pericia del piloto era el novena y cinco por ciento del éxito. Sólo tu sangre fría y habilidad te salvarán del desastre.



Spitfire

40



P.V.P. Cassette, 2.200 Pts. Disco, 2.900 Pts.

para **AMSTRAD**

PRODUCE Y
DISTRIBUYE

microbyte

TU PROGRAMA DE RADIO

claro!



Arribas2

- Entrevistas a fondo
- Exitos en Soft
- Noticias en Hard
- Concursos

LA COPE A TOPE.

— RADIO POPULAR 54 EMISORAS O.M. —

En Barcelona Radio Miramar



SQ LE AYUDARA A CREAR SUS CANTATAS

Parece que ha pasado mucho tiempo desde que comenzamos nuestra exploración sobre los comandos de sonido del Basic del Amstrad. En anteriores artículos hemos descubierto cómo funcionan los parámetros de canal, tono, duración y volumen y también hemos tocado las envolventes del volumen y tono.



continúa de todo esto hemos revisado otra vez el parámetro de canal y estudiamos lo que era un bit significativo, comprendiendo cómo se podrían retener, liberar o sincronizar notas en los diferentes canales.

Y ahora, en este último artículo, echaremos un vistazo a la función SQ.

A primera vista el comando SQ podría parecerse a la función matemática raíz cuadrada de un número. Sin embargo, su finalidad es revisar la cola de espera del sonido (Sound Queue), de ahí el llamarse SQ. Si no tiene muy claro lo que es la cola de un canal de sonido, repase los artículos que tratan sobre el tema en anteriores números de la revista.

SQ se utiliza para obtener información sobre el estado en que se encuentra un canal. Puede decirnos, por ejemplo, si está sonando una nota por él —no siempre nos resulta evidente si estamos utilizando tres canales a la vez— cuántos espacios libres hay en la cola o si la primera nota está retenida o citada con otra.

Tener toda esta información es extremadamente útil cuando estamos intentando comprender por qué nuestras «obras maestras» no funcionan tal como habíamos previsto.

SQ y bits significativos

Cuando quiera conocer algo sobre un canal use SQ, seguido del parámetro de canal entre paréntesis. De este modo, cuando necesite saber algo sobre el canal A utilice:

```
PRINT SQ(1)
```

y, por supuesto, pulse la tecla Enter. Como no tenemos ninguna nota sonando, nos encontra-

mos con que nos aparece en la pantalla el número 4. Encontraremos que con:

```
PRINT SQ(2)
```

y

```
PRINT SQ(4)
```

nos devuelve el mismo resultado. Bueno, el número es fijo pero, ¿que quiere decir? El significado de esta respuesta es:

- Que no hay ninguna nota sonando.
- Que no hay ninguna otra retenida al principio de la cola de espera del canal de sonido.

- La primera nota de la cola, si la hay, no está citada con otra de otro canal.

- Hay cuatro espacios libres en la cola de sonido.

Es mucha información la que hemos conseguido con un solo número, ¿verdad? La razón de que una única cifra pueda arrojar tanta luz sobre el estado de la cola de espera de un canal es porque el número devuelto por SQ es un «bit significativo». La Tabla 1 nos muestra lo que nos da a entender cada uno de ellos.

Número	Bit	Activado aparece:
1	0	Espacios en la cola
2	1	Espacios en la cola
4	2	Espacios en la cola
8	3	Citada con A
16	4	Citada con B
32	5	Citada con C
64	6	Primera nota retenida
128	7	Canal sonando

Tabla 1: Bit significativo

Así que vamos a deducir lo que quiere decir el número 4 que hemos obtenido con:

```
PRINT SQ(1)
```

cuando no está sonando ninguna nota por el canal A. De cualquier modo, ya que nos vamos a ocupar de analizar un bit significativo, tiene sentido usar la función BIN\$ para transformar el número en sus 8 dígitos binarios. De este modo, y siempre que no suene ninguna nota por el canal A, cuando usemos:

```
PRINT BIN$(SQ(1),8)
```

conseguiremos que nos devuelva:

```
0000100
```

que es la versión binaria del número decimal 4. La Figura 1 nos muestra al decimal 4 como un número binario y además nos indica la posición de cada uno de sus bits.

Número de bit	7	6	5	4	3	2	1	0
Byte binario	0	0	0	0	0	1	0	0

Figura 1: Posiciones de los bit del binario de 4

Si ahora combinamos el dato que hay en la Figura 1 con la información que nos da la Tabla 1 nos será posible enterarnos de muchas cosas acerca del estado del canal A partiendo de un sencillo 4.

Vamos a analizar el bit 7. Es un 0, por lo que deducimos que no hay sonando ninguna nota. Cuando lo esté haciendo alguna de bit, o dígito binario, deberá estar activado y el valor decimal correspondiente será al menos 128, e incluso puede ser mayor dependiendo de los otros bits que forman parte del byte.

El bit 6 es también 0, y nos dice que la primera nota de la cola no está retenida. Si el bit estuviera activado esto significaría que dicha nota está esperando a que el comando RELEASE la libere y entonces sonar.

También hay ceros en los bits de las posiciones 3, 4 y 5. Con esto, la función SQ nos está indicando que la primera nota de la cola de sonido, si es que la hubiese, no está citada con ninguna otra.

Si el bit 5 hubiese estado activado deduciríamos que la primera nota del canal está esperando para sincronizarse con otra del canal C. El bit 4 activado quiere decir que está citada con una del B.

Como ya puede suponer, un 1 en el bit 3 nos indica que la nota inicial está esperando otra por el canal A.

¿Cuántos huecos quedan en la cola de espera?

Los tres últimos bits se utilizan precisamente para señalar el número de huecos libres que hay en la cola de espera. En este caso los bits 2, 1 y 0 tienen el número binario 100, que es el decimal 4. Lo que demuestra, como podíamos esperar ya que no hay ninguna nota reproduciéndose, que hay cuatro espacios libres en la cola de sonido de un determinado canal.

Si los tres últimos bits fuesen 011, significaría que hay 3 espacios libres en la cola, si son 010. 2. Vamos a dejarle que adivine cuántos lugares sin utilizar hay cuando los tres últimos bits son 001 y 000.

Como podemos observar, el número devuelto por SQ contiene ya una gran cantidad de

información aunque el **Amstrad** no haya producido todavía ningún sonido.

Avanzaremos un poco para mostrar lo que la función **SQ** es capaz de hacer cuando el micro está «fabricando» sonidos, pero primero le aconsejamos definir la tecla Enter con:

KEY 139, "SOUND 135,0,0" + CHR\$(13)

Utilicela para limpiar los canales de notas extrañas cuando ¡llegamos a un atasco!

Y ahora, ¡a producir sonidos! Vamos a generar una nota en el canal A con:

SOUND 1,200,2000,7

y veamos qué puede decirnos **SQ** al respecto.

Si consigue hacerlo mientras está sonando la nota, averiguará que tecleando:

PRINT SQ(1)

le aparecerá en la pantalla el número 132 para desgracia suya. Imaginamos que estará de acuerdo con nosotros en que su equivalente binario nos da mayor información. Descubriremos un hábil: PRINT BIN\$(SQ(1),8) hará este trabajo por nosotros, devolviéndonos el número binario: 10000100. La Figura 11 nos muestra este valor junto con el correspondiente a cada bit.

Número de bit	7	6	5	4	3	2	1	0
Byte binario	1	0	0	0	0	1	0	0

Figura 11: Posiciones de bits para el binario de 132

Con ella y con la Tabla 1 seré capaz de ver que como el bit 7 está activo, el canal está trabajando. Ya que los bits 2, 1 y 0 contienen el número binario 100, no nos coge de sorpresa que haya cuatro sobrantes en la cola de sonido.

Ahora vamos a dejar dos notas sucesivas en el canal A y veamos qué nos tiene que decir **SQ**. Introduzca: SOUND 1,300,2000,7 seguido de: SOUND 1,300,1000,7 con lo que tendremos dos largas notas, la segunda de un tono más bajo.

Mientras esté sonando la primera:

PRINT SQ(1)

nos devolverá el número 131. Usando:

PRINT BIN\$(SQ(1),8)

nos da la expresión binaria de 131, mucho más útil, devolviéndonos: 10000011

Nos indica que el canal A está trabajando (el bit 7 está activado) y que sólo hay tres espacios libres en la cola de espera (bits 2 a 0). Se corresponde con lo que sabemos ya que podemos oír la nota y, puesto que la segunda aún no ha comenzado todavía estará en la cola. De los cuatro espacios libres originales ya sólo nos quedan tres.

Tan pronto como la segunda nota comienza a sonar, teclee:

PRINT BIN\$(SQ(1),8)

otra vez y verá que el resultado es ahora:

1000100

De nuevo el bit 7, conteniendo un 1, nos está diciendo que el canal permanece activo. Sin embargo, ahora que la primera nota ha terminado y empieza a sonar la segunda, volvemos a tener cuatro huecos desocupados en la cola de espera. Le recomendamos que coloque en ella dos, tres o cuatro notas y vea cuáles son los resultados en cada caso con:

PRINT BIN\$(SQ(1),8)

siempre vayan saliendo las notas de la cola y suenen.

Y, ¿qué hay de los bits 3, 4 y 5? Son los que nos indican si una nota está esperando para sincronizarse con una de otro canal o no. Limpie la cola de espera del canal A pulsando la tecla Enter y escriba:

SOUND 17,200,1000,7

Define una nota de tono 200, con una duración de 10 segundos y volumen 7 en la cola del canal A, citándola con una del canal B. La Tabla 11 podría refinar nuestra memoria sobre todo lo relativo al parámetro de canal.

Número	Bit Activ.	Resultado
1	0	Usa canal A
2	1	Usa canal B
4	2	Usa canal C
8	3	Citada con A
16	4	Citada con B
32	5	Citada con C
64	6	Retenido hasta RELEASE
128	7	Limpia el canal

Tabla 11: Parámetros de canal valores y acciones

Si ahora utilizamos: PRINT BIN\$(SQ(1),8) produce: 00010011

El bit 7 es cero, no hay ninguna nota que esté sonando, la primera nota no está retenida, ya que el bit 6 también está a cero. El bit 5 es 0 debido a que la nota no está citada con ninguna del canal C. Sin embargo, si lo está con una del B, por eso el bit 4 sí está activado. Por razones obvias el bit 3, que indica una cita con el canal A, está desactivado.

El número binario formado por los bits 2, 1 y 0 es 011, que quiere decir que hay tres huecos libres en la cola de espera. El cuarto espacio estará ocupado por nuestra nota que está aguardando desesperadamente una del canal B. Sáquela de este lamentable estado con:

SOUND 10,1,1,1

y pasemos a conseguir colocar una nota en el canal B para citarla con otra del canal A. Lo hacemos mediante: SOUND 10,200,1000,7

En esta ocasión: PRINT BIN\$(SQ(2),8) nos dará: 00001011

Fíjese en el 2 que está entre paréntesis de-

SONIDO

trás de **SQ**. Estamos relacionando la función con el canal B.

Por el momento no debemos tener dificultades en asociar el hecho de que porque la nota esté citada con una del canal A, se active el bit 3. Como en el caso anterior los tres últimos bits nos indican que hay tres espacios libres en la cola.

SOUND 17,1,1,1

proporciona a la nota la pareja adecuada.

Para mostrarnos el significado del bit 5, teclee:

SOUND 34,200,1000,7

que coloca una nota en la cola de espera del canal B, citándola con una del C. Ahora:

PRINT BIN\$(SQ(2),8)

nos da: 00100011

demostrando con ello que cuando citamos una nota con otra del canal C, el bit 5 se pone a 1.

Pasemos a examinar el bit 6. Se activa cuando la primera nota de la cola está retenida. Vamos a colocar una en la del canal A y pararla allí con: SOUND 65,200,1000,7 y preguntando por:

PRINT BIN\$(SQ(1),8)

obtendremos: 01000011

Como se puede ver, el bit 6 está activado, indicando que la primera nota del canal A está retenida.

Si ahora introducimos: RELEASE 1 observará que: PRINT BIN\$(SQ(1),8) devuelve: 10000100 cuando haya terminado.

El bit 7 actúa como un switch

Habría notado que de nuevo el bit 7 se activa cuando el canal está trabajando, esto es, cuando está reproduciendo la nota. Si realmente desea comprobar sus conocimientos sobre **SQ**, intente usarlo con canales en los que las primeras notas no sólo estén retenidas, sino que también estén esperando para sincronizarse con otras. Luego vea qué ocurre cuando liberamos esas notas de sus dobles ataduras.

SQ no sólo se utiliza para ver el estado en el que se encuentran y corregir las colas de espera de cada canal.

También se puede usar para solucionar ciertas pegas o inconvenientes con los que nos encontramos cuando utilizamos programas con sonido. El Programa 1 muestra lo que queremos decir.

Como verá cuando ejecute el programa, los comandos PRINT y SOUND no van acompañados. Los números avanzan hasta el seis y luego se aumenta uno cada intervalo de un segundo. Mientras tanto las notas están sonando y continúan después que aparezca el mensaje «Ready».

Recordará de antes que esto es el resultado de tener una cola de espera llena que retiene la ejecución del resto de instrucciones Basic.

Bueno, pues con SQ podemos remediarlo introduciendo en el micro un WHILE... WEND retardador que retenga el bucle mientras esté sonando la nota. Lo hacemos con una línea parecida a:

```
45 WHILE SQ(1) + = 127: WEND
```

```
10 REM PROGRAMA I
20 FOR ruido=1 TO 10
30 PRINT "Bucle número ";ruido
40 SOUND 1,10,ruido,100,7
50 NEXT ruido
```

Cuando la nota suena del bucle WHILE... WEND hace ciclos inútiles frenando al FOR... NEXT. Por supuesto que al finalizar la nota también lo hace el bucle retardador, y el programa continúa. Sigue con los mensajes y las notas escalonadas, pero ha retrasado el programa Basic.

El Programa II presenta otro problema.

Con él escucharemos 10 notas, una tras otra. Pero por un momento sponga que lo que ahora queremos es que esta maravillosa melodía esté sonando mientras el micro resuelve alguna operación matemática. La finalidad podría ser mantenernos entretenidos mientras dure, por ejemplo, una aburrida colección de sumas. ¿Cómo podemos hacerlo? Si lo que necesitamos es que nuestra melodía acompañe a la tabla de multiplicar del número dos puede probar añadiendo una línea semejante a:

```
25 PRINT 2*sonido
```

pero la idea no es del todo buena. Lo que buscamos es un método que tiene el **Amstrad** para poder hacer las dos cosas al mismo tiempo.

```
10 REM PROGRAMA II
20 FOR ruido=1 TO 10
30 READ tono
40 SOUND 1,tono,100,7
50 NEXT ruido
60 DATA 100,200,300,400,500
70 DATA 600,400,300,200,100
```

yo, tocando una canción y ejecutando un programa Basic. No es tan difícil como parece. Todo lo que necesitamos es el comando ON SQ() GOSUB. El Programa III nos enseña la manera de hacerlo.

En este caso van apareciendo una serie de mensajes mientras escuchamos una «dulce melodía» y todo se lo debemos a ON SQ GO-

SUB. Vemos el programa más detalladamente.

En la línea 20 aparece por primera vez el nuevo comando. Todo lo que hace es decirle al **Amstrad** que cuando haya un lugar libre en la cola de espera del canal A interrumpa momentáneamente lo que está haciendo y vaya a la subrutina especificada por el número de línea.

El número entre paréntesis detrás de SQ indica qué canal es el que vamos a utilizar para producir la interrupción. Tan pronto como se haya ejecutado la subrutina, el programa regresa a la línea donde se había quedado.

En otras palabras, ON SQ GOSUB activa una interrupción. Cuando se cumple la condición —un espacio libre en la cola de espera del canal apropiado— el micro detiene lo que estaba haciendo para luego saltar a la subrutina especificada, obedece las instrucciones que allí encuentra, y regresa al lugar donde se había quedado. Mientras la subrutina sea corta y las interrupciones escasas y separadas entre sí, el usuario casi nunca notará los cortes.

```
10 REM PROGRAMA III
20 ON SQ(1) GOSUB 70
30 WHILE NOT verdadero
40 PRINT "El Amstrad está haciendo
dos cosas a la vez"
50 WEND
60 REM rutina produciendo sonido
70 READ tono
80 IF tono=0 THEN RESTORE:GOTO 70
90 SOUND 1,tono,100,7
100 ON SQ(1) GOSUB 70:RETURN
110 DATA 10,30,50,70,90
120 DATA 80,70,60,50,40,0
```

Puesto que la línea 20 ha activado una interrupción y como actualmente no hay nada esperando en la cola del canal A, el programa va inmediatamente a la subrutina de la línea 70. Allí lee un valor la variable «tono» de los datos incluidos en las líneas 110 y 120 y usa el comando SOUND de la línea 90 para reproducir una nota.

A continuación, mientras está sonando la nota el RETURN de la línea 100 devuelve al micro al programa principal, pero sólo después de activar otra interrupción con ON SQ GOSUB. Ahora espera hasta que haya un nuevo espacio libre. En realidad las primeras veces se pondrá en marcha enseguida ya que, al principio, hay cuatro espacios libres vacíos esperando llenarse. Después habrá que esperar hasta que una de las notas termine y aparezca un hueco en la cola.

Cuando ha terminado la interrupción el micro regresa al interminable WHILE... WEND formado por las líneas 30 y 50 que imprime un mensaje mientras está sonando la nota. Tan pronto como termina una de ellas y queda espacio libre en la cola, se dispara la interrupción y se ejecuta de nuevo la rutina de la línea 70, obsequiándonos con otra nota.

Hay que señalar una cosa importante. Tan

pronto como un ON SQ GOSUB ha enviado al micro a la subrutina, la interrupción se desactiva. Sólo funciona una vez.

Si desea usarla nuevamente es necesario emitir otro ON SQ GOSUB para que se active otra vez. Esta es la razón por la que la línea 100 tiene otro ON SQ GOSUB permitiendo de nuevo la interrupción. Cada vez que hemos llamado a la rutina, la interrupción se ha desactivado. La línea 100 se asegura de activarla antes de regresar al programa principal.

A propósito, si utiliza un comando SOUND o la función SQ en sus programas tendrá que reponer la interrupción después, aunque esté además desconectada.

Ahora intente omitir la línea 80 del Programa III y vea si es capaz de explicar lo que pasa. El micro nos dice que los datos que acabaron mucho antes de que hayan sido utilizados los 11 valores de *tono*. ¿Por qué?

La respuesta se encuentra en la línea 100. Continúa buscando más notas para colocarlas en la cola cuando aparezca un espacio vacante. Después que se haya metido 11 notas en la cola de espera todavía sigue buscando más. Debido a que ahora no está el RESTORE de la línea 80, ya que la hemos suprimido, aparece el mensaje:

DATA exhausted in 70

El resultado es que el programa «casca» (se interrumpe) antes que todas las notas hayan alcanzado el comienzo de la cola y hayan sido reproducidas.

Y éste es el momento en el que también nosotros nos hemos quedado sin datos. Consultando todo lo que se relaciona con el sonido comprobamos que no se nos queda nada en la cola de espera, excepto el Programa IV que pone en acción de nuevo a SQ y ON SQ GOSUB precisamente.

Intente averiguar cómo funciona, y después pruebe usar todo lo que ha descubierto en esta serie para escribir su propia música.

Aprenderá mucho más experimentando con los comandos de sonido de su micro que leyendo y estudiando sobre el tema. Y se divertirá un montón, que al fin y al cabo es lo que hay que tratar a la hora de ponerse a programar.

Propónganos algo simpático y lo veremos en **MICROHOBBY AMSTRAD**. Por nuestra parte le enviamos un saludo.

```
10 REM PROGRAMA IV
20 ON SQ(1) GOSUB 70
30 WHILE NOT verdadero
40 PRINT "El Amstrad está haciendo
dos cosas a la vez"
50 WEND
60 REM rutina produciendo sonido
70 READ tono
80 contador=contador+1
90 SOUND 1,tono,100,7
100 IF contador=100 THEN ON SQ(1) GOSUB 70
110 RETURN
120 DATA 10,30,50,70,90
130 DATA 80,70,60,50,40
```

INOVADAD! PARA AMSTRAD 464-664-6128-8256-8512

MASTER-RENTA

8512-14.900
8256-14.900
6128-14.900

MASTERCOM

8512-19.900
8256-19.900
6128-19.900

MASTERGEST

8512-14.900
8256-14.900
6128-14.900

MASTERBLOCK

8512- 6.900
8256- 6900
6226- 6.900
464 - 2.900

MASTERTXT

6128- 4.800
464 - 3.800

MASTERCOPY

6128- 3.900
464 - 2.900

MASTERPROFE 1

6128- 2.900
464 - 1.900

MASTERQH

8512- 3.900
8256- 3.900
6128- 3.900
464 - 2.500
MSX - 2.900

MASTERBINGO

6128- 2.900
464 - 1.900

MASTER-RULETA

6128- 2.900
464 - 1.900

MASTERHOROSCOPO

6128- 3.600
464 - 2.300

MASTER-RELOJ

6128- 2.500
464 - 1.500

Realiza las declaraciones de la Renta, tanto ordinarias como simplificadas, pudiendo cubrir los impresos oficiales o realizar un listado de los datos, tanto en pantalla como por impresora. Realiza todos los cálculos en 1 minuto.

Gestor de efectos comerciales. Contempla descuentos de remesas mínimos, impagados, líquidos, límites de descuentos, etc. Por pantalla o por impresora. Clasifica vencimientos, clientes, plazas y estudio de costes financieros de las remesas.

Control de cuentas corrientes de bancos. Controla todos los movimientos realizados, ingresos, pagos, etc., pudiendo conocer el saldo en cualquier momento y en el formato del recibo del banco con el que esté trabajando en ese momento. Por pantalla o por impresora. Saldo general de todos los movimientos y todos los bancos, balance general.

Agenda telefónica con directorio. Con búsquedas por Nombre, Dirección o Teléfonos. Imprime etiquetas para sobres.

Utilizable en cualquier tipo de impresora, pudiendo seleccionar partes del texto en diversos modos de escritura: Subrayado, alargado, cambiar márgenes, tabulaciones, insertar caracteres o líneas, etc.

Copiador de pantalla en cualquier tipo de impresora compatible con AMSTRAD. Trabaja los 3 modos de pantalla, pudiendo elegir la zona de pantalla a copiar.

Programa educativo referente a figuras planas tales como triángulos, cuadrados, circunferencias, etc. y volúmenes tales como esferas, cilindros, pirámides, etc., explicando todas sus características.

Control de carreras de caballos con pronósticos tanto individuales como conjuntos entre varios caballos. Base de datos 200 caballos y 300 carreras. TAMBIEN DISPONIBLE PARA MSX.

Edita cartones, extracciones de bolas manual o automático, listado de premios y comprobación.

Es tan real que usted se encuentra envuelto en el casino de Montecarlo.

Su astrólogo particular:
Calcula su tabla de nacimiento según la hora, fecha y lugar de nacimiento, dándole datos sobre su personalidad. Tendencias del futuro. Algoritmos verdaderos.

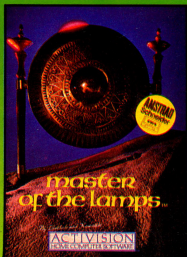
Reloj programable con alarma.

MASTERSOFT

Centro Comercial Sto. Domingo
Ctra. Burgos, km 28
Algete (MADRID). Tel.: 622 12 89

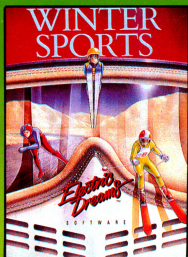
Buscamos distribuidores.
Envíos contra reembolso a toda España.

LOS MEJORES JUEGOS PARA AMSTRAD



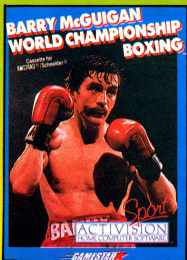
Nunca fue Aladino tan generosamente premiado por los genes. Puele sobre una increíble alfombra mágica en tres dimensiones.

CM



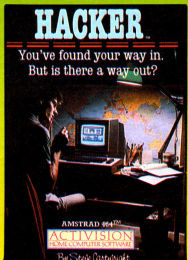
Si no puedes practicar tus deportes de invierno favoritos te ofrecemos oportunidad de hacerlo con tu ordenador. Ocho competiciones distintas te esperan. Aprovechate a equisarte y disfruta con el descenso, slalom, salto ski, etc.

S



Juego en el que puedes crear a tu propio boxeador. Elige su raza, estilo físico e imagen. Entrénalo y demuestra sus habilidades.

CS



Compleja aventura donde los jugadores deben buscar a través de las diferentes pistas y problemas como resolver el misterio.

CS

y además... **MINDSHADOW, GHOSTBUSTER.**

próximo lanzamiento
BACK TO THE FUTURE, ON COURT TENNIS

Disponibles para:

SPECTRUM S
COMMODORE C
MSX M

EN TIENDAS ESPECIALIZADAS Y GRANDES ALMACENES,
O DIRECTAMENTE Distribuido por:
POR CORREO O TELEFONO A: **PROEIN, S.A.**

Velázquez, 10 - 28001 Madrid Tels. (91) 276 22 08/09