

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO II N.º 44

160 Ptas.

Canarias 165 pts.

**CURSO DE
INTELIGENCIA
ARTIFICIAL,
¡DESPEGAMOS!**

**HAN LLEGADO
LOS GENERADORES
DE PROGRAMAS**

**NUCLEUS:
UN PROGRAMA
PARA OLVIDARSE
DE PROGRAMAR**

**DEFIENDE TU NAVE
DEL ASALTO DE
EL TEMIBLE DRAKE**

SOFTWARE

**Dam Busters,
cuando volar se hace
una realidad**



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.

POLY PLOT 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.

TYPING CRASH COURSE

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.

*** los 4 juntos
23.800 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

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.º Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

Fernando Choumel

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sapienza

Robert Chatwin

Francisco Portalo

Pedro Sudán

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalia Gómez

Alberto Suñer

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Chemo Sacristán

Portada

M. Barco

Ilustradores

J. Igual, J. Pons, F. L. Frontón,

J. Septien, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

Maria Andino

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

María 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 Calzado

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400

(Fuencarral) 28049 Madrid.

Teléfonos: Suscrip.: 734 65 00

Redacción: 734 70 12 *

Dto. Circulación

Paulino Blanco

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEC, S. A. Ctra. de Irún,

Km. 12,450 (MADRID)

Fotocomposición

Navocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 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 solidario de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

MICROHOBBY

AMSTRAD

sumario

Año II • Número 44 • 1 al 7 de Julio de 1986
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas. + 10 ptas. sobretasa aérea
Ceuta y Melilla, 155 ptas.

6

Primeros pasos

Continuamos avanzando en la comprensión de las instrucciones que involucran la lectura del teclado. **KEY e INKEYS** componen nuevamente los elementos que nos van a permitir un control total del teclado.

**18**

Para PCW...

Programar se está pasando de moda. Cada vez la importancia del programa final recae más y más en un análisis concienzudo, detallado y optimizado al máximo. Hasta ahora no había más remedio que perder el tiempo en tediosos tareas, desde ahora con **NUCLEUS**, generar programas en **BASIC** es ya una realidad. Descubre todos las ventajas e inconvenientes de este tipo de programas.

24

Inteligencia artificial

Comenzamos a caminar por la senda de la inteligencia artificial. Por el momento aprendamos a dominar un lenguaje imprescindible, el **LISP**. No dejes de leer esta sección y aprende desde el primer momento este potente lenguaje.

12

Código máquina

Analizamos los puertos del **Amstrad**: descubre cómo puedes hacer verdaderas diabluras «atacando» directamente al controlador de video y sonido.

28

Serie oro

Defiende tu nave del temible ataque de **DRAKE**. Prueba tu habilidad y reflejos con este programa de maravillosos gráficos.

16

Mr. Joystick

DAM BUSTER, un simulador de vuelo donde sentirás el rugir de los motores, donde tendrás que gobernar cada una de las partes de tu aparato. Por fin, busca a tu enemigo, apunta y dispara.



AMSTRAD SIN LIMITES



OLE, TORO



CAMELOT WARRIORS

El misterio, la fantasía y la más sofisticada técnica en diseño gráfico han hecho posible este clásico de DINAMIC.

Camelot Warriors. Imaginación sin límite.

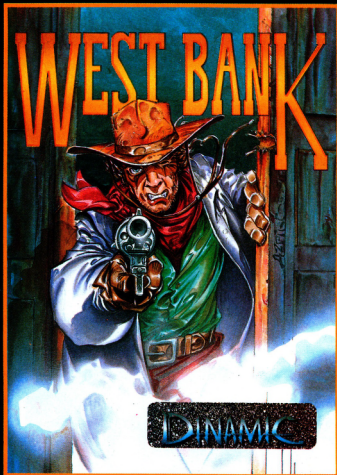
OLE, TORO

Tres elementos se dan cita en el rito de la arena: la vida, la muerte y el arte. Ole, Toro. Originalidad sin límite.

WEST BANK

La rapidez de reflejos es la clave de tu supervivencia, y los nervios de acero un seguro para la victoria.

West Bank. Adictividad sin límite.



EXPO SOFT

86

SALON DEL SOFTWARE
DE APLICACION
Y SISTEMAS INFORMATICOS

EXPOSOF 86

Del 10 al 12 de junio se ha desarrollado en el Palacio de Congresos y Exposiciones de Madrid, **EXPOSOF 86**, primer salón de software de aplicación y sistemas informáticos.

En un corto espacio de tiempo, hemos tenido la oportunidad de comprobar cómo cada vez se va difundiendo más en el mundo de la informática, la sana costumbre de tener informado al consumidor de todas las opciones y novedades que tiene en el mercado.

En esta ocasión el salón iba destinado a los programas de aplicación, como el entorno de periféricos que rodea al ordenador.

Como ellos mismos, **EXPOSOF 86**, indican en sus notas de prensa lo que se pretendía en esta ocasión era mostrar la «inteligencia» que mueve la máquina, no la máquina en sí.

Y esta «inteligencia» iba destinada, principalmente a los **PC** (Ordenadores personales), y evidentemente hablando de los **PC**, éstos estaban presentes en el salón de la mano de la poderosa **IBM** y de los compatibles de otras marcas.

Así, y de acuerdo con esto y con la potencia del mercado nacional en el terreno del software, vimos programas encaminados hacia sectores empresariales específicos y otros a colectivos profesionales determinados.

Y de entre las actividades profesionales para las que se ofrecían soluciones, estaban las destinadas a arquitectos, abogados, médicos, editores, etc.

Además de ofrecer programas para los sectores de enseñanza, automoción, transporte, salud, etc.

En lo que se refiere a **Amstrad**, vimos en el stand de PROA, S.A., dos programas de desarrollo vertical para el PCW 8256. Estos programas son el **HARMA**, programa de aplicación de arquitectura, para dibujar planos a escala y acotados. El de Presupuestos y Mediciones, también de aplicación para arquitectura, sirve para realizar mediciones y presupuestos de obras y trabajos en general.

Esta era la exigua representación de software para **Amstrad**.

Dentro del interesante campo del diseño gráfico por ordenador (**CAD-CAM**), del que ya hablaremos más adelante en el futuro, vimos funcionar el programa **AUTOCAD** de la casa **AUTODESK** en el stand de **SOFTRONICS**.



Los dibujos que pudimos observar, eran de tres dimensiones de una oficina como la que podemos encontrarnos en cualquier lugar, pero lo bonito era ver el desarrollo de los dibujos, así como la perfección de los mismos.

Y aunque el plano de una oficina realmente no causa mucha impresión, en los prospectos de difusión del programa, se podían dibujar de buques, piezas de maquinaria, y hasta una de las naves tipo Columbia, que demostraban la potencia de este programa de diseño asistido por ordenador.

Otro programa de **AUTODESK**, el **CAD-CAMERA**, es también sumamente interesante y convierte automáticamente dibujos en papel, en dibujos de fichero **AUTOCAD**.

CAD-CAMERA trabaja con una cámara exploradora y por lo tanto la exploración óptica no toma más tiempo que el trazado del mismo dibujo. Los planos se pueden almacenar en disco y posteriormente convertir a dibujos **AUTOCAD**, luego se podrán editar y trazar en la escala deseada.

primera PLANA

A nivel de difusión de las técnicas de la informática, **FUNDESCO** (Fundación para el Desarrollo de la Función Social de las Comunicaciones), tenía obras verdaderamente interesantes para todo aquel que quiera saber las repercusiones que está teniendo el proceso de informatización en la sociedad.

Y como muestra del interés de estos libros, tomemos por ejemplo algunos casos:

- **INTELIGENCIA ARTIFICIAL EN ESPAÑA.** Introducción y situación en España.
- **INFORMATICA Y ESCUELA.**
- **LA TECNOLOGÍA DEL SOFTWARE.**
- **LOS ORDENADORES Y EL CAMBIO SOCIAL.**

Pero aparte de éstos había otros también interesantes pero que harían la lista tal vez un poco larga.

Durante los días en los que transcurrió el salón, se celebraron conferencias tales como:

- **TRATAMIENTO DIGITAL DE IMAGENES.**
- **EL SOFTWARE DE LA SALUD.**
- **LOS SISTEMAS EXPERTOS: UNA NUEVA GENERACION DE SOFTWARE INTELIGENTE.**



Estas conferencias seguro que ayudarán a difundir temas sobre los que en España todavía estamos retrasados debido a un estudio e investigación notablemente deficitarios.

Pues nada más que contar de este **EXPOSOF 86**, sino decir que el próximo año no dudamos que la concurrencia al salón será mucho más importante que la de éste, que no ha sido muy numerosa y pensamos que probablemente esto haya sido así por la poca difusión que ha tenido este acontecimiento.

EL MOVIMIENTO ES LIBRE

Ya conocemos la forma de hacer que un cuerpo, forma o símbolo se deslice por la pantalla cuando se lo indicamos al ordenador. Para iniciar el movimiento basta con pulsar una determinada tecla y punto, el móvil irá en la dirección deseada.



Esto está muy bien, pero no nos conformemos. Intentaremos en esta ocasión dar «dimensión» al movimiento.

¿Qué queremos decir? Por el momento sólo lo hemos visto la manera de conseguir que nuestro «ovni», por ejemplo, se mueva una posición hacia la derecha cada vez que se presione la tecla «d» —por elegir una.

Y esto no es suficiente para lo que nosotros queremos. Además de la posibilidad de avanzar hacia la derecha, nos gustaría tener la opción de movernos en sentido contrario. ¿Cómo hacerlo?

¿Se acuerda de nuestro artículo anterior? Descubrimos que existe una función que permite leer un carácter introducido por el teclado, es decir, detecta si se ha pulsado o no una tecla y además genera un valor alfanumérico, o cadena, que consiste en el carácter de la tecla que se ha pulsado.

INKEY\$

es, precisamente, esa función y la vamos a utilizar para conseguir nuestro objetivo.

¿Nos ponemos en marcha? Para empezar teclee con cuidado el Programa 1. Es semejante al que publicábamos la semana pasada como Programa 4. Con él recordaremos la manera de aplicar en la práctica esta nueva instrucción.

Programa uno

Todo el «meollo» de la cuestión está en la línea 100. La función INKEY\$ estará explorando cíclicamente si se ha pulsado una tecla. Cuando así lo hayamos hecho, nos devolverá un valor alfanumérico que nosotros asignamos a la variable «tecla\$».

A continuación solamente analizaremos si el contenido de esta variable «tecla\$» coincide o no con el de la tecla elegida para movernos.

En nuestro caso hemos considerado que era bastante fácil asociar un movimiento hacia la derecha con la letra «d», pero podríamos fijar cualquier otra. Cambie la línea 110 pion-

do en ella la letra que más le guste y compruebe que el programa también funciona, pero ahora será necesario pulsar otra tecla si queremos que nuestro móvil se desplace como antes.

Intente hacerlo con mayúsculas, minúsculas o con cualquier signo de puntuación o carácter especial. Siempre obtendrá los mismos resultados (esperemos que positivos).

Cuando el valor alfanumérico devuelto por la función INKEY\$ coincida con la tecla elegida —«d»— el programa hará que aumente el valor de la coordenada «x» y, por tanto, Don Sonrisitas se desplazará un lugar hacia la derecha: se moverá.

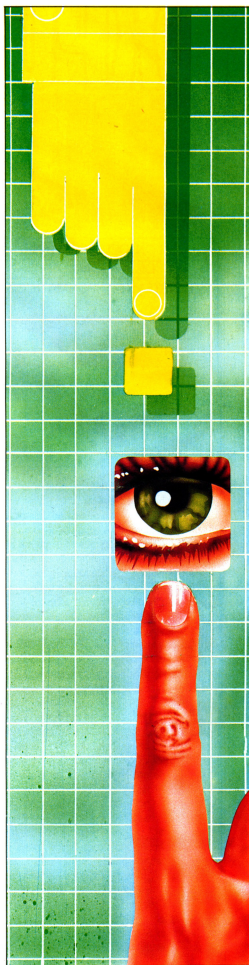
Pero, como hemos dicho antes, estamos decididos a dar más «movimiento» a nuestro amigo. ¿Qué le parece si ahora intentamos que vaya hacia la izquierda?

Como primera medida a tomar, elegiremos el valor de la tecla que, al pulsarla, hará posible que así sea. ¿Está bien la «x», de izquierda? Vale.

```
10 REM PROGRAMA 1
20 CLS
30 x=20
40 y=10
50 LOCATE x,y
60 PRINT CHR$(224)
70 FOR i=1 TO 500:NEXT i
80 LOCATE x,y
90 PRINT " "
100 tecla$=INKEY$
110 IF tecla$="d" THEN x=x+1
120 GOTO 50
```

Así que conseguiremos que el «CHR\$(224)» se pasee por la pantalla «caminando» hacia la derecha cuando presionemos la tecla «d» y en sentido contrario con la «x». Ya tenemos algo definido, sólo nos queda ver la manera de hacer que esto funcione.

Todo el problema radica en el tratamiento que hagamos de la coordenada horizontal —«x»— en cada caso. Cuando en el Programa 1 la función INKEY\$ detectaba que se había pulsado una tecla, después nosotros, o más bien nuestro ordenador, comprobaba si había sido la elegida. De acuerdo con esta comprobación aumentábamos o no el valor de «x», lo que implicaba que se produjera o no el movimiento. Una parte del problema está ya resuelto.



Primeros PASOS

El desplazamiento en sentido contrario, ¿qué supone? Si lo piensa un poco despacio verá que la respuesta no es tan complicada como pueda parecerle. Si hacemos avanzar un móvil hacia la izquierda, estamos «**disminuyendo**» el valor de «x». ¿Está de acuerdo? No olvide que el origen de coordenadas horizontales en la pantalla de textos, que es la que estamos usando, comienza por el valor 1 y en el margen izquierdo de la misma.

Conclusión. Si queremos de nuestro amigo que se «pasee» hacia la izquierda es suficiente con ver si hemos presionado la tecla escogida para que así sea, y después, en caso afirmativo, decrementaremos en 1 el contenido de la variable «x» que nos marca el valor de la coordenada horizontal.

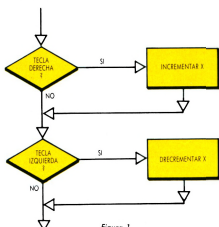


Figura 1

Fig. 1: Desplazamiento en dos sentidos.

La figura 1 nos muestra cuál sería el proceso lógico seguido para lograr que un cuerpo se desplace en ambos sentidos.

Y ha llegado la hora de ver si esto es así o hay algo que falla. ¡A codificar!

En esta ocasión nos parece que la solución es bastante sencilla. El Programa 2 es una versión del anterior, pero «corregida y aumentada».

Programa dos

Escribalo con su ordenador y atrévase a ejecutarlo. Recuerde que fijamos la tecla «d» para el movimiento hacia la derecha y al «i» hacia la izquierda. ¿Funciona? ¡Sí, claro!

¿Qué ha cambiado respecto al Programa 1? Hemos añadido solamente una nueva línea — la 120. Todo lo demás es absolutamente igual. En «tecla\$» está almacenado el valor devuelto por la función INKEY\$.

Con la línea 120 hacemos el tratamiento práctico con instrucciones Basic, de todo lo que hemos comentado anteriormente.

Si se cumple la condición que va detrás de IF:

tecla\$ = «i»

implica que hemos pulsado la letra «i» y que, por lo tanto, Don Sonrisitas se tendrá que mover hacia la izquierda. Por eso la «acción» que

ejecutaremos cuando así sea es la que va a continuación de THEN:

$$x = x - 1$$

con la que decrementamos en 1 el valor de la coordenada «x» y acercamos así a nuestro amigo al borde izquierdo de la pantalla. ¿Comprendido?

Ejecute de nuevo el programa. Ahora pulse la «i» continuamente a ver que pasa. Parece que todo es normal. El símbolo se va desplazando tranquilamente por la pantalla como si no ocurriera nada.

Pero al llegar al borde, su Amstrad le «obsequia» con un mensaje de error: ¡**Vaya!** Le aparece algo así como:

Improper argument in 50

y el programa se detiene. ¿Qué nos quiere decir con esto?

El mensaje de error que nos ha salido nos está indicando que en la línea 50 hay un argumento de una función o un parámetro de una orden cuyo valor no es correcto. Analicémoslo.

Lo primero que debemos hacer es ver qué hay en la línea 50, claro. Con:

LIST 50

tendremos en la pantalla un listado, o visualización, de dicha línea. Resulta que es:

```

10 REM PROGRAMA 11
20 CLS
30 x=20
40 y=10
50 LOCATE x,y
60 PRINT CHR$(224)
70 FOR i=1 TO 500:NEXT i
80 LOCATE x,y
90 PRINT " "
100 tecla$=INKEY$
110 IF tecla$="d" THEN x=x+1
120 IF tecla$="i" THEN x=x-1
130 GOTO 50
  
```

50 LOCATE x,y

¿Que ocurre entonces? Leyendo detenidamente el mensaje de error podemos deducir que el valor de un parámetro (o de los dos) de la instrucción **LOCATE** no es correcto.

Podemos imprimirlos y ver qué contienen. Teclee:

PRINT x

y le aparecerá en la pantalla el valor que tiene en este momento la variable «x».

A continuación escriba:

PRINT y

y se visualizará el contenido de la variable «y».

Así podremos comprobar si son buenos o no dichos valores.

Resulta que «x» contiene un 0 e «y» un 10. ¿Cuál es el incorrecto? Antes de contestarnos piense que estamos ante los parámetros de una instrucción LOCATE. ¿Recuerda que valores son los que esta orden puede utilizar sin error?

La coordenada «y», o número de la fila, puede estar comprendida entre 1 y 25 en el modo (o MODE) de trabajo que habitualmente utilizamos. Hemos dicho que en esta ocasión vale 10, así que parece que por su parte no habrá problemas.

Pero, ¿y la «x»? Esta coordenada tiene un valor correcto siempre que esté comprendida entre 1 y 40 (en MODE 1, claro), ya que son 40 las columnas que tiene la pantalla. ¿Cuánto vale ahora? Con la instrucción:

PRINTX

dada anteriormente encontramos que contenía «un cero». Y ahí está el error, ya que la coordenada «x» de una orden LOCATE no puede valer menos de 1.

Así que el programa se nos detiene y el ordenador nos da un mensaje de error en el momento que, al ir decrementando el contenido de la «x», llega a tomar un valor menor que 1, o sea cero en este caso, y utilizamos este parámetro en la línea 50:

50 LOCATE x,y

Habrà que corregir este error, ¿no? La manera más sencilla de hacerlo es impedir que se dé. Aunque esto parece una cosa demasiado evidente, creemos necesario analizarlo despacio.

¿Qué queremos decir con «impedir que se dé» esta situación? Precisamente lo que acaba de leer.

El error se da cuando a la instrucción LOCATE de la línea 50 le llega una coordenada «x» que vale 0. Impidámoslo antes de llegar ahí.

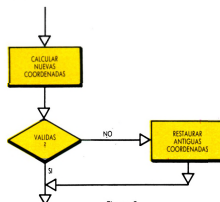


Figura 2

Figura II: Análisis nuevas coordenadas.

Es decir, calculamos el valor de «x» para el nuevo punto en el que va a estar colocado Don Sonrisitas. A continuación analizamos si la coordenada «x» recién calculada es válida o no.

Si es válida, ¡adelante! Pero si no lo es, al tener un valor igual a cero como vimos anteriormente, devolvemos a la variable el contenido que tenía y punto. La figura 2 nos presenta gráficamente los diversos caminos a seguir tras este análisis.

FIGURA II

Con esto «evitamos» el error antes que se produzca y así el programa «correrá» sin detenerse.

Ahora escriba y ejecute el Programa 3 y comprobará que esto ya es otra cosa.

Programa tres

Hemos añadido las instrucciones que van a impedir que se den las condiciones de error. Con la línea 130 cuando «x» toma el valor cero (no válido para LOCATE) lo convertimos en el anterior: 1. Del mismo modo con la 140 evitamos que ocurra algo semejante pero en el otro extremo de la pantalla.

Con estas condiciones podemos delimitar la zona de movimiento de nuestro objeto. En este caso puede moverse entre los valores de «x» comprendidos entre 1 y 40. Si tiene curiosidad haga los siguientes cambios en el Programa 3.

130 IF x=15 THEN x=16

y

140 IF x=25 THEN x=24

ejecútelos de nuevo. ¿Qué cambios ha observado?

El más evidente, y único, es que la zona en la que se desplaza Don Sonrisitas es mucho más estrecha que antes. La verdad es que no hubiera sido necesario ejecutar el programa para comprobarlo. Eche un vistazo a las modificaciones que hemos introducido y comprenderá por qué ahora solamente se va a mover entre las columnas 16 y 24. Imaginamos que no encontrará ninguna pega en ello.

Bueno, más cosas. Ya sabemos cómo mover un cuerpo por la pantalla detectando si se ha pulsado o no una determinada tecla. Pero sólo lo hacemos en horizontal. Si queremos que además suba o baje, ¿qué tendremos que hacer?

Como ocurre con casi todas las preguntas que le hacemos, en esta ocasión la respuesta también es muy sencilla: tratar la coordenada «y» de la misma forma que lo hemos hecho para la «x».

Detallemos todo esto. Como antes, primero habrá que determinar las teclas que nos van a producir el movimiento. Asignemos a la «a» la misión de «subir» y a la «b» la de «bajar».

Luego incluiremos en el programa unas instrucciones que nos analicen si hemos pulsado o no estas teclas. Cuando presionemos la «a» —o la tecla de subir— la coordenada «y» ten-

drá que disminuir ya que su origen está en el borde superior de la pantalla mientras que si lo hacemos con la «b» —o tecla de bajar— el valor de «y» aumentará.

Una vez calculada la nueva posición vertical del móvil, analizaremos si es válida o no para la instrucción LOCATE para, en el caso de que no sea, evitar que se produzca el error con su mensaje correspondiente y la detención del programa.

```

10 REM PROGRAMA III
20 CLS
30 x=20
40 y=10
50 LOCATE x,y
60 PRINT CHR$(224)
70 FOR i=1 TO 500:NEXT i
80 LOCATE x,y
90 PRINT " "
100 tecla$=INKEY$
110 IF tecla$="d" THEN x=x+1
120 IF tecla$="i" THEN x=x-1
130 IF x=0 THEN x=1
140 IF x=41 THEN x=40
150 GOTO 50
  
```

¡Ojo! Los valores de la coordenada «y» que resultan correctos son los comprendidos entre 1 y el 25, ya que la pantalla de textos está dividida en 25 filas cualquiera que sea el modo —o MODE— en el que estamos trabajando.

El Programa 4 es la solución práctica a nuestro problema.

Programa cuatro

Con la línea 100 detectamos si se ha pulsado una tecla y el valor que nos devuelve la función INKEY\$ lo almacenamos en la variable «tecla\$» como siempre.

```

10 REM PROGRAMA IV
20 CLS
30 x=20
40 y=10
50 LOCATE x,y
60 PRINT CHR$(224)
70 FOR i=1 TO 500:NEXT i
80 LOCATE x,y
90 PRINT " "
100 tecla$=INKEY$
110 IF tecla$="d" THEN x=x+1
120 IF tecla$="i" THEN x=x-1
130 IF tecla$="e" THEN y=y-1
140 IF tecla$="b" THEN y=y+1
150 IF x=0 THEN x=1
160 IF x=41 THEN x=40
170 IF y=0 THEN y=1
180 IF y=26 THEN y=25
190 GOTO 50
  
```

Las comprendidas entre la 110 y la 140 nos analizan si la tecla pulsada es alguna de las que nosotros hemos elegido para el movimiento. También actualizan las coordenadas correspondientes dependiendo de cuál de ellas ha sido presionada.

En las que van desde la 150 a 180 comprobamos que las nuevas coordenadas están den-

tro de los límites elegidos. Con ellas evitamos que la nueva posición del móvil nos dé error en el **LOCATE** de la línea 50.

Por lo demás, todo es igual a lo que ya hemos estado viendo pero ahora hemos conseguido mover a Don Sonrisitas por toda la pantalla.

Además de **INKEYS** hay otra función que nos permite detectar si se ha pulsado o no una tecla.

INKEYS nos va a devolver un valor distinto de -1 cuando examine el teclado y compruebe que se ha presionado una tecla determinada. No nos da el carácter asociado a la tecla, sino un valor distinto de -1 cuando así haya ocurrido.

Su forma más general es:

INKEY (núm. tecla)

donde la cifra que hay entre paréntesis es el número que el ordenador tiene asociado a cada una de las teclas.

Veamos un ejemplo práctico. El Programa 5 nos lo muestra.

Programa cinco

Es un sencillo bucle sin fin que lo único que hace es presentarse en la pantalla el valor que nos devuelve la función:

INKEY (55)

55 —el número entre paréntesis— es el valor asociado a la barra espaciadora. Por tanto, la función:

INKEY (55)

nos devolverá -1 mientras no estemos pulsando el espacio, pero en el momento que lo hagamos, nos dará un resultado diferente.

Ejecute el programa y compruébelo. Le proponemos que además de la **barra espaciadora** pruebe pulsar otras teclas, como la de mayúsculas por ejemplo, manteniendo ésta presionada. Anote los valores que aparecen en la pantalla y saque sus conclusiones.

```
10 REM PROGRAMA V
20 CLS
30 LOCATE 1,1
40 PRINT INKEY(47)
50 GOTO 30
```

Como dijimos antes, mediante esta función podemos **detectar** si está pulsada o no una determinada tecla. Si se cumple la condición de que el valor devuelto por esta función es distinto de -1 implica que sí ha sido presionada.

Con el Programa 6 se convencerá de ello. Es un bucle sin fin semejante al anterior.

Primeros PASOS

Programa seis

En este caso nos aparece en la pantalla un mensaje que nos comunica cuándo hemos pulsado el «espacio». En la línea 40 está la forma de hacerlo.

```
10 REM PROGRAMA VI
20 CLS
30 LOCATE 1,10
40 IF INKEY(47) <> -1 THEN PRINT "Tienes pulsado la barra espaciadora"
50 GOTO 30
```

Y le dejamos por el momento. Intente pasar todos los programas que utilizan la función **INKEYS** a otros semejantes pero empleando **INKEY**. Le diremos que los números de teclas asociados a las que hemos usado son:

s=tecla 60
b=tecla 54
d=tecla 61
i=tecla 35

¡Suerte y al teclado! La próxima vez, más.

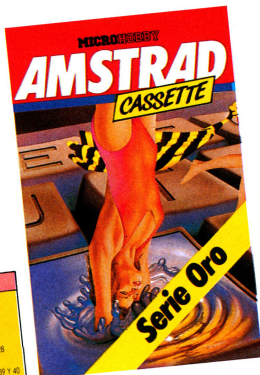
MICROHOBBY AMSTRAD SEMANAL

LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS, PARA QUE VD. NO TENGA QUE TECLEARLOS

Todos los programadores y aficionados a la microinformática sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y que no tengas que estar horas sobre el teclado de tu ordenador tratando de descifrar incomprensibles mensajes de error, **AMSTRAD SEMANAL** te ofrece cada mes los programas publicados de los cuatro números correspondientes en una cinta de cassette, sólo por **756 ptas.** (sin más gastos por envío).

Todos los programas de nuestras cintas se encuentran desprotegidos, con el objeto de facilitar su copia en disco y la revisión de los listados.

Envíanos con la menor demora posible, el cupón correspondiente.



Cinta n.º 10	
CASA 1	CASA 2
DISCMAN	AMSCAL*
EL MURO	ELECTIVO
BEGIN 371 A 377	RAM DISC '28
MACROCOMPILER*	COMPANI
ANALISIS 38	ANALIS 37, 39 Y 40
	IDEAS 37 A 39

* Sólo 664 y 6728

EXPLO-M. BAS

Por Juan Pérez García



Imprime en la pantalla una parte de la memoria RAM del **Amstrad** imprimiendo 8 bytes por línea.

A la izquierda se indica la dirección de memoria en hexadecimal del primer byte de la línea.

En el centro se imprimen los 8 bytes en hexadecimal.

A la derecha se imprimen los 8 bytes en ASCII, teniendo en cuenta sólo los 7 bits menos significativos, pero en rojo si el octavo bit está en 1 (esto permite leer fácilmente los nombres a los que se les ha puesto a 1 bit más significativo de algún carácter, como los nombres de variables).

El programa puede ser útil para aquellos que quieren explorar la memoria del **Amstrad**. El programa pide la dirección inicial (DESDE) y la dirección final (HASTA) a presentar. Si el listado de memoria es muy largo no se para al llenar la pantalla, pero podemos pararlo en cualquier momento pulsando una tecla y ponerlo en marcha volviendo a pulsar otra vez una tecla.

Ejemplo: Una buena zona a listar es desde 368 (&170) en adelante, ya que aquí es donde se almacena el programa Basic. Si queremos ver otro programa distinto que éste podemos hacer un RENUM 65515,1 después cargamos el programa a observar con CHAIN MERGE "(nombre segundo programa)",65515.

(Nota: la última línea del segundo programa deberá ser inferior a 65515.)

Explicación:

línea 70-100

Nos pide la zona de memoria a listar. Si damos el segundo valor menor o igual que el primero el programa nos lista la zona de memoria desde himem hasta &C000.

línea 120-140

Lista 8 bytes en hexadecimal.

línea 150-170

Lista los 8 bytes anteriores en ASCII.

línea 180

Si pulsamos una tecla el programa se para hasta que volvamos a pulsar otra vez una tecla.

Variables:

DESDE

Dirección de memoria inferior a listar.

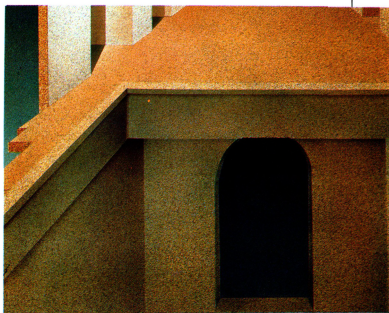
HASTA

Fin zona de memoria a listar.

i, j
n

Variables bucle for-next.
Valor del byte a listar.

AMSTRADIDEAS



```
10 *****
****
20 ** AMSTRAD:  EXPLORACION DE MEMO
RIA *
30 **   Por:  Juan Perez Garcia
*
40 *****
****
50 MODE 1
70 PRINT"ENTRAR DIRECCIONES A EXPLORAR"
80 PRINT STRING$(29,208)
90 INPUT;"DESDE:";DESDE:PRINT TAB(16);:INPUT"HASTA:";HASTA
100 IF DESDE>HASTA THEN DESDE=HIMEM:
HASTA=&C000:16-1:PRINT CHR$(11);"DESDE:";DESDE;TAB(16);"HASTA:";HASTA
110 PRINT
120 FOR i=DESDE TO HASTA STEP 8:
PEN 1:PEN 2:PRINT HEX$(i+j,4);":":
130 FOR j=0 TO 7:n=PEEK(i+j):IF n<128 THEN PEN 1:PAPER 0 ELSE PEN 3:
PAPER 0
140 PRINT " ";HEX$(n,2);:NEXT j:PRINT " ";
150 FOR j=0 TO 7:n=PEEK(i+j):IF n<128 THEN PEN 1:PAPER 0 ELSE PEN 3:
PAPER 0
160 IF n<32 THEN n=144
170 PRINT CHR$(1);CHR$(n);:NEXT j:PRINT
180 IF INKEY$<>"" THEN WHILE INKEY$<>"":WEND:WHILE INKEY$="" :WEND
190 NEXT i
200 PEN 1
```

En tu kiosk te espera algo muy inteligente

El AMSTRAD Especial número 2 incluye una cinta de casette adherida a la portada con un lenguaje Lisp completo que

te permitirá comprender y dominar las técnicas más complejas de inteligencia artificial. Por si fuera poco, en nuestra

cinta se incluyen también dos concursos: uno, de diseño gráfico de pantallas, para artistas, te permitirá ganar hasta 170.000 pesetas en premios. En el segundo regalamos un ordenador Amstrad CPC6128.

El número 2 de AMSTRAD Especial trata un amplio espectro de interesantes temas, como un comparativo de impresoras, que le ayudará a elegir la más adecuada a sus necesidades, como multitud de programas y rutinas de utilidad en lenguaje máquina como un paquete de soft integrado, con tres programas en uno y un largo etcétera que sería demasiado prolijo detallar.

Si no lo encuentras en tu kiosk, solicítalo directamente a nuestra Editorial.

MICROHOBBY
AMSTRAD
Especial
Revista Independiente para Usuarios de Ordenadores Amstrad Año I N.º 2

475 ptas.

SOFTWARE INTEGRADO
3 PROGRAMAS EN UNO

TABLETA GRAFICA
GRAFPAD II:
EL ARTE POR ORDENADOR
A TU ALCANCE

TE OFRECEMOS UN
LENGUAJE DE
PROGRAMACION LISP
COMPLETO EN CINTA
DE CASSETTE

NUESTRO PROGRAMA CREADOR
DE CRUCIGRAMAS DESAFIA A TU INGENIO

IMPRESORAS:
COMO HACER LA
MEJOR ELECCION

ATENCION A NUESTRO
FABULOSO CONCURSO:
PUEDES GANAR UN
CPC-6128 CON
SOLO CARGAR LA CINTA

Re llena este cupón y envíalo a **HOBBY PRESS, S. A.** Ap. de Correos 232. Alcobendas, Madrid.

Nombre _____ Apellidos _____
Domicilio _____ C. Postal _____
Localidad _____ Provincia _____ Fecha de nacimiento _____
Teléfono _____ Profesión _____

¿Eres suscriptor de **MICROHOBBY AMSTRAD**? Si ☐ No ☐
Deseo recibir el Especial de **MICROHOBBY AMSTRAD** n.º 2 al precio de 475 ptas. (IVA incluido)

FORMA DE PAGO
☐ Talón bancario adjunto a nombre de **HOBBY PRESS, S. A.**
☐ Mediante Tarjeta VISA. N.º _____ Fecha de caducidad _____
☐ Contra reembolso (supone 75 ptas. de gastos de envío).
Fecha y firma: _____

A 01

INSTRUCCIONES DE ENTRADA Y SALIDA

Con el presente capítulo, daremos por finalizado el estudio de las instrucciones utilizadas por el micropocesorador Z80, que lleva incorporado nuestro Amstrad.



Como estaremos hoy las instrucciones que se utilizan para leer datos de un puerto determinado.

La primera de este tipo de instrucciones, se representa de la forma:

IN A, (n)

El operado «n», se emplea en la mitad del fondo (A0 a A7) del bus de direcciones para seleccionar el dispositivo de entrada/salida correspondiente a una de las posibles 256 puertas.

El contenido del Acumulador (registro A), también aparece en la mitad superior de (A8 a A15) del bus de direcciones.

Entonces un byte de la puerta o terminal seleccionado se emplea en el bus de datos y se escribe en el acumulador en la CPU.

Si por ejemplo el contenido del acumulador es 23 y el byte 78 es accesible en el dispositivo periférico asignado a la dirección de terminal o puerta 1 de **entrada/salida**, tras la ejecución de

IN A, (01)

el acumulador contendrá 78.

La siguiente instrucción a estudiar viene representada de la siguiente forma:

IN r, (C)

El contenido del registro C se emplea en la mitad baja del bus (A0 a A7) de direcciones para seleccionar el periférico de entrada/salida a uno de los posibles 256 terminales.

El contenido del registro B se emplea en la mitad alta (A8 a A15) del bus de direcciones en ese momento.

Entonces se emplea un byte del terminal seleccionado en el bus de datos y se escribe en el registro «r» en la CPU.

El registro «r» identifica cualquiera de los registros de la CPU, (A, D, E, H, L, B, C).

Si por ejemplo el contenido del registro C es 7, el contenido del registro B es 10, y el byte 78 es accesible al dispositivo periférico asignado a la dirección del terminal de entrada/salida 7, después de la ejecución de

IN D, (C)

el registro D contendrá 78.

Instrucción INI

Otra de las instrucciones de entrada y salida, es la representada de la forma:

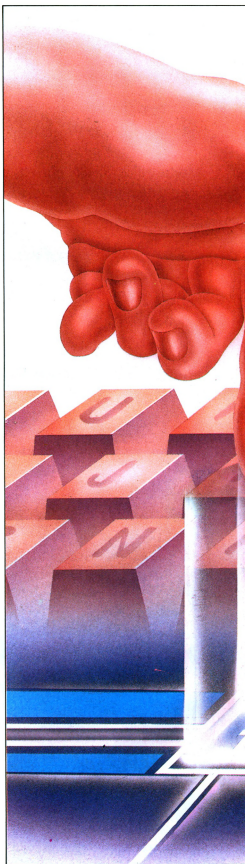
INI

El contenido del registro C se emplea en la mitad baja del bus de direcciones para selección el dispositivo de entrada/salida de uno de los terminales posibles.

El registro B se puede utilizar como contador de bytes, y su contenido se emplea en la mitad alta (A8 a A15) del bus de direcciones.

Entonces un byte del terminal seleccionado se emplea en el bus de datos y se escribe en la CPU.

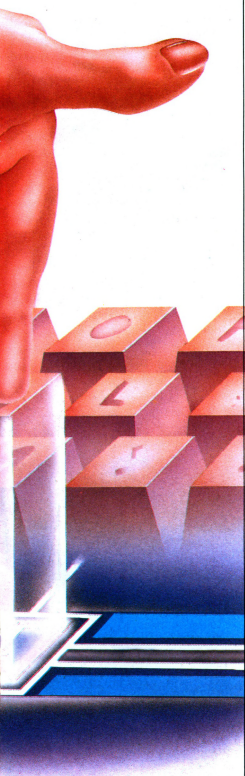
El contenido del registro par HL se emplea entonces en el bus de direcciones y el byte de entrada se escri-



be en la dirección de memoria correspondiente.

Finalmente se decrementa el byte «counter» y se incrementa el registro par HL.

Si por ejemplo el contenido del re-



INI
la dirección de memoria 1000 contendrá 78, el registro par HL contendrá 1001 y el registro B contendrá 9.

Orden INIR

Una instrucción similar a esta última es la que veremos a continuación, y que viene representada de la forma indicada a continuación:

INIR

Una vez ejecutada dicha instrucción, el contenido del registro C se emplacea en la mitad baja del bus de direcciones para seleccionar un dispositivo de entrada/salida en una de las 256 terminales posibles.

El registro B se usa como un contador de bytes, y su contenido se emplacea en la mitad alta del bus de direcciones.

Entonces se emplacea un byte del terminal seleccionado en el bus de datos y se escribe en la **CPU**.

El contenido del registro par HL se coloca en el bus de direcciones y el byte de entrada se escribe en la dirección de memoria correspondiente.

Una vez hecho esto, se incrementa el registro par HL y se decrementa el contador de bytes.

Si el decremento hace que B sea cero, la instrucción se da por finalizada. Si por el contrario B no resulta ser cero se decrementa en dos el PC y se repite la instrucción.

Si B se pone a cero antes de la ejecución de la instrucción, entrarán 256 bytes de datos. También se reconocerán las interrupciones después de cada transferencia de datos.

Por ejemplo, si el contenido del registro C es 7, el contenido del registro B es 3, el contenido del registro par HL es 1000, y la siguiente secuencia de bytes es accesible al periférico asignado con la dirección de terminal de **entrada/salida 7**:

51 99 03

entonces tras la ejecución de

INIR

el registro par HL contendrá 1003, el registro B contendrá 0 y las direcciones de memoria tendrán con contenidos que se indican:

DIRECCION	CONTENIDO
1000	51
1001	99
1002	03

registro C es 70 si el contenido del registro B es 10, si el contenido del registro par HL es 1000, y el byte 78 es accesible al periférico asignado a la dirección de terminal de **entrada/salida 7**, tras la ejecución de

Código MAQUINA

Instrucción IND

Otra de las instrucciones a estudiar es la que viene representada por:

IND

El contenido del registro C se emplacea en la mitad baja del bus de direcciones para seleccionar el dispositivo de entrada/salida.

El registro B puede utilizarse como contador de bytes y su contenido se emplacea en la mitad alta del bus de direcciones.

Ahora un byte del terminal seleccionado en el bus de datos y se escribe en la **CPU**.

El contenido del registro par HL se emplacea en el bus de direcciones y se escribe el byte de entrada en la dirección de memoria indicada por HL.

Finalmente se decrementan el contador de bytes y el registro par HL.

Si por ejemplo el contenido del registro C es 7, el contenido del registro B es 10, el contenido del registro par HL es 5000 y el byte 24 es accesible al periférico asignado con la dirección del terminal de entrada/salida 7, entonces tras la ejecución de

IND

la dirección 5000 contendrá 24, el registro par HL contendrá 4999 y el registro B contendrá 9.

Y, por último, INDR

La última de las operaciones de entrada y salida que vamos a estudiar es la que viene representada en la forma que se indica a continuación:

INDR

Una vez ejecutada dicha instrucción el contenido del registro C se emplacea en la mitad baja del bus de direcciones para seleccionar el dispositivo de entrada/salida correspondiente.

El registro B es utilizado como contador de bytes y su contenido se coloca en la mitad superior del bus de direcciones.

Una vez hecho esto se emplaça un byte de terminal seleccionado en el bus de datos y se escribe en la CPU.

El contenido del registro par HL se emplaça en el bus de direcciones y se escribe el byte de entrada en la dirección de memoria correspondiente (la indicada por HL).

Ahora se decrementan el registro par HL y el registro contador de bytes.

Si el decremento hace que B sea cero entonces finaliza la instrucción de lo contrario el PC se decrementa en dos unidades y se repite la instrucción.

Debemos tener en cuenta que si el registro B es puesto a cero antes de la ejecución de la instrucción, entrarán 256 bytes de datos. También se reconocerán las interrupciones después de cada transferencia de datos.

Si por ejemplo el contenido del re-

gistro C es 7, el contenido del registro B es 3, el contenido del registro par HL es 5000 y la siguiente secuencia de bytes es accesible al periférico asignado:

20 54 09

entonces después de la ejecución de INDR

el registro para HL contendrá 4997, el registro B contendrá 0 y las direcciones de memoria correspondientes tendrán los siguientes contenidos.

DIRECCION	CONTENIDO
4998	20
4999	54
5000	09

Un ejemplo típico de este tipo de instrucciones, es la realización de la lectura de teclado, como podemos ver en el ejemplo del programa número 1. En dicho ejemplo realizamos una lectura de teclado, y si se está pulsando la tecla «M», este es imprimida en pantalla.

```

Hisoft GENA3.1 Assembler. Page      1.

Pass 1 errors: 00

A000          10          ORG  #A000
                20 ;
                30 ;LECTURA DE LA TECLA 'M'
                40 ;
A000 3E44      50 INIC:  LD  A,#44
A002 F3        60      DI
A003 0192F7    70      LD  BC,#F792
A006 ED49      80      OUT (C),C
A008 06F6      90      LD  B,#F6
A00A ED79     100      OUT (C),A
A00C 06F4     110      LD  B,#F4
A00E ED78     120      IN  A,(C)
A010 01B2F7    130      LD  BC,#F7B2
A013 ED49     140      OUT (C),C
A015 FB        150      EI
A016 CB77     160      BIT  6,A
A018 20E6     170      JR   NZ,INIC
A01A 3E4D     180      LD  A,"M"
A01C CD5ABB    190      CALL #BB5A
A01F 18DF     200      JR   INIC
A021 C9        210      RET
  
```

Pass 2 errors: 00

INIC A000

Table used: 24 from 129

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 podréis 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:

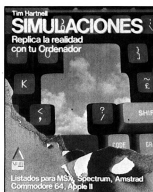
Nombre y apellidos,
dirección y teléfono.

Indicando claramente en el sobre:

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

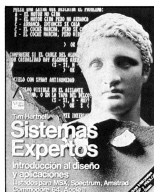
Ordena tus propias ideas

Le sacarás partido a tu ordenador



SIMULACIONES: REPLICA LA REALIDAD CON TU ORDENADOR

Tim Hartnell
1.643 ptas.



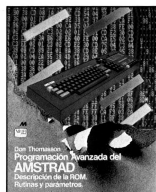
SISTEMAS EXPERTOS. INTRODUCCION AL DISEÑO Y APLICACIONES

Tim Hartnell
2.120 ptas.



RUTINAS EN LENGUAJE MAQUINA PARA AMSTRAD

Joe Pritchard
1.590 ptas.



PROGRAMACION AVANZADA DEL AMSTRAD.

Don Thomasson
1.166 ptas.



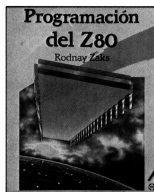
INTELIGENCIA ARTIFICIAL. CONCEPTOS Y PROGRAMAS

Tim Hartnell
1.484 ptas.



EL LIBRO GIGANTE DE LOS JUEGOS PARA AMSTRAD

K. Bergin y A. Lacey
1.378 ptas.



PROGRAMACION DEL Z 80

Rodney Zaks
2.915 ptas.



CODIGOS Y CLAVES SECRETAS: CRIPTOGRAFIA EN BASIC.

Listados para MSX, Spectrum, Amstrad, Commodore 64, Apple II
Gareth Greenwood
1.378 ptas.

- ☐ Les ruego me envíen el catálogo de su editorial.
☐ Les ruego me envíen los siguientes títulos:

TOTAL _____

- ☐ Adjunto talón bancario a
GRUPO DISTRIBUIDOR EDITORIAL, S. A.
☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).
☐ Giro postal.

Nombre _____
Profesión _____
Dirección _____
C. P. _____ Localidad _____
Provincia _____
PA _____

ANAYA
MULTIMEDIA

Adquíralos en su librería habitual.

Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a:
Apdo. de Correos 14632, Ref. D. de C. 28080 MADRID

ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA

DAM BUSTERS



Hasta ahora, todos los simuladores de vuelo eran parecidos. Dam Busters añade a la emoción de pilotar un bombardero, el aliciente de reproducir una histórica incursión en la Alemania nazi.



Son las 22.30 horas de la noche del 16 de mayo de 1943. En la cabina de mandos del bombardero pesado Lancaster, todo funciona sin contratiempos; rumbo noroeste nos aguarda la costa germana. Nuestra altura de vuelo ha sido cuidadosamente elegida; 700 pies por debajo de nuestras alas, la luna descubre las abruptas formaciones rocosas de la costa.

Por encima de los mil pies de altura, nuestro aparato sería detectado por el radar en tierra, o interceptado por las patrullas de caza nocturnas.

Una altitud inferior nos pondría en grave riesgo de colisión con obstáculos en tierra, especialmente al dirigirnos rumbo sur hacia el corazón de la **Alemania nazi**, donde abruptos valles y formaciones montañosas, rodean nuestro objetivo: las presas de *Moene*, *Eder* y *Sorpe*.

Somos un miembro del escuadrón 617 (el original Dam Busters), y nuestra misión es privar a las industrias del valle del Ruhr de su fuente más importante de abastecimientos de agua, asediando a la poderosa máquina de **guerra nazi**, un golpe del que jamás se podrá recuperar.

Cuando solamente hemos penetrado unas millas en la costa germana, un intenso haz luminoso surge de la oscuridad, escudriñando el cielo en nuestra busca; el artillero abre fuego sobre el punto en tierra, pero somos alcanzados de lleno por el haz, inundando de luz nuestra cabina.

Tras ésta, nuevas bases de búsqueda nocturna dirigen sus rayos hacia nosotros: esta vez

el artillero tiene más acierto y algunas de ellas se desvanecen en la oscuridad.

Los destellos luminosos nos permiten divisar las cadenas de globos cautivos, que nos obligan a efectuar un rápido ascenso; 800, 1.000 pies y en la lejanía algo se aproxima rápidamente hacia nosotros.

El artillero de cola, dirige instintivamente el cañón de su ametralladora FN 20 hacia el objeto no identificado; delante de nosotros, la silueta de un **Messerschmidt Me 110**, toma forma ametrallando el aire a nuestro alrededor.

El fuego de nuestras armas tiene éxito, y las llamas que desprende el aparato iluminan su caída a tierra. Tras su ataque, otros miembros de la escuadrilla se dirigen a nosotros, convirtiéndolo en un infierno la zona que nos rodea.

Cuando conseguimos deshacernos de ellos, la situación es desesperada, tres de nuestros motores arden en llamas y nuestros esfuerzos por apagarlos son inútiles, hemos llegado demasiado tarde, nuestro bombardero entra en pérdida y una terrible explosión envía nuestros restos a suelo alemán.

Menos mal que solamente se trata de un programa de ordenador, y no tenemos más que pulsar un tecla para intentar otra incursión.

Hemos conocido muchos juegos en los cuales el tema central es pilotar un avión, pero ninguno de las características y originalidad de éste.

En los anteriores programas, siempre pilotábamos cazas de combate, en los cuales sólo teníamos que preocuparnos de localizar a las amenazas en vuelo, ponernos a su cola y ametrallarlos hasta su explosión.



En **Dam Busters**, nace un nuevo concepto de simulador de combate, esta vez manejamos un bombardero pesado, y hemos de ocuparnos simultáneamente de las tareas que realizan sus seis tripulantes.

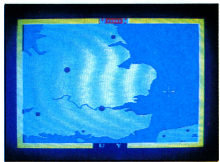
De esta forma somos a la vez piloto, copiloto, artillero de cola, artillero frontal, bombardero, y primer maquinista. Pero el hecho





que verdaderamente diferencia a este programa de los demás, es la elección de una verdadera misión de guerra y su reproducción con la mayor exactitud posible.

Lejos de los anteriores simuladores de vuelo, en los que despegar ya era un mérito, por la gran cantidad de indicadores y medidores a controlar, teniendo que utilizar para ello más



de una docena de teclas y en los que el joystick era demasiado impreciso para controlar el aparato.

Pilotamos un avión francamente sencillo de dominar; solamente hemos de utilizar el joystick y siete teclas que nos sirven para cambiar de puesto en la dotación del avión.

Una vez que estamos en determinada pantalla, el joystick se ocupa de todo; de esta manera cualquier tarea se realiza gráficamente, sin tener que sabernos de memoria un repertorio de más de 30 teclas, que sería el necesario para controlar este avión.

En este aspecto hay que señalar la facilidad con que se manejan los controles de las salas de máquinas; una vez en la pantalla de la sala de motores, nos encontramos con los cuatro medidores de inyección, los medidores de inyección de los cuatro motores, sus respectivos ocho palancas y las cuatro de los extintores de cada motor.

Con tanta palanca e indicador, su manejo en contra de lo que podría esperarse, es inmediato. Un toque de joystick, hace moverse un punto negro bajo el bloque de palancas.

Una vez situado bajo la que queremos accionar, basta con pulsar fuego y tirar el joystick hacia arriba o abajo, mágicamente la palanca se moverá en la dirección indicada.

En caso de que deseemos mover simultáneamente las cuatro palancas, para conseguir el mismo efecto en todos los motores, sólo tenemos que buscar la posición central con el punto negro, y a toque del joystick todas subirán suavemente.

Una técnica fácil y en la que no hay que pulsar ninguna tecla, con lo cual nos hacemos con el control del avión en cinco minutos. En otros programas necesitamos colocar una plantilla sobre el teclado del ordenador, debido al gran número de teclas usadas.

La posición normal de navegación es desde la cabina del piloto; en ella observamos lo que hay al frente de nuestra marcha, gobernando: brújula, altímetro, horizonte artificial y velocidad del aire.

En cualquier situación de peligro, un número intermitente nos indica la pantalla en la que se encuentra éste, y con un toque de tecla estamos en ella al instante.

Un caza alemán aparece, se divisa desde la cabina del piloto, el número dos aparece en el panel de mandos, un toque de tecla y estamos en la ametralladora de cola dirigiendo el punto de mira hacia nuestro blanco, fuego en el joystick y el caza cae en llamas.

Tenemos una sala de mapas, que contiene la región geográfica que va desde el sur de Inglaterra, hasta el centro de la **Alemania nazi**. En la cartografía, aparece la posición de nuestro bombardero y los principales puntos de defensa antiaérea, además de la posición de las presas que abastecen a la industria del Ruhr.

Cuando el globo, avión o haz luminoso aparece ante nuestros ojos, lo hace con realismo y auténtica sensación de acercamiento.

Mister JOYSTICK

El dibujo utilizado en las distintas salas de máquinas, es hiperrealista y representa los instrumentos de navegación a la perfección; el utilizado en las distintas cabinas de tiro, o el del piloto, está realizado con un gran sentido de la perspectiva, pareciendo realmente que nos encontramos dentro del aparato, en contraste con la oscuridad que nos rodea.

Un programa con una concepción totalmente nueva en los dedicados a aviones, fácil de manejar, basado en un hecho histórico, y en el que en todo momento nos vemos ocupados en las distintas salas del aparato; de la sala de máquinas a la de mapas, de la cabina del piloto a las distintas ametralladoras. Separándole mucha distancia de esos simuladores de combate, en los cuales después de haber tardado unas horas en despegar, estamos minutos y minutos persiguiendo a un supuesto **caza**, que media hora después aparece fugazmente ante nuestras narices. Para alejarse con extrema rapidez, o en el mejor de los casos aniquilarnos de una certera ráfaga de sus cañones ametralladoras, mientras nosotros buscamos inútilmente la tecla de los misiles termidrigidos, entre un **maremagnum** de dos docenas y media de teclas de control.

Con la diferencia esencial, de que nuestra última misión consiste en lanzar nuestras bombas, a la altura, velocidad y distancia exactas, para reventar los poderosos muros de hormigón de las presas alemanas.



NUCLEUS

Al adquirir un ordenador y plantear el tipo de Software que se necesita, las posibilidades van desde comprar un programa estándar hasta encargarlo a medida, o hacérselo uno mismo. En medio de estos extremos, diversos programas intentan facilitar la labor a los usuarios sin grandes conocimientos de programación, pero que no encuentran un programa a su medida, y no desean encargarlo por alguna razón, normalmente de tipo económico, dado el elevado coste de los mismos. Es en este sector donde se encuadra el programa NUCLEUS, y no es muy arriesgado decir que con un formidable porvenir.

N

úmeros anteriores de esta revista, en esta misma sección y en la de **Banco de pruebas**, han descrito distintos programas de Bases de datos con los que el manejo de grandes masas de información resultaba bastante simplificado. Alguno de ellos incluso proporcionaba suficientes funciones de programación para combinarlas y obtener programas para manejar dichos archivos de una manera compacta. Ya se ha dicho, que **NUCLEUS** estaba encuadrado en este segmento. Sin embargo, este programa no se parece en nada a los anteriores. Es uno de los pocos programas en el que una persona con escasa idea de programación y las ideas sobre el programa que necesita muy claras, puede conseguir resultados verdaderamente asombrosos. Eso es así, porque la filosofía del programa es totalmente opuesta a la de los otros.

Describir este programa resulta bastante más complicado de lo habitual. En realidad no es tal programa sino un paquete de programas, que vienen en **TRES DISCOS**, y con un voluminoso manual realizado en español. En honor a la verdad, debo decir que cuando empecé a leerlo, no me enteraba absolutamente de nada. Las cosas empezaron a aclararse cuando llevaba leídas «apenas» doscientas veinte páginas del mismo. A partir de ahí, leyendo y practicando, y después volviendo al principio del manual, los resultados que obtuve me dejaron impresionado. El problema no es que el manual esté mal escrito, sino quizás que está mal ordenado. Claro que como viene en una carpeta de anillas, nada impide cambiar el orden. Desde luego, el primer consejo es empezar por la página doscientos veinte, donde define lo que es el **corazón del programa**, y una vez entendido y practi-

FICHA TECNICA

NOMBRE	Nucleus.
TIPO	Programa de utilidades.
EQUIPO MINIMO	Amstrad PCW 8256.
EQUIPO ACONSEJADO	Amstrad PCW 8256 a con otra unidad de disco.
DISTRIBUIDOR	OFITES Informática. Avd. Isabel II, 16-80. 20011 San Sebastián.
PRECIO	26.780.

cado con esto, pasar a aprender las diversas posibilidades que proporciona.

¿Qué hace este programa? Simplemente programas. Realiza programas en **BASIC**, que además posteriormente se pueden listar, modificar y completar a gusto de cada uno, y ésta es su principal ventaja. Es evidente que en principio, no se puede hacer un programa que contemple absolutamente todas las posibilidades de programación en el área de la gestión, pero estos programas tienen una gran parte que resulta a nivel de planteamiento, idéntica. Esto es, la creación de ficheros, la modificación o actualización de los datos de estos ficheros, y la impresión de resultados, informes, etc... Todo esto, lo realiza **Nucleus**. ¿Cómo? Veámoslo.

Planteamiento

A la hora de empezar a crear nuestro programa con el **Nucleus**, lo primero que hay que hacer es apagar el ordenador. Después, se coge lápiz y papel, y se piensa lo que se pretende. Los datos mínimos que se van a necesitar y que son importantes, para no tener que repetir los procesos son: número de archivos, diseño de los registros de dichos archi-





vos, campos dentro de esos registros que están relacionados con campos de otros archivos, y el campo o campos por los que se quiere ordenar o clasificar estos archivos. La modificación de cualquiera de estos datos hace que la mayoría de los procesos que se hagan posteriormente, deban repartirse. Por tanto, estos datos son fundamentales. Ya con estos datos, se puede conectar el ordenador, y empezar a trabajar.

Los discos

Dado que el paquete contiene tres discos, es conveniente tener claro las funciones de cada disco:

1. **COMIENZO DEL DÍA:** Esta cara del disco es con la que se debe cargar siempre el programa. Contiene los programas que adaptan el programa al ordenador, permiten utilizar el disco virtual, y proporcionan al acabar el programa la manera de trasladar los datos del disco virtual a otro disco de datos. Esto es fundamental, y si se apaga el ordenador, antes de copiar los ficheros del disco virtual al de datos, todo lo hecho hasta ese momento desde que se empezó la sesión de trabajo se pierde. Este disco se necesita siempre. Los programas realizados con el Nucleus, también necesitan para funcionar que este disco se cargue antes.

2. **MANTENIMIENTO DE DEFINICIÓN:** Esta cara del disco, realiza la tarea principal. Crea un fichero de parámetros, donde el programa guarda las claves del programa, el menú principal de sociedades, y a través de él, se crearán los ficheros que luego se utilizan en el programa. Se definen los archivos, el diseño del registro de éstos y la posible relación que haya entre campos de distintos archivos. También, para que no falte un detalle, proporciona un listado de los datos introducidos, que será necesario en las etapas posteriores de la aplicación. Este listado se puede ver en una de las ilustraciones del artículo.

3. **CREACIÓN DE PROGRAMAS. ACTUALIZACIÓN:** A través del menú de esta cara se realiza el programa que creará los ficheros en el disco de datos, y el programa que permite actualizar de diversas maneras la información de dichos ficheros.

4. **GENERADOR DE INFORMES:** El menú de esta cara es uno de los que permiten crear programas que realizan informes escritos de diversos tipos, incluyendo la realización de etiquetas.

5. **CREACIÓN DE PROGRAMAS. CARTAS:** A través de este menú se posibilita el disponer de un editor de cartas o informes, de manera que se le pueda enviar la misma carta a cada cliente (por ejemplo) de manera personalizada.

6. **CREACIÓN DE PROGRAMAS DE DOCUMENTO:** Muy similar al anterior, en lo que respecta a la edición, este programa permite

Para... PCW

además la realización de operaciones con los campos numéricos, para realizar determinados documentos. El uso más claro de esta opción sería la realización de las nóminas de la empresa.

Hasa aquí una breve descripción de las posibilidades de esta aplicación. Pero este programa merece una revisión más exhaustiva. Vamos a contar paso a paso, la creación de un pequeño programa, explicando detalladamente no sólo lo que hacemos, sino también lo que hace el propio programa.

Descripción del programa

En todos los casos, lo primero que hay que hacer es conectar el ordenador con el disco de **CP/M Plus**, e iniciar el programa con la cara del disco que pone: **Comienzo del día**. El programa empieza preguntando y validando la fecha actual. Después, un mensaje permite cambiar de disco. Como hay que empezar por el principio, se introduce la cara del disco: «Mantenimiento de definiciones». Nos aparece el menú de «**Compañías**», donde se selecciona la opción 1, y pasa al menú del sistema, que es el que permite a su vez pasar al menú principal de esta cara: «Menú de programas». Este menú tiene una serie de opciones, que se van a usar en su totalidad.

La primera opción, crea el fichero de parámetros Nucleus, que conservará información sobre lo que se realice después. Esta información incluirá la descripción del sistema y todos los datos necesarios para acceder a los ficheros y programas siguientes.

La segunda opción es «**Mantenimiento de definición de sistema**». A través de ésta, se crea el nombre del conjunto de programas que se van a realizar después. A cada sistema, y se puede crear más de uno, se le da un nombre, un código de dos letras que serán posteriormente incluidas en los nombres de todos los programas y ficheros, como medio de identificarlos y distinguirlos de los que pudiera haber en otras aplicaciones, y un número a través del cual se selecciona este sistema en los primeros menús. En este proceso sólo hay que teclear el nombre y la clave del sistema que se va a crear, dado que el número lo proporciona automáticamente el sistema.

El siguiente paso sería ir a la tercera opción, «**Mantenimiento de definición de ficheros**», a través de la cual se crean los ficheros que se van a utilizar. Cada vez que se da de alta un fichero, el ordenador pide si el

COMPACT NUCLEUS MANT. DE DEFINICIÓN DE FICHERO 06s NUFNAME2 30/05/86

Número de Sistema 20 AMSTRAD SEMANAL
Número de fichero 1 EMPLEADOS

Número de campo 1
Nombre de campo N
Tipo de campo (A/V/F/I)
Número de caracteres
Número de líneas
Número de dígitos antes punto 2
Número de dígitos tras punto 0
Permitir valores negativos? N
Valor mínimo
Valor máximo
Proteger contra borrado? N

¿ Detalles correctos ? S

fichero va a ser Maestro o de Transacciones, y después se pasa a introducir, uno por uno, los campos de dicho fichero. Dado que estos ficheros son indexados, uno o más —hasta cinco— campos se deben indicar como campos claves, para la ordenación y localización de los registros. En este sentido, uno de esos campos clave debe ser siempre el primero que se introduzca. De cada campo, el programa nos pedirá el nombre, si el campo es numérico, alfanumérico, de fecha o de escritura arábiga. Según la contestación que se le dé a esta pregunta, el programa pedirá el número de caracteres y las líneas que ocupa en el caso de **alfanumérico**, y número de decimales, valor máximo y mínimo, y negativos en el caso numérico. Después pasa a pedir el siguiente campo, y así hasta que se dé por terminada la introducción de campos. Una opción muy interesante es la posibilidad de impedir por separado la modificación de algún campo. Una vez acabada esta tarea, el programa pide que se defina el o los campos que servirán de clave de indexación, pudiendo otorgar a estos claves otro nombre alternativo al que tiene en el fichero, y una longitud menor. Esto es particularmente útil para reducir el tamaño de las tablas de indexación, en el caso de desear clasificar un fichero por un campo cuya longitud total sea excesiva, pudiéndose así acortar dicha longitud sólo en la **indexación**. Los campos que se designen como clave son fundamentales, a la hora de poder relacionar los registros de un fichero con los de otro. Esto debe hacerse siempre a través de dichos campos. Por ejemplo, si tenemos un fichero donde se almacenan las horas extras realizadas por esos empleados, los dos ficheros se pueden relacionar mediante un campo que fuese el número de empleado. Pues bien, este campo deberá estar en los dos ficheros, y ser una clave en al menos uno de ellos. Una vez creados los ficheros que se vayan a necesitar, se vuelve al menú principal.

Si se tienen dos o más ficheros, y se desea conectarlos entre sí, hay que pasar después

por la opción cuatro: **«Mantenimientos de conexiones de ficheros»** donde se especificará el nombre del sistema, fichero y campo que se quieren relacionar con otro sistema, fichero y campo. La información que proporciona el programa es magnífica, dado que cada vez que se debe seleccionar un sistema, fichero o campo, aparecen en pantalla los nombres entre los que hay que elegir. Al menos uno de los campos que se seleccionen para relacionar debe haber sido especificado como campo clave.

La última opción de este menú imprime toda la información introducida hasta el momento. Esto es de suma importancia dado que muchas veces no se puede tener toda la información en la cabeza, y conviene disponer de un recordatorio. Imprime el diseño de los registros, nombres, conexiones con otros ficheros, y en fin, cualquier información que se pueda necesitar posteriormente.

Hasta aquí se ha completado la etapa más importante y vital de la creación del programa. Los datos hasta ahora introducidos, se podrían modificar en cualquier momento, pero entonces los procesos realizados a continuación deberían volverse a realizar, dado que no se actualizan automáticamente.

Creación de los programas de entrada

La mayoría de los programas tienen dos partes bien diferenciadas. Son el proceso de entrada de datos, y el proceso de salida de datos, una vez procesados. Para crear los programas que se encargan de la entrada de datos, se introduce la cara del disco etiquetada como: **«Creación de programas». Actualización.** Se pasa por los menús de Compañía y Sistema, y al llegar al menú de Programa aparecen tres opciones. Al seleccionar la primera, **«Crear programa de creación de ficheros»**, el programa pide que se seleccione el sistema, entre los que se hayan creado. Después aparece el número de líneas que el programa genera. En este momento, Nucleus está generando un programa en Basic, que creará los ficheros en el disco en el que se ejecute. Será pues, la primera opción del menú de este sistema. Si el menú no ha sido creado ya por otro programa, pregunta si se desea crear el menú, y en este caso la respuesta será afirmativa. El programa crea un menú para este sistema, donde irán apareciendo los nombres de los programas que se generen.

El siguiente paso sería crear un programa de actualización. A través de él se introducirán los datos en los ficheros ya creados. Primero pregunta para qué fichero, habrá que crear un programa. En caso de más de un fichero, habrá que crear un programa para cada fichero. Una vez elegido el fichero, presenta los campos del mismo, y pregunta si se quieren actualizar todos los campos. Esto es por si el valor de algún campo debe producirse automáticamente en vez de manualmente, en cuyo caso el programa no pedirá ese dato al hacer funcionar el programa. Una vez seleccionados los campos a actualizar, viene la fase en la que se diseña la pantalla de petición de datos. Para esto, el programa presenta cada campo con su posición de comienzo y final, y lo mismo para los datos que irán en dichos campos. Si no se está de acuerdo con esa pantalla, se puede modificar a voluntad. El

COMPACT NUCLEUS MANT. DE DEFINICIÓN DE FICHERO 03s NUFNAME3 30/05/86

Número de Sistema 20 AMSTRAD SEMANAL
Número de fichero 2 ENTRADAS

Número de clave 1
Nombre de la clave principal Número Empleado
Selección campo 1 Número Empleado
Número de caracteres

Número Empleado
FECHA
Salario Base
Horas Extra
Total Salario

¿ Detalles correctos ? S

programa avisará si algún dato se solapa con otro. Una vez conforme con los datos presentados, se puede realizar una prueba de la pantalla, y si no se está de acuerdo al verla, se puede volver a la pantalla anterior y modificarla. Una vez dada la conformidad, el programa pregunta si será un programa permanente o temporal. Algunos programas no tienen por qué ser permanentes. Se puede necesitar realizar algún informe que luego no tenga utilidad, y no hay por qué ocupar espacio en el disco innecesariamente, en cuyo caso se le da la opción temporal. Una vez que se apague el ordenador, este programa se borrará. Si se elige que sea permanente, Nucleus pregunta «Descripción», y el nombre que se le dé será el que aparezca en el menú de este sistema. No obstante, en cualquier momento se puede modificar cualquier menú, de una manera de la que se hablará posteriormente.

Si se actualiza un campo que haya sido previamente definido como campo de conexión entre los ficheros, se podrá optar por presentar y utilizar datos del registro del campo conectado, que será localizado automáticamente. Esto da la posibilidad, entre otras cosas, de tener en un fichero maestro los datos personales e introducir en otro fichero maestro los datos personales e introducir en otro fichero de transacciones otro tipo de datos, tales como facturas, registrar las horas..., y así, cuando se dé en el fichero de transacciones el número de cliente o empleado, el programa lee del fichero maestro unos datos y del de transacciones otros de una manera casi instantánea.

Antes de acabar con el apartado de introducción de datos, hay que explicar que los menús que aparecen en la aplicación y los que genera con nuestros programas se pueden modificar. Para hacerlo, cuando pide el número de opción hay que introducir «MOD». Nucleus impide la modificación de éstos por personas autorizadas mediante el uso de palabras clave. Al solicitar la modificación de algún menú, pide la palabra clave. Si no se ha modificado la clave, ésta es en todos los menús «XXX» pero se puede modificar por cualquier combinación no mayor de seis caracteres. Mediante estas claves también se puede impedir el acceso a determinadas opciones de cualquier menú creado. Es fundamental no olvidar la clave creada, dado que no hay manera de visualizarla y se podría inutilizar algún programa.

Creación de los programas de salida

Una vez que ya se tienen los programas y ficheros destinados a introducir los datos en el archivo, pasamos a la segunda parte. Esto es, sacar dichos datos mediante la impresora, según nuestras necesidades. Para esto, Nu-

cleus dispone de tres programas distintos, aunque su fin es parecido.

El primero está en la cara etiquetada como «Generador de Informes». Este programa está destinado principalmente al listado del contenido de ficheros, y tres opciones permiten seleccionar el tipo de informes deseado, con bastante flexibilidad. La primera opción es

distintas, y que las cumplan todas, o que cumplan alguna de ellas. Después de seleccionar los registros que serán impresos, pregunta si se desea línea de totales, si ésta debe estar al final de cada página, también cada vez que cambie la clave, o sólo al final del informe. Posteriormente, pregunta de qué campos realiza dichos totales, pudiéndose seleccionar los

Fecha 05/06/86 P. 19 de 1

CLAVE Data Company

COMPACT NUCLEUS		DEFINICION DE FICHERO	
NÚMERO DE SISTEMA	20	MICROHOBBY AMSTRAD 1	
NÚMERO DE FICHERO	1	MAESTRO.EMP	
NOMBRE DE FICHERO B SIC0 MAFIL001			
TIPO DE FICHERO Maestro			
LONGITUD DE REGISTRO 32			
NÚMERO DE RANGOS 3			
NÚMERO DE TIPOS DE REG. 1			
PROTECCIÓN DE FICHERO Ninguna			

CAMPO	NO. DESCRIPCION	TIPO DE CAMPO	NÚMERO DE CARACTERES	DISTO. ANTES DE PUNTO	DISTO. TRAS PUNTO	PERMITIR NEGATIVOS	MÍNIMO VALOR	MÁXIMO VALOR	NÚMERO DE LINEAS	BORRAR COM. PROTES.	COM. BYTE	BASIC LON. NOME.	
1	Nº Referencia	Numerico		2	0	N					1	2	CON
2	Nombre	Alfanumerico	20								3	20	CON
3	Departamento	Numerico		1	0	N					2	2	C16
4	Sueldo Base	Numerico		5	0	N					25	4	CON
5	S. I. R. P. F.	Numerico		2	2	N					29	4	C11

CL. CLAVE	NÚMERO DE CARACTERES	RAN. NO.	CAMPO NO.
1 Nº Empleado	2	1	1
2 Empleado	10	2	2
3 Clave S.	5	3	4

DETALLES DE CONEXION					
NO. NÚMERO DE CAMPO	CONECT.	NOM. DE SISTEMA	NOMBRE DE FICH.	TIPO DE REG.	NOMBRE DE LA CLAVE
1 Nº Referencia	conect.	MICROHOBBY AMSTRAD 1	MOVIENTOS		Empleado C1.

«Crear programa de Impresión M/F», y realiza un programa que lista totalmente un fichero dado. Esta opción no permite elegir ningún formato, imprimiendo el fichero con todos los campos y cabeceras.

La segunda opción es «Crear programa de reporte seleccionado». Esta opción permite elegir los campos del registro que se desean imprimir, y si algunos de estos campos son campos conectados con otros ficheros, permite también imprimir datos correspondientes a ese otro fichero. Una vez seleccionados los campos a imprimir, aparece una pantalla que describe la manera en la que el programa distribuye los campos en el papel. Después, pregunta si se desea modificar la posición de impresión de algún campo, y cuando se está conforme, realiza una prueba de impresión. Una vez dada la conformidad definitiva, permite decidir por qué campo se desea que salga el informe, y la parte más importante de esta opción, que es seleccionar los registros que van a ser impresos. Para esto, presenta una pantalla desde la cual se pueden elegir las condiciones que deben cumplir los registros a imprimir. Estos se refieren principalmente a que los campos de los registros sean mayores, menores, iguales..., a otros campos o a alguna expresión. Se permiten hasta tres condiciones

que se considere oportuno. Como se puede ver, los informes que se pueden hacer mediante esta opción no pueden ser más completos.

La última opción de esta sección es la impresión de etiquetas. Para esto lo primero que se hace es definir el tipo de etiquetas que se van a usar, longitud y ancho de las mismas en pulgadas, número de etiquetas por línea, y naturalmente el fichero del que se van a imprimir y los campos de dicho fichero que se van a imprimir. Si se imprime algún campo de conexión con otro fichero, permite imprimir también datos de dicho fichero, y además avisa si algún campo de los dados para ser impresos, excede de la longitud de la etiqueta. Naturalmente, después proporciona una prueba de impresión, y permite elegir el campo por el que se van a imprimir ordenadamente las etiquetas, y qué registros se deben imprimir, de manera idéntica a la usada en el reporte seleccionado, que se ha explicado anteriormente. Hasta aquí el programa contenido en la cara del disco «Generador de informes».

Pasando a la siguiente cara de disco, ésta está etiquetada como «Creación de programas de Cartas». La utilidad de este programa está en la posibilidad de disponer de un formato de carta, con algunos campos

que se rellenen automáticamente con datos de los ficheros. Se dispone para ello de un pequeño editor de textos, manejado por teclas de función que son definidas por el usuario. Para definir o modificar dichas teclas se usa la opción uno del menú, «Corregir parámetros edición». Una vez adaptado el editor a nuestros gustos, se pasa a la opción dos, «crear programa de cartas, donde se elige el fichero y campos cuyos datos se van a fusionar con la carta, el orden en el que se imprimirán las cartas, y seleccionar igual que en los programas anteriores qué registros van a ser impresos. Una vez hecho esto, la definición del programa de cartas estará terminado, y se volverá al menú. Entonces habrá que crear el texto de la carta, a través de la opción tres, **«Crear y modificar texto de cartas».**

La pantalla muestra entonces los campos de los ficheros dispuestos para ser fusionados, junto con su longitud y el código que debe incluirse en la carta para que lo sustituya por dicho dato. Después se entra en el editor de Texto. Primero se definen los márgenes de impresión, y ya se está listo para describirla de la misma manera que si se estuviera ante una máquina de escribir. La pantalla presenta la carta, el número total de líneas imprimibles, número de líneas hasta el final de la página, líneas no imprimibles, etc. Al pulsar la tecla STOP aparece en la parte superior de la pantalla la lista de opciones de texto disponibles, y que son las normales en este tipo de programas. Además, avisa si no se deja espacio suficiente para fusionar los campos del registro. Naturalmente, también prevee la posibilidad de hacer una prueba de impresión.

A través de esta opción también se puede modificar una carta que ya exista, pero para borrar una carta hay que seleccionar la opción cuatro, borrar carta. De esta manera se elimina la carta del disco, liberando espacio en él.

La última cara del último disco viene en la aplicación Nucleus está etiquetada como **«Creación de programas de Documento».** Los programas de esta cara del disco están pensados para tanto rellenar impresos con campos de los ficheros de transacción, como para generar documentos que requie-

ran de dichos datos, y de operaciones que se realicen con ellos. Un ejemplo claro podría ser realizar la nómina de una empresa, en la que figuraría el nombre del empleado, los diversos conceptos por los que cobra, las deducciones y el sueldo neto. Para ello se dispone de las mismas facilidades de edición de texto y selección de ficheros y campos que en los programas anteriores, proporcionando las líneas de total necesarias, en los campos que se le indiquen etc...

Todavía hay más

Si bien esto es todo el programa, no se ha dicho todo de él. Desde luego, una descripción más exhaustiva llevará toda la revista, pero hay características que no se pueden dejar en el tintero. Por ejemplo, si usted ha leído este artículo buscando un programa que realice esto y lo otro, y al llegar a este punto no ha encontrado determinada característica, no se desilusione. La estructura de esta aplicación no sólo permite sino incluso invita a que los programas realizados se completen. Naturalmente, para completarlos hace falta saber **BASIC**, pero dado que el trabajo sucio lo da hecho, no resulta difícil completarlo o modificarlo li-

geramente. Esto es fácil dado que todos los programas que Nucleus realiza son en Basic, no están protegidos, y la información necesaria se puede obtener de una u otra manera.

La estructura de los menús está muy bien conseguida, y siempre valida los datos introducidos, evitando así cualquier error. La posibilidad de introducir claves, resulta un gran acierto. La información suministrada por pantalla resulta excelente, no ofreciendo ningún problema en este aspecto.

El único problema con el que se pueden encontrar es si su ordenador sólo dispone de una unidad de disco. En ese caso, se utiliza el disco virtual como disco de almacenamiento, y sobre todo al principio, hasta que se coge la costumbre de interrumpir la tarea de vez en cuando para hacer Backup de dicho disco en otro de datos, el disco virtual se llena fácilmente, y el programa se queda colgado, perdiendo el proceso que se esté realizando. Es quizás lo único con lo que hay que tener especial cuidado. Los afortunados que dispongan del nuevo **PCW 8512** con dos unidades de disco disfrutarán de ese programa en un grado mucho mayor, hasta el punto de que muchos usuarios del 8256 se pensarán seriamente el adaptar la segunda unidad.

La impresión sacada del uso de este programa es francamente buena. No es que lo pueda hacer todo, pero este programa logra eliminar el trabajo más repetitivo y pesado de la programación de gestión. Por otro lado, es fácilmente adaptable para la realización de un sinfín de tareas, y al alcance del usuario medio. No se aprende a manejar en diez minutos, pero una vez que se domina se puede competir fácilmente con programas comerciales. Lástima que el manual de instrucciones que le acompaña no resulte tan bueno. Hay facetas que es mejor aprender probando, pues se tarda menos tiempo que leyendo. No obstante, esto quizás es el resultado de una mala estructuración de dicho manual, no del contenido del mismo. Pero el resumen del programa y del manual es muy bueno.

COMPACT NUCLEUS EDITOR DE DOCUMENTOS NUCLEUS 02s NUPOR2 06/06/86

1//// LIN. NO IMPRIMIB. //// 9:

10

Longitud de página 10

Primera línea imprimible 10

Ultima línea imprimible 10

10: ¿Detalles correctos? (S/N) ■

COMPACT NUCLEUS CREAR PROG. DE REPORTE SELEC. 05s NUPOR2 06/06/86

NO	DESCRIPCION CAMPO	DATOS	IMPR. CABECERA 1	IMPR. CABECERA 2	ESPAC. ANTES	NUOVO
1	Horas Extras	20	Horas	Fecha		
2	Pluses		Pluses	Nombre		
3	Nombre		Departamento	Base		
4	Departamento		I.R.P.F.			
5	Sueldo Base					
6	% I.R.P.F.					

¿ Quiere una prueba de impresión ? (S/N/C/M) ■

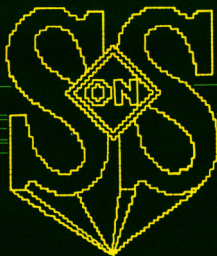
Sound-on-Sound

La cinta virgen para ordenador C15 y C20

¡NUEVA!



**Fabulosos
REGALOS**



Cintas de alta resolución



Sound-on-Sound es una marca registrada, producida y distribuida por
INDESCOMP, S.A. Avda. de Francisco, 35. Pol. Ind. de Gualda (Madrid).
Teléfono: 671 22 00-04.

Comprando una cinta Sound-on-Sound, usted puede obtener uno de estos regalos:

- Un ordenador PCW 8256 AMSTRAD.
- Un ordenador CPC 6128 AMSTRAD.
- Un ordenador CPC 6128 AMSTRAD.
- Una IMPRESORA para AMSTRAD.
- Un cassette electrónico y
- Un cassette software INDESCOMP.

PROGRAMACION FUNCIONAL Y LISP

Roberto Garrote Bernal

Hoy en día se considera que existen al menos dos grandes familias de lenguajes de programación: lenguajes imperativos y lenguajes funcionales o aplicativos.

E

n lenguaje imperativo es el que realiza sus objetivos mediante el cambio de valor de las variables a través de la asignación. Entre estos lenguajes se encuentran **BASIC** y **Pascal**. Los lenguajes imperativos están bien adaptados a los procesadores que se emplean actualmente, pero está empezando a aparecer un tipo diferente de computadoras, que se caracteriza por la gran cantidad de procesadores que utilizan. La potencia que nos ofrecen estas máquinas no se puede aprovechar bien con los lenguajes imperativos, razón por la cual se están adoptando los lenguajes funcionales.

Lenguajes y funciones

Un lenguaje aplicativo, como **LISP**, es el que consigue sus objetivos mediante la aplicación de funciones a argumentos, para lo cual no necesita modificar valores de variables. La primera pregunta que puede surgir es: ¿qué es una función? Una función es una regla que hace corresponder a cada tupla de argumentos un valor. Por ejemplo, la función hace corresponder a cada par de números su suma. Una función se puede definir dando una regla para calcular el valor que toma la función para cada argumento, como por ejemplo:

```
suma3(x):=x+3
```

El símbolo dos puntos-igual (:=) debe leerse como «se define como». Por tanto, la función suma3 se define como «el valor del argumento más 3». Para representar funciones de esta forma hay que suponer que existen algunas ya definidas y que podemos utilizarlas sin preocuparnos de cómo se calculan: sólo nos importa cuál es el valor que devuelven para unos argumentos concretos. Estas funciones se llaman «primitivas». En los ejemplos anteriores hemos supuesto que la función «+» era una función primitiva y por tanto estaba definida. Algunas funciones primitivas de LIPS son CAR,

CDR, CONS, COND y también +. Para definir las dos funciones de más arriba escribiríamos:

```
> (DE SUMA (X Y) (+ X Y))  
> (DE SUMA3 (X) (+ X 3))
```

(Os recuerdo que con el Especial **AMSTRAD** n.º 2 se regala un intérprete de LISP. Para entender la serie de artículos sobre Inteligencia Artificial que vamos a escribir en esta revista no es necesario tenerlo, pero te será de gran ayuda. La versión de LISP que hemos diseñado, MINILISP, sigue bastante al Standard LISP en lo que a funciones básicas se refiere, pero también cuenta con algunas características especiales de las que haremos uso en algunos momentos. En particular, los ejemplos de LISP que aparezcan en los artículos de esta sección estarán escritos en MINILISP).

Cómo se crea una función LISP

La palabra DE quiere decir que lo que sigue es la definición de una función. Después de DE se escribe el nombre de la función que queremos definir: **SUMA** y **SUMA3** en nuestros casos. Tras el nombre de la función se coloca la lista de sus parámetros. Los parámetros de una función son los valores de los que depende su resultado. Hay dos tipos de parámetros: parámetros formales, que son los que aparecen en la definición de la función, y parámetros actuales, que son los valores con los que se llama a la función.

Por ejemplo, X e Y son los parámetros formales de la función SUMA, mientras que en la llamada (SUMA 2 3) los parámetros actuales son 2 y 3. Los parámetros formales se usan en la definición de una función como si fueran variables. Cuando se llama a una función los parámetros formales se «ligan» a los parámetros actuales para calcular el valor de la función, es decir, a los parámetros formales se les asigna el valor de los parámetros actuales. En la llamada (SUMA 2 3), X estaría ligada al valor 2 e Y al valor 3.

El proceso por el cual se ligan parámetros

formales a parámetros actuales se llama «**paso de parámetros**». Este mecanismo de paso de parámetros es el que hace innecesaria la asignación en un lenguaje de programación funcional.

Pero bueno, ¿no estábamos hablando de cómo se definían funciones en LISP? En efecto. Decíamos que tras el nombre de la función había que colocar la lista de sus parámetros. Hay algunas funciones que no tienen parámetros, como por ejemplo:

```
> (DE TONTA () (NIL))
```

(Hay algunas funciones un poco más «listas» que tampoco tienen parámetros). Para las funciones sin parámetros se debe escribir (), que representa a la lista vacía, en lugar de la lista de sus argumentos. Una vez que se han dado los argumentos hay que proporcionar la regla con la cual se van a calcular los valores. En el caso de los dos primeros ejemplos, **SUMA** y **SUMA3**, estas reglas son (+ X Y) y (+ X 3), respectivamente. Más adelante insistiremos en cómo se definen funciones en LISP. Así es que ¡seguro que acabas aprendiendo!

Funciones que usan funciones

¿Qué tipo de cosas se pueden utilizar para definir funciones en LISP? Bueno, ya hemos visto que se pueden utilizar las funciones primitivas. Además, podemos emplear las funciones que hemos definido nosotros para construir nuevas funciones más complicadas.

```
> (DE SUMA (X) (SUMA2 (SUMA3X)))
```

El uso de unas funciones para calcular los valores de otras se llama composición funcional y es fundamental en los lenguajes aplicativos. En la definición de SUMA5 hemos empleado las funciones **SUMA** y **SUMA 3**. La función SUMA tiene dos argumentos: el primero es 2, pero el segundo es (SUMA3 X). ¿Cómo se entiende el geniecillo de MINILISP con este segundo argumento? La respuesta es que debe evaluarlo. ¿Evaluqué? Evaluarlo. El intérprete de LISP debe calcular el valor de (SUMA3 X) para saber qué tiene que SUMAR con 2. Apliquemos **SUMA6** a un argumento concreto para ver cómo calcula el intérprete de LISP su valor. Si estás leyendo este artículo delante de la computadora y tienes el genio de MINILISP dispuesto a servirte y ya le has dado las definiciones anteriores, entonces harías bien en escribir

```
> (TRACE ' (SUMA SUMA3 SUMA5))
```

con lo que conseguirás la «traza» de las funciones SUMA, SUMA3 y SUMA5 cuando se estén ejecutando. (La traza de una función es el «valor» de sus argumentos cuando éstos han sido evaluados y el valor de la función cuando ha sido calculado. Sigue leyendo que ahora vienen los ejemplos y lo entenderás mejor.)

Ahora le pedimos a nuestro genio que calcule

> (SUMA5 4)

La respuesta de MINILISP es la siguiente:

```
...
ARGUMENTO DE EVAL:
(SUMA6 4)
ARGUMENTO(S) DE SUMA5:
(4)
ARGUMENTO(S) DE SUMA3:
(4)
VALOR DE SUMA3:
7
ARGUMENTO(S) DE SUMA:
(2 7)
VALOR DE SUMA:
9
VALOR DE SUMA5:
9
TIEMPO 2.62 VALOR:
9
```

¿Qué significa todo esto?

Voy a explicar un poco todo esto. Los puntos suspensivos nos indican que el genio ha leído nuestra orden y se ha puesto a pensar cómo llevarla a cabo. Al poco tiempo de aparecer estos puntos el genio escribe «**ARGUMENTO DE EVAL:**» y lo que ha leído, (SUMA5 4) en nuestro caso, para que así podamos comprobar si esto es lo que de verdad queríamos de él. A partir de aquí empieza a actuar la función **TRACE**, que está siempre a la expectativa de encontrar una función que trazar. Para evaluar (SUMA5 4) hay que calcular primero el valor de su argumento. Como el valor de 4 es 4, la función **TRACE** nos lo indica con el mensaje

ARGUMENTO(S) DE SUMA5:
4

El intérprete de LISP «asigna» el valor del argumento recién calculado al parámetro formal correspondiente. Para pensar cómo lo hace el intérprete de LISP podemos imaginarnos que al hacer esta asignación se sustituye el parámetro formal, *x*, por su valor actual, 4. Entonces el valor de (SUMA5 4) es el mismo que el de (SUMA 2 (SUMA3 4)). Se intenta ahora trazar la función **SUMA**, pero el geniecillo se encuentra con que el segundo argumento aún no lo conoce: (SUMA3 4). El genio hace un alto en su camino y se dedica a calcular (SUMA3 4). De **SUMA3** ya conoce los valores de sus argumentos —en este caso sólo tiene uno—, pues el valor de 4 es 4, así es que puede comenzar a trazar esta función

ARGUMENTO(S) DE SUMA 3:
4

Para calcular el valor de (SUMA3 4) basta sumar 3 y 4, así que lo calcula y nos devuelve su valor

VALOR DE SUMA3:
7

En este momento el intérprete de LISP ya conoce los valores de todos los argumentos de la función **SUMA**, razón por la cual puede empezar a trazarla

ARGUMENTO(S) DE SUMA:
(2 7)

Como para calcular el valor de la función **SUMA** sólo se usa la función + y ésta no hay que trazarla (porque no se lo hemos dicho, que si no...), entonces la función **TRACE** nos entrega el valor de (SUMA 2 7)

VALOR SUMA:
9

Este valor pasa a ser el valor de la función **SUMA5** y así se nos indica

VALOR DE SUMA5:
9

Como ya ha calculado lo que le hemos pedido nos informa del tiempo que ha consumido y nos vuelve a indicar el valor. Estas dos últimas informaciones nos las da siempre, esté o no activada la función **TRACE**, pues en otro caso no habría forma de saber cuál es el resultado de los cálculos (esto último no es completamente cierto. En posteriores artículos ya veremos cómo se le puede ordenar al geniecillo de **MINILISP** que escriba). Ahora puedes comparar resultados desactivando la función **TRACE**. Para ello puedes escribir

> (UNTRACE (TRACE)))

con lo que no trazará ninguna función, o bien

> (UNTRACE ' (SUMA3 SUMA5))

con lo que sólo deja de trazar las funciones **SUMA3** y **SUMA5**. Al pedirle de nuevo que calcule (SUMA5 4) observarás que tarda menos tiempo. Esto es porque ahora no se entretiene en escribir la traza de las funciones.

Aparte de la composición de funciones existen otras dos formas de construir funciones en **LISP**: mediante expresiones condicionales y usando recursión. «Ahora mismo explico todo esto.» Las expresiones condicionales equivalen a la instrucción **IF... THEN... ELSE...** del **BASIC**. Las expresiones condicionales se indican mediante la función **COND**. Veamos un ejemplo

```
> (DE OPUESTO (X) (SUB O X))
> (DE ABSOLUTO (X))
> (T (OPUESTO X))
> ))
```

La función **OPUESTO** nos devuelve el opuesto de su argumento, o sea, el argumento cambiado de signo. Por ejemplo (OPUESTO 3) vale -3 y (OPUESTO -7) es 7. La función **ABSOLUTO** devuelve el valor absoluto de su argumento, por lo que (ABSOLUTO 3) vale 3, pero (ABSOLUTO -7) vale 7. En la definición de **ABSOLUTO** se emplea una expresión condicional. Vamos a ver cuál es su significado.

Después de la palabra **COND** se escriben

Inteligencia ARTIFICIAL

pares de expresiones: el primer elemento del par es lo que se pregunta y el segundo es el valor que toma la función en el caso de que la respuesta a la pregunta sea afirmativa. El primero de estos pares es ((> x 0) x) que equivaldría a la instrucción de **BASIC**

IF X > 0 THE ABSOLUTO = X ELSE...

En **LISP**, los valores booleanos (cierto, falso) se representan por **T** y **NIL** respectivamente. Sin embargo, se considera que la respuesta a una pregunta es cierta si no es **NIL** y es falsa si es **NIL**. Es decir, que cualquier cosa que no sea **NIL** se puede considerar como cierto. Por lo tanto, el segundo par, (T (OPUESTO X)), se podría escribir en **BASIC** como

IF 1 = 1 THE ABSOLUTO = OPUESTO (X) ELSE...

o bien, simplemente

ABSOLUTO = OPUESTO (X)

Renunciándolo todo, tendríamos que la función de **LISP**

```
(COND (( > x 0) X)
(T (OPUESTO X)))
```

equivale a la instrucción de **BASIC**

IF X > 0 THE ABSOLUTO = X ELSE ABSOLUTO = OPUESTO (X)

Veamos otro ejemplo que seguramente ya conocerás: se trata de la función factorial

```
> (DE FACTORIAL (N))
> (COND ((= N 0) 1)
> (T (* N (FACTORIAL (SUB N 1)))
> ))
```

Por si aún no habéis leído el artículo del Especial **AMSTRAD** n.º 2, diré primero qué es lo que hacen algunas de las funciones que he utilizado para definir **FACTORIAL**. Todas ellas están incorporadas en **MINILISP**. La función = compara dos números para ver si son iguales: si lo son devuelve **T** y si no lo son devuelve **NIL**. La función * por supuesto sirve para multiplicar dos números y **SUB** es la función para restar: devuelve el valor de su primer argumento menos el valor del segundo. ¡Pero que demonios, si también he usado la función **FACTORIAL**! ¿No era esta función la que queríamos definir? En efecto, utilizar lo que se quiere definir dentro de la propia definición es una de las maravillas del **LISP** y se llama recursión. Si tienes el intérprete de **MINILISP** puedes comprobar que **FACTORIAL** funciona bien, pero no intentes hacerla ejecutar con números negativos.

Pronto contaré cómo funciona esto de la recursión y por qué es peligroso preguntarle al geniecillo por (FACTORIAL -1).

DINAMIC Y MICRO 1 TE REGALAN UN AMSTRAD PCW512



¿COMO PUEDES GANAR ESTE PREMIO?

Muy fácil:

1. Corta el cupón que viene en esta página y rellénala con tus datos.
2. De aquí a tres meses saldrán hasta cinco preguntas, en las páginas de nuestra revista. Contéstalas y anota las respuestas en el cupón.
3. Una vez contestadas las cinco preguntas, envíanos el cupón, junto a una tarjeta promocional que podrás encontrar en todos los juegos de DINAMIC, a

HOBBY PRESS

Carretera de Irún, km. 12,400.

Fuencarral (Madrid).

Indicando claramente en el sobre: Para MEGACONCURSO AMSTRAD-DINAMIC.

4. Recuerda que aunque sólo envíes un cupón completo con las cinco respuestas, cada tarjeta promocional de los juegos Dinamic te da derecho a una participación en el concurso, por cada tarjeta una posibilidad.

SUERTE

1.
pregunta



Cuál es la primera versión de un programa DINAMIC para AMSTRAD



PROFANATION

ROCKY

WESTBANK

Cuando tengas las cinco respuestas, envía este cupón a MEGACONCURSO AMSTRAD-DINAMIC. HOBBY PRESS, Ctra. de Irún, km. 12,400. ALCOBENDAS (Madrid).

Nombre Apellidos
Domicilio
Ciudad Provincia
C. Postal Teléfono Profesión

Respuestas

1.
2.
3.
4.
5.

Los mejores en AMSTRAD

La mejor acción en el mejor programa.

Vive el juego! ¡Conviértete en máquina! Máximo entretenimiento.

El más adictivo y entretenido. Localiza todas las joyas y completa las 324 pantallas en 16 niveles.



Spectrum
Commodore
Amstrad



Spectrum
Commodore
Amstrad
Amstrad Disk



Spectrum
Commodore
MSX
Amstrad Disk

El éxito mundial! Con tu ayuda Bomb Jack puede conseguir el objetivo.



Spectrum
Commodore
Amstrad
Amstrad Disk

Tu eres el boxeador. Lucha y gana en este sin igual combate contra 8 campeones.



Spectrum
Commodore
Amstrad
Amstrad Disk

Evita la destrucción del Universo. ¡Sal al espacio exterior! ¿Llegarás a tiempo?



Spectrum
Commodore
Amstrad
Amstrad Disk

Procura que Jason no te alcance. Puede matarte.



Spectrum
Commodore
Amstrad
Amstrad Disk

El hombre de acero vuela sobre Metropolis.



Spectrum
Commodore
Amstrad



ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid.
Tel. 459 30 04. Tel. Barna. 209 33 65.
Telex: 22690 ZAFIR E

Programas editados, fabricados y distribuidos en España con la garantía Zafiro. Todos los derechos reservados.

DRAKE DEFENDER

Vive la angustiosa pero heroica resistencia de una nave intergaláctica durante el ataque sufrido por los crueles piratas espaciales, en el transcurso del viaje para aprovisionar de víveres a una de las lejanas colonias mineras de la confederación.



Este es un juego muy sencillo (hasta cierto punto). Sólo tiene dos pantallas. En la pantalla principal el defensor de una nave espacial puede situarse en cuatro lugares diferentes para proteger su nave. Cada 100 puntos se pasa de ésta a otra pantalla en la que hay que recorrer un laberinto en menos de veinte segundos. La dificultad, en ambas pantallas, es progresiva. El programa contiene instrucciones sobre el manejo.

Observaciones

La primera línea del programa sirve para utilizar los caracteres castellanos definidos en el programa. Al editar esta línea y borrarla hasta después del REM, y pulsando posteriormente ENTER aparecerán dichos caracteres.

ESTRUCTURAS

140	Instrucciones.
350	Inicialización juego.
400	Dibujo pantalla.
700	Juego.
770	Hombre arriba izquierda.
860	Hombre arriba derecha.
950	Hombre abajo izquierda.
1040	Hombre abajo derecha.
1130	Movimiento misiles.
1220	Misil nave 1.
1340	Misil nave 2.
1460	Misil nave 3.
1580	Misil nave 4.
1700	Muerte en pantalla principal.
1760	Muerte en meteoritos por tiempo.
1830	Muerte en meteoritos por choque.
1920	Paso pantalla.
2010	Control puntos.
2040	Pantalla meteoritos.
2300	Final juego.
2440	Marcador.
2500	Colores por fase.
2620	Caracteres.
2800	Caracteres castellanos.

Para utilizarlos basta con usar la tecla COPY para copiarlos en el lugar correspondiente.

El programa lleva incorporada la llamada al firmarse CALL &BBO3 (que limpia el buffer del teclado), y cuyo efecto secundario es que el programa no se pueda detener más que al principio.

Antes del programa principal hay un pequeño programa de presentación y carga.

TABLA DE VARIABLES

hi\$ (1)	Parte superior hombre izquierda.
hi\$ (2)	Parte inferior hombre izquierda.
hd\$ (1)	Parte superior hombre derecha.
hd\$ (2)	Parte inferior hombre derecha.
misil\$	Misil izquierda.
misil\$	Misil derecha.
nave\$	Nave propia.
met\$	Meteoritos.
bas\$	Base de entrada en la pantalla de meteoritos.
fase	Número de fase: no puede sobrepasar 10.
vidas	Número de vidas.
puntos	Puntuación.
rec	Récord.
pantalla	Número de pantalla.
mn (1)	Indicador de posición del misil de la nave 1.
mn (2)	Indicador de posición del misil de la nave 2.
mn (3)	Indicador de posición del misil de la nave 3.
mn (4)	Indicador de posición del misil de la nave 4.
fac1, fac2	Números aleatorios determinantes del lanzamiento de un misil.
lm	Número aleatorio que determina que nave lanza el misil.
cad	Variable que controla la frecuencia en el lanzamiento de misiles.
tm	Controlador de tiempo para lanzamiento de misiles.
tl	Tiempo para las interrupciones.
num	Número de orden del misil a mover.
Dim met (38,22)	Posibles posiciones de los meteoritos.
cm	Cantidad de meteoritos.

```

10 REM for
n=244 to 255:chr$(n):next
20 REM:=====
=====
=====
30 REM ASALTO
40 REM (c) Dylan Hoffman 1984
50 REM:=====
=====
=====
60 MODE 1:BORDER 0:IN$ 0,0:PAPER 0:
IN$ 1,2,24:LOCATE 12,12:PEN 1:PRINT
"PARA LA CINTA:FOR n=1 TO 100:NE
XT n:LOCATE 1,24:PRINT" Pulse u
na tecla para continuar "
70 IF INKEY="" THEN 70
80 B0SUB 2620
90 IN$ 1,24:CLS
100 LOCATE 6,13:PRINT"QUIERES INSTR
UCCIONES? (S/N)"
110 IF INKEY(60)<>"1" THEN 140
120 IF INKEY(46)<>"1" THEN 320
130 GOTO 110
140 REM:=====
=====
=====
150 REM INSTRUCCIONES
160 MODE 2
170 PRINT:PRINT TAB(13)"INSTRUCC
ION ES"
180 PRINT:PRINT
190 PRINT" Tu arriesgada mision e
s defender la nave que te ha sido
recomendada de los crueles pirata
s espaciales,que te atacan sin piedad
para apoderarse de tu cargamen
to.

"
200 PRINT" El generador que sual
nistra energia a los cañones laser
ha sido saboteado.Pero los tecnic
os han conseguido, en el ultimo mom
ento, hacerlo funcionar al mismo.
Por ello, lo unico que puedes hac
er es lanzar a los cohetes que lo
s pi-"
210 PRINT"ratas te lanzan justo cu
ndo se encuentren delante del cañon
"
220 PRINT" *misil$CHRS(224):PRINT
230 PRINT" Por otra parte, peric
damente tienes que atravesar una
zona de meteoritos para llegar a
una puerta espacial, que te permit
a saltar en el espacio-tiempo."PR
INT:
230 CALL &BBO3:PRINT:PRINT:PRINT"
PULSA UNA TE
CLA PARA CONTINUAR
"
240 PRINT" Los controles son:

$ Pantalla coh
etes:

(0): posicion superior izquierda
"
250 PRINT" (P): posicion supe
rior derecha. (A): pos
icion inferior izquierda.
(L): posicion inferior derecha.

"
260 PRINT" Para disparar, pulsa 1
a tecla de la posicion en que te ha
lies."PRINT
270 PRINT" Pantalla meteoritos: u
tilizar las teclas de cursor para d
esplazar la nave es-quivando los me
teoritos hasta una de las puertas (
"bas$;" ).
280 PRINT:PRINT:PRINT:PRINT"
(T) INSTRUCCIONES
/ (J) JUEGO
"
290 IF INKEY(35)<>"1" THEN 140

```

serie PRO



```

300 IF INKEY(45)<=1 THEN 320
310 GOTO 290
320 REM*****
330 REM JUEGO
340 MODE 1
350 "inicializacion
360 fase=0:vidas=5:puntos=0:pantalla
  1: fase=1: cad=7: t1=0.8: FOR n=1 TO
  4: an(n)=1: NEXT t1: ca=150
370 CLS
380 WINDOW #0,1,40,1,23:PAPER #0,0:
  WINDOW #1,1,40,24,25:PEN #1,1:PAPER
  #1,2:CLS#1:LOCATE #1,1:PRINT#1,"
  PUNTOS: "LOCATE #1,20,1:PRINT#1,"PA
  NTALLA: "LOCATE #1,1,2:PRINT#1,"VID
  AS: "LOCATE #1,20,2:PRINT#1,"FASE: "
390 ORIGIN 0,32
400 "dibuja pantalla
410 GOSUB 2440
420 INK 0,1:INK 1,26:INK 2,18:BORDE
  R 1:GOSUB 2500: colores
430 FOR n=1 TO 1000
440 xp=INT(RND#640):yp=INT(RND#360)
  :col=INT(RND#3)+1
450 PLOT xp,yp,col
460 NEXT
470 MOVE 256,224:DRAMR 32,32,2:DRAM
  R 64,0:DRAMR 32,-32
480 MOVE 256,224:DRAMR 32,32,2:DRAM
  R 60,0:DRAMR 32,-32
490 MOVE 224,160:DRAMR 32,46:DRAMR
  32,0:DRAMR 16,18:DRAMR 32,0:DRAMR 1
  6,-18:DRAMR 32,0:DRAMR 32,-46
500 MOVE 224,160:DRAMR 32,46:DRAMR
  32,0:DRAMR 16,18:DRAMR 32,0:DRAMR 1
  6,-18:DRAMR 32,0:DRAMR 32,-46
510 x=238:y=145:long=164
520 FOR n=1 TO 16
530 x=x+2:y=y+1:long=long-4
540 MOVE x,y:DRAMR long,0
550 NEXT
560 PEN 2
570 LOCATE 14,10:PRINT CHR$(223)+CHR
  $(223)+CHR$(225)+CHR$(215)
580 LOCATE 24,10:PRINT CHR$(214)+CHR
  $(226)+CHR$(223)+CHR$(223)
590 LOCATE 12,14:PRINT CHR$(223)+CHR
  $(223)+CHR$(225)+CHR$(215)
600 LOCATE 26,14:PRINT CHR$(214)+CHR
  $(226)+CHR$(223)+CHR$(223)
610 LOCATE 19,14:PRINT CHR$(214)+CHR
  $(143)+CHR$(143)+CHR$(215)

```

```

620 FOR n=304 TO 336 STEP 4
630 MOVE n,160:DRAMR 0,62,1
640 NEXT
650 PEN 3
660 FOR n=1 TO 3
670 LOCATE 5,1+n:PRINT naval$(n):LOC
  ate 33,1+n:PRINT naval$(n):LOCATE
  3,19+n:PRINT naval$(n):LOCATE 35,19
  +n:PRINT naval$(n)
680 NEXT
690 PEN 1:FOR n=1 TO 2:LOCATE 18,8+
  n:PRINT hi$(n):NEXT
700 "Juego
710 t=TIME
720 IF INKEY(67)<=1 THEN 770
730 IF INKEY(27)<=1 THEN 860
740 IF INKEY(69)<=1 THEN 950
750 IF INKEY(36)<=1 THEN 1040
760 IF TIME>t+3000 THEN 1130 ELSE
  E 720
770 "hombre arriba izquierda
780 CALL &BD19
790 SOUND 1,70,5,7
800 FOR n=1 TO 2:LOCATE 23,8+n:PRIN
  T " "LOCATE 16,12+n:PRINT " "LOCATE
  25,12+n:PRINT " "NEXT
810 PEN 1:FOR n=1 TO 2:LOCATE 18,8+
  n:PRINT hi$(n):NEXT
820 LOCATE 13,10:PRINT CHR$(224)
830 IF an(1)=7 THEN LOCATE 12,10:PR
  INT " "an(1)=1:puntos=puntos+2:LOC
  ate #1,8,1:PRINT#1,puntos:GOSUB 201
  0
840 FOR n=1 TO 50:NEXT:LOCATE 13,10
  :PRINT " "
850 GOTO 760
860 "hombre arriba derecha
870 CALL &BD19
880 SOUND 1,70,5,7
890 FOR n=1 TO 2:LOCATE 18,8+n:PRIN
  T " "LOCATE 16,12+n:PRINT " "LOCATE
  25,12+n:PRINT " "NEXT
900 PEN 1:FOR n=1 TO 2:LOCATE 23,8+
  n:PRINT hi$(n):NEXT
910 LOCATE 28,10:PRINT CHR$(224)
920 IF an(2)=7 THEN LOCATE 29,10:PR
  INT " "an(2)=1:puntos=puntos+2:LOC
  ate #1,8,1:PRINT#1,puntos:GOSUB 201
  0
930 FOR n=1 TO 50:NEXT:LOCATE 28,10
  :PRINT " "
940 GOTO 760
950 "hombre abajo izquierda
960 CALL &BD19

```

```

970 SOUND 1,70,5,7
980 FOR n=1 TO 2:LOCATE 18,8+n:PRIN
  T " "LOCATE 23,8+n:PRINT " "LOCATE
  25,12+n:PRINT " "NEXT
990 PEN 1:FOR n=1 TO 2:LOCATE 16,12
  +n:PRINT hi$(n):NEXT
1000 LOCATE 11,14:PRINT CHR$(224)
1010 IF an(3)=7 THEN LOCATE 10,14:PR
  INT " "an(3)=1:puntos=puntos+2:LOC
  ate #1,8,1:PRINT#1,puntos:GOSUB 20
  10
1020 FOR n=1 TO 50:NEXT:LOCATE 11,1
  4:PRINT " "
1030 GOTO 760
1040 "hombre abajo derecha
1050 CALL &BD19
1060 SOUND 1,70,5,7
1070 FOR n=1 TO 2:LOCATE 18,8+n:PRIN
  T " "LOCATE 23,8+n:PRINT " "LOCATE
  16,12+n:PRINT " "NEXT
1080 PEN 1:FOR n=1 TO 2:LOCATE 25,1
  2+n:PRINT hi$(n):NEXT
1090 LOCATE 30,14:PRINT CHR$(224)
1100 IF an(4)=7 THEN LOCATE 31,14:PR
  INT " "an(4)=1:puntos=puntos+2:LOC
  ate #1,8,1:PRINT#1,puntos:GOSUB 20
  10
1110 FOR n=1 TO 50:NEXT:LOCATE 30,1
  4:PRINT " "
1120 GOTO 760
1130 "movimiento misiles
1140 fac1=INT(RND#cad)+1:fac2=INT(R
  NDCad)+1
1150 la=INT(RND#4)+1
1160 IF fac1=fac2 AND an(1)=1 THE
  N an(1)=1
1170 FOR num=1 TO 4
1180 ON num GOTO 1220,1340,1460,158
  0
1190 an(num)=an(num)+1
1200 NEXT
1210 GOTO 710
1220 "misil nave 1
1230 CALL &BD19
1240 IF an(1)=1 THEN 1200
1250 SOUND 1,300,5,7
1260 ON an(1) GOTO 1270,1280,1290,1
  300,1310,1320,1330
1270 LOCATE 7,5:PRINT " "LOCATE 11
  90
1280 LOCATE 7,5:PRINT " "LOCATE 8,6
  :PRINT misil#GOTO 1190
1290 LOCATE 8,6:PRINT " "LOCATE 9,7
  :PRINT misil#GOTO 1190
1300 LOCATE 9,7:PRINT " "LOCATE 10,
  8:PRINT misil#GOTO 1190
1310 LOCATE 10,8:PRINT " "LOCATE 11
  9:PRINT misil#GOTO 1190
1320 LOCATE 11,9:PRINT " "LOCATE 12
  10:PRINT misil#GOTO 1190
1330 LOCATE 12,10:PRINT " "LOCATE 1
  3:PRINT misil#an(1)=1:GOTO 1700
1340 "misil nave 2
1350 CALL &BD19
1360 IF an(2)=1 THEN 1200
1370 SOUND 1,310,5,7
1380 ON an(2) GOTO 1390,1400,1410,1
  420,1430,1440,1450
1390 LOCATE 34,5:PRINT misd#GOTO 1
  190
1400 LOCATE 34,5:PRINT " "LOCATE 33
  ,6:PRINT misd#GOTO 1190
1410 LOCATE 33,6:PRINT " "LOCATE 32
  ,7:PRINT misd#GOTO 1190
1420 LOCATE 32,7:PRINT " "LOCATE 31
  ,8:PRINT misd#GOTO 1190
1430 LOCATE 31,8:PRINT " "LOCATE 30
  ,9:PRINT misd#GOTO 1190
1440 LOCATE 30,9:PRINT " "LOCATE 29

```



```

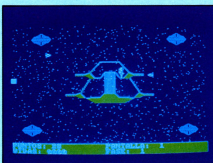
1,10:PRINT mid$;GOTO 1190
1450 LOCATE 29,10:PRINT "":LOCATE 2
8,10:PRINT mid$:an(2)=-1:GOTO 1700
1460 "mil nave 3
1470 CALL $B019
1480 IF an(3)=1 THEN 1200
1490 SOUND 1,320,5,7
1500 ON an(3) GOTO 1510,1520,1530,1
540,1550,1560,1570
1510 LOCATE 5,19:PRINT mid$:GOTO 1
190
1520 LOCATE 5,19:PRINT "":LOCATE 6,
18:PRINT mid$:GOTO 1190
1530 LOCATE 6,18:PRINT "":LOCATE 7,
17:PRINT mid$:GOTO 1190
1540 LOCATE 7,17:PRINT "":LOCATE 8,
16:PRINT mid$:GOTO 1190
1550 LOCATE 8,16:PRINT "":LOCATE 9,
15:PRINT mid$:GOTO 1190
1560 LOCATE 9,15:PRINT "":LOCATE 10
14:PRINT mid$:GOTO 1190
1570 LOCATE 10,14:PRINT "":LOCATE 11
13:PRINT mid$:an(3)=1:GOTO 1700
1580 "mil nave 4
1590 CALL $B019
1600 IF an(4)=1 THEN 1200
1610 SOUND 1,320,5,7
1620 ON an(4) GOTO 1630,1640,1650,1
660,1670,1680,1690
1630 LOCATE 36,19:PRINT mid$:GOTO
1190
1640 LOCATE 36,19:PRINT "":LOCATE 3
5,18:PRINT mid$:GOTO 1190
1650 LOCATE 35,18:PRINT "":LOCATE 3
4,17:PRINT mid$:GOTO 1190
1660 LOCATE 34,17:PRINT "":LOCATE 3
3,16:PRINT mid$:GOTO 1190
1670 LOCATE 33,16:PRINT "":LOCATE 3
2,15:PRINT mid$:GOTO 1190
1680 LOCATE 32,15:PRINT "":LOCATE 3
1,14:PRINT mid$:GOTO 1190
1690 LOCATE 31,14:PRINT "":LOCATE 3
0,14:PRINT mid$:an(4)=1:GOTO 1700
1700 "muerte
1710 FOR n=30 TO 2 STEP -2:SOUND 1,
500,5,7,0,0:NEXT
1720 LOCATE 13,10:PRINT SPACE(1):L
OCATE 29,10:PRINT SPACE(1):LOCATE
11,14:PRINT SPACE(1):LOCATE 30,14:
PRINT SPACE(1)
1730 vidas=vidas-1:GOSUB 2440
1740 IF vidas=0 THEN 2300
1750 GOTO 710
1760 "muerte en meteoritos por tiam
po
1770 FOR n=30 TO 2 STEP -2:SOUND 1,
500,5,7,0,0:NEXT
TIEMPO CE
"
1790 FOR n=1 TO 300:NEXT
1800 vidas=vidas-1:GOSUB 2440
1810 IF vidas=0 THEN 2300
1820 GOTO 1920
1830 "muerte en meteoritos por choq
ue
1840 FOR n=30 TO 2 STEP -2:SOUND 1,
500,5,7,0,0:NEXT
1850 FOR n=1 TO 81:a(n)=INT(RND*255)
+1:NEXT
1860 SYMBOL 245,a(1),a(2),a(3),a(4)
,a(5),a(6),a(7),a(8)
1870 LOCATE x11,y11:PRINT CHR$(24
5)
1880 FOR n=1 TO 300:NEXT
1890 vidas=vidas-1:GOSUB 2440
1900 IF vidas=0 THEN 2300
1910 GOTO 1920
1920 "paso pantalla meteoritos
1930 ERASE met
1940 pantalla=pantalla+1
1950 IF (pantalla MOD 5)=0 THEN vid
as=vidas+1
1960 IF fase<10 THEN fase=fase+1
1970 IF cad=0 THEN cad=cad+1
1980 FOR n=1 TO 4:m(n)=1:NEXT
1990 ti=0,B
2000 CLS:GOTO 400
2010 "control puntos
2020 IF (puntos MOD 10)=0 AND ti>0.
7 THEN ti=ti-0,1
2030 IF (puntos MOD 100)>0 THEN RE

```

```

TURN
2400 "pantalla meteoritos
2050 DIN met(39,23)
2060 CLS:INK 1,26:INK 2,6:INK 3,18
2070 FOR n=1 TO ca
2080 xm=INT(RND*38)+1:ym=INT(RND*23
14:icola=INT(RND*2)+1
2090 PEN col:=1:LOCATE xm,ym:PRINT
met$

```



```

2100 met(xm,ym)=1
2110 NEXT
2120 IF ca<350 THEN ca=ca+50
2130 PEN 1:FOR n=1 TO 3:LOCATE 40,1
0:n:PRINT bas$:NEXT
2140 LOCATE 1,12:PRINT navp$(2)
2150 xn=1:ym=12
2160 tiempo=TIME
2170 IF INKEY(60)>-1 THEN 1830
2180 IF INKEY(60)<-1 THEN xn1=yn1
n1:yn1=yn1-xn-1:v=1:GOSUB 2230
2190 IF INKEY(1)<-1 THEN xn1=yn1
n1:yn1=yn1-xn+1:v=2:GOSUB 2230
2200 IF INKEY(8)<-1 AND xn=1 THEN
vn1=yn1:yn1=yn-xn-1:v=3:GOSUB 2
230
2210 IF INKEY(2)<-1 THEN xn1=yn1
n1:yn1=yn1+1:v=4:GOSUB 2230
2220 IF TIME>tiempo+6000 THEN 1760
ELSE 2280
2230 IF xn>39 THEN xn=1
2240 IF yn<1 THEN yn=23
2250 IF yn>23 THEN yn=1
2260 IF met(xn,yn)=1 THEN 1830
2270 CALL $B019:LOCATE xn1,yn1:PR
INT SPACE(1):LOCATE xn,yn:PRINT na
vp$(v)
2280 IF xn=39 AND (yn=11 OR yn=12 O
R yn=13) THEN puntos=puntos+20:GOTO
1920
2290 RETURN
2300 "final juego
2310 SOUND 1,100,100,7
2320 PEN 1:LOCATE 9,12:PRINT " JUE
GO TERMINADO "
2330 CALL $B035:WHILE INKEY="" :MEN
D
2340 MODE 1
2350 FOR puntos<rec THEN 2400
2380 rec=puntos
2390 LOCATE 12,10:PRINT"NUEVO RECOR
D 1111"
2400 LOCATE 3,24:PRINT"QUIERES JUA
R OTRA PARTIDA? (S/N)"
2410 IF INKEY(60)<-1 THEN 320
2420 IF INKEY(46)<-1 THEN CLS:END
2430 GOTO 2410
2440 "marcador
2450 LOCATE #1,8,1:PRINT#1,puntos
2460 LOCATE #1,8,2:PRINT#1,STRING$(
vidas,217):SPACE$(1)
2470 LOCATE #1,26,2:PRINT#1,fase
2480 LOCATE #1,30,1:PRINT#1,pantall
a
2490 RETURN
2500 "colores
2510 ON fase GOTO 2520,2530,2540,25
50,2560,2570,2580,2590,2600,2610
2520 INK 3,13:RETURN
2530 INK 3,24:RETURN

```

```

2540 INK 3,20:RETURN
2550 INK 3,22:RETURN
2560 INK 3,25:RETURN
2570 INK 3,16:RETURN
2580 INK 3,15:RETURN
2590 INK 3,3:RETURN
2600 INK 3,8:RETURN
2610 INK 3,6:RETURN
2620 "caracteres especiales
2630 SYMBOL $FTER 217 -
2640 RESTORE 2690
2650 FOR n=217 TO 244
2660 READ a,b,c,d,e,f,g,h
2670 SYMBOL n,a,b,c,d,e,f,g,h
2680 NEXT
2690 DATA 56,92,95,93,61,25,47,47,5
5,230,36,60,24,24,24,24,28,58,250,1
86,188,152,244,244,236,103,36,60,24
,24,24,24,128,64,188,215,235,188,64
,128,1,2,61,235,215,61,2,1,0,0,255,
153,255,0,0,0,0,0,255,0,0,0,0,7,6
3,255,255,255,63,15,3
2700 DATA 224,252,255,255,255,252,2
40,192,0,0,0,0,0,0,15,0,0,3,14,
54,238,214,0,0,0,192,112,108,119,10
7,0,0,0,0,0,0,240,31,57,102,255,2
35,102,57,31,246,153,102,255,255,10
2,153,246,111,153,102,255,255,102,1
53,111,248,156,102,255,255,102,156,
248
2710 DATA 15,0,0,0,0,0,0,214,238,
54,14,3,0,0,0,107,119,108,112,192,0
,0,240,0,0,0,0,0,0,24,24,60,102
,102,102,255,255,192,248,252,199,19
9,252,248,192,3,31,63,227,227,63,31
,3,255,255,102,102,60,24,24,60,
74,145,171,213,137,82,60
2720 DATA 58,38,235,151,73,106,54,8
2730 ih$(1)=CHR$(217):ih$(2)=CHR$(2
18):"hombre izquierda
2740 hd$(1)=CHR$(219):hd$(2)=CHR$(2
20):"hombre derecha
2750 mid$(CHR$(221):mid$(CHR$(222
1):"misiles
2760 naval$(1)=CHR$(227):CHR$(228)+
CHR$(229)+CHR$(230):naval$(2)=CHR$(
231)+CHR$(232)+CHR$(233)+CHR$(234):
naval$(3)=CHR$(235)+CHR$(236)+CHR$(
237)+CHR$(238):"nave alienigena
2770 navp$(1)=CHR$(239):navp$(2)=CH
R$(240):navp$(3)=CHR$(241):navp$(4)
=CHR$(242):"nave propia
2780 met$(CHR$(244):"meteorito
2790 bas$=CHR$(243)
2800 "caracteres castellanos
2810 SYMBOL 246,8,16,120,12,124,204
,122
2820 SYMBOL 247,8,16,60,102,126,96,
60
2830 SYMBOL 248,8,16,56,24,24,24,60
2840 SYMBOL 249,8,16,60,102,102,102
,60
2850 SYMBOL 250,8,16,102,102,102,10
2,62
2860 SYMBOL 251,102,0,102,102,102,1
02,62
2870 SYMBOL 252,126,0,220,102,102,1
02,102
2880 SYMBOL 253,254,0,198,230,246,2
22,200,198
2890 SYMBOL 254,60,102,102,48,24,0,
24
2900 SYMBOL 255,0,24,0,24,24,24,24
,24
2910 RETURN
2920 STOP
2930 SAVE "ASALTO"

```



ORDEMANIA SOFT

por fin..

GESPAC

Paquete integrado de gestión que le permite a Vd., de forma fácil y optimizando el tiempo, la gestión global de su empresa.

Este paquete incorpora los programas de CONTABILIDAD, FACTURACION y CONTROL DE STOCKS además de un programa de CONTROL DE PEDIDOS y todo de forma interactiva. Controle su empresa por sólo **29.900 pts.**

Contabilidad

Contabilidad de fácil manejo y de gran potencia que permite trabajar con cuentas de hasta cuatro niveles, con capacidad según diskette de 500/1.000 cuentas y de 2.000/10.000 asientos.

Permite modificar o dar de baja apuntes ya integrados en el Mayor, programaciones de cierres, ficheros de Contabilidad y Cuenta de Explotación, ejecución de balances comparativos, reenlázandolos por meses, clave acceso restringido, etc...

19.900 ptas.

Facturación

Programa de gran agilidad y rapidez que incorpora el Control de Clientes, con gran capacidad de datos, Artículos, Albaranes, Facturas y recibos.

Generación automática y manual de documentos, valoración, a voluntad de los albaranes, todo tipo de listado, incluido el del IVA de las facturas emitidas para la declaración de Hacienda, etc...

15.500 ptas.

Control de stocks

Gran capacidad de datos, le permitirán a Vd. llevar con claridad y sencillez el control de su stock.

El programa le permite llevar un libro de entradas/salidas, reorganizarlo, hacer listados de stocks..., le avisará de los límites de stocks, mínimo y máximo por artículo, etc...

Todo para la llevanza de su almacén.

14.900 ptas.

disponibles para: PCW 8256
Asimismo, PCW 8512

Asimismo, Contabilidad disponible para CPC 664/6128 **9.900 ptas.**

OFITES INFORMATICA

DISTRIBUIDOR EXCLUSIVO DE ORDEMANIA SOFT

IVA INCLUIDO

Condiciones especiales para distribuidores
Si tiene alguna dificultad en obtenerlos dirijase a

Ofites
Informatica

Avda. Isabel II, 16, 8º
Tel. 453544 453533
28011 SAN SEBASTIAN

NUMEROS

Vamos a comenzar por lo más básico: las cuatro reglas aritméticas. Por supuesto que sabemos que todos dominan la aritmética sencilla, todos... menos el Amstrad, y eso puede traernos problemas.



Igual que los humanos, los ordenadores precisan expresiones unívocas, ante:

PRINT 8+5

la respuesta es clara, imprimirá un hermoso 8 en la pantalla. Pero si se encuentra con:

PRINT 5+3*8

hasta nosotros dudamos si la respuesta es 64 o 29. Pero el ordenador no puede hacerlo porque se bloquea. No admite ambigüedades.

Para evitarlas existe una serie de reglas incorporadas al Basic que nos permiten manipular los números sin ningún problema. Es lo que se llama Regla de la Precedencia de Operadores y quiere decir que por un acuerdo asignamos un orden de ejecución a los diferentes símbolos que operan sobre los números.

El orden de prioridad establecido es:

- Potenciación.
- Multiplicación y división.
- Suma y resta.

En cuanto a las operaciones con la misma prioridad se evalúan de izquierda a derecha. Por eso nuestro ordenador ha contestado con un 29 al problema anterior.

Sin embargo, todo lo que hemos dicho anteriormente no es rígido. Todos los lenguajes tienen prevista la posibilidad de que nos interese saltarnos las reglas. Y el modo de hacerlo es mediante nuestro antiguo conocido del colegio: el paréntesis. Vamos a conseguir un 64 como re-

sultado de la operación anterior. Basta con hacer:

PRINT (5+3)*8

Vamos a darle una regla de oro: ante la duda utilice los paréntesis siempre que los necesite. Es muy difícil depurar errores cometidos al jugar con los números.

Otra regla más, deje que trabaje el ordenador. Usted sólo siéntese y... disfrute.

Exponenciación

A nivel de operaciones básicas nos queda por hablar de la exponenciación. Consiste en tomar un número llamado BASE y multiplicarlo por sí mismo tantas veces como lo indica un segundo llamado EXPONENTE.

Su representación gráfica es AB, donde A es la base.

El estudio de la exponenciación es muy interesante porque nos va a permitir aprender mucho más acerca del tratamiento de números, y de la capacidad de cálculo de los ordenadores.

Por muy potente que sea un ordenador el rango de números que maneja es limitado, entre otras cosas porque ocupan memoria al representarse internamente. Cada máquina tiene asignado un número fijo de bytes para contener una cantidad. De ahí el techo del cálculo.

Pero, ¿cómo averiguar el rango? Fácilmente, cualquier número dividido por cero es infinito, ¿verdad? Para nosotros lo es, pero para el ordenador «infinito» es el número máximo que puede representar, así que:

PRINT 3/0

por ejemplo, nos dará el «techo» de cálculo del ordenador. Este número viene siendo un 1 seguido de 38 ceros: una cifra inimaginable.

Por lo que respecta al límite inferior, el sistema para encontrarle es igual de fácil:

PRINT (3/0)-1

nos dará como resultado menos «in-

finito» y que corresponde a un 10 elevado a -39.

Y en el caso que, en una operación, nos salgamos de rango, ¿qué pasa? Pues que el Amstrad nos obsequia con un mensaje de error: «OVERFLOW» que quiere decir «sobrepasamiento» y el programa se detiene.

Aunque no lo parezca, este mensaje de error es positivo. Podemos aprovecharlo para calcular el máximo exponente al que pueda elevarse un número, como hace el Programa 1.

Está formado por cuatro bucles:

1. El principal (líneas 70-170).
2. La subrutina de entrada de datos (200-220).
3. Saca el error, caso que se produzca, indicando el valor máximo (240-260).
4. Subrutina de captura de errores (280-310).

Este último es el punto más interesante, puesto que si se produce el error nos lo identifica. Cada error posee un código numérico asociado que puede encontrarse en el capítulo 7 del Manual.

Si el error es diferente a «OVERFLOW», el programa no lo trata, o lo ignora, mediante la orden RESUME NEXT de la línea 280 que fuerza al Basic a reanudar la ejecución del programa en la línea siguiente a donde se produjo el error.

Sin embargo, las realmente importantes son la 300 y 310. La primera decremente el exponente en una unidad, mientras que la orden: RESUME de la segunda obliga al Amstrad a continuar la ejecución en la línea 120, la misma donde se produjo el error. Y así lo hará una y otra vez hasta que no detecte ninguno.

Es un programa interesante pero no perfecto, porque si bien nos impide sobrepasar el techo de cálculo, no ocurre lo mismo «por debajo» y cuando juguemos con el observaremos que los números negativos acabarán de dar cero. ¿Por qué?

Primeros repasos

```

10 REM PROGRAMA I
20 ON ERROR GOTO 280
30 :
40 CLS
50 overflow.error=6:ok=0:flag.error
=0
60 :
70 WHILE NOT ok
80 :
90 GOSUB 200
100 :
110 PRINT "Calculando"
120 res=base^ex
130 :
140 GOSUB 240
150 :
160 PRINT base;"elevado a";ex;"vale
":res
170 WEND

180 END
190 :
200 INPUT "Base: ";base
210 INPUT "Exponente: ";ex
220 RETURN
230 :
240 IF flag.error THEN PRINT "Ha ha
bido un overflow":PRINT "El maximo
exponente posible es:";ex
250 flag.error=0
260 RETURN
270 :
280 IF ERR<>overflow.error THEN RES
UME NEXT
290 flag.error=-1
300 ex=ex-1
310 RESUME
320 :

```

PUBLICIDAD



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

MASTER COMPUTER

Centro Comercial Guadalupe
Ctra. Canillas, 136-1.ª planta
Tel. 200 80 65 **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**)

MASTER HARD

Servicio Técnico
Para **AMSTRAD**
en Galicia,
León
y Asturias.

C/ Magdalena, 213
El Ferrol
Tels.: (981) 35 84 32



¡Operación cambio!

Valoramos:

Tu AMSTRAD 464 en 50.000 ptas.
Un Spectrum+ en 30.000 ptas.
Amstrad CPC 664 en 70.000 ptas.
En la compra del AMSTRAD CPC 6128 o PCW 8256.
Consulte para monitor color.
Precios especiales en impresoras y accesorios.

☎ Tardes 270 34 97.

OPERACION CAMBIO

Pásate a monitor color por
25.000 ptas.

Valoramos:

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

MICRO DEALER AMSTRAD CENTER

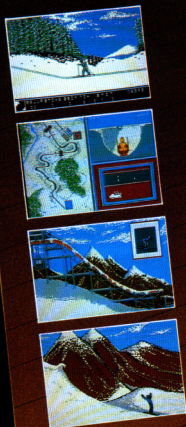
MAYORISTAS DE INFORMATICA

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 pulgadas, cable, etc.
Sinclair, Commodore, New Print,
Spectravideo, Compatibles IBM
C/ Comandante Zorita, 13. 28020 Madrid
Tels. 233 07 81 - 233 07 35

Primero fué SUMMER GAMES
despues SUMMER GAMES II

y ahora...

WINTER GAMES



Ha conseguido el oro en los Juegos de Verano y Juegos de Verano II. ¡Ahora estamos en los Juegos de Invierno!, y que increíble marco, un completo país de invierno realizando seis competiciones de acción. Puede competir contra sus amigos o el ordenador. Primero elija el país que quiera representar. Practíquelo, prepárese y aprenda una estrategia para ganar en cada competición. Ahora comience la ceremonia de entrega de premios. ¿Será usted quien consiga el oro en la ceremonia de entrega de premios?

La búsqueda del oro continúa... y está todo aquí: la estrategia, el reto, la competición, el arte y la pompa de los Juegos de Invierno.

- Seis competiciones de invierno: Bobsled, salto de ski, patinaje artístico, patinaje libre estilo, Hot Dog Aéreo y el ski de fondo.
- Ceremonias de apertura, cierre y entrega de premios con himnos nacionales.
- Compita contra el ordenador o contra sus amigos o familia.
- Control único por el joystick, necesita destreza y cronometraje.
- Uno a ocho jugadores.

EPYX
COMPUTER SOFTWARE

Fabricado y distribuido en exclusiva por:

COMPULOGICAL S.A.
Santa Cruz de Marceda, 31 - 28015 Madrid - Tel. 241 1063

Distribuido en Cataluña y Baleares por: **Disponible para Commodore 64, Spectrum y Amstrad**
DISCLU, S.A. - Balmes, 58 - BARCELONA - Tel. (93) 302 39 08 - P.V.P. 2.300 Ptas.

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

OLVIDA TODO LO QUE HAS VISTO

No,
en
U.S.A.



Commodore 64/128

KUNG-FU MASTER
¡¡EL DEFINITIVO!!

LA VERSION OFICIAL DE LAS MAQUINAS

ERBE *dades*

¿BUSCAS EMOCIONES FUERTES?



Green Beret (bomba verde), es el programa de mayor acción que hayas visto jamás en un ordenador.
Decir Green Beret es decir un hombre perfectamente entrenado para la lucha.
¡Prepárate...



"V"

La serie de ciencia-ficción televisiva invade el ordenador. La Tierra es visitada por alienígenas de aspecto humano. Sin embargo, esos tentáculos intentan revelar su objetivo final... conquistar la Tierra y esclavizar a la raza humana.



La crítica ha dicho: "Realizar un juego más completo debe ser poco menos que imposible".
En el papel de detective Jack Marlow, prepárate a enfrentarte al rey de la Mafia en el más apasionante juego que hayas visto.