

AÑO II N.º 49

MICROHOBBY

AMSTRAD

Semanal

AÑO II N.º 49

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

160 Ptas.

Canarias 165 pts.

TASWORD:
PROCESO DE TEXTOS
EN 128 K

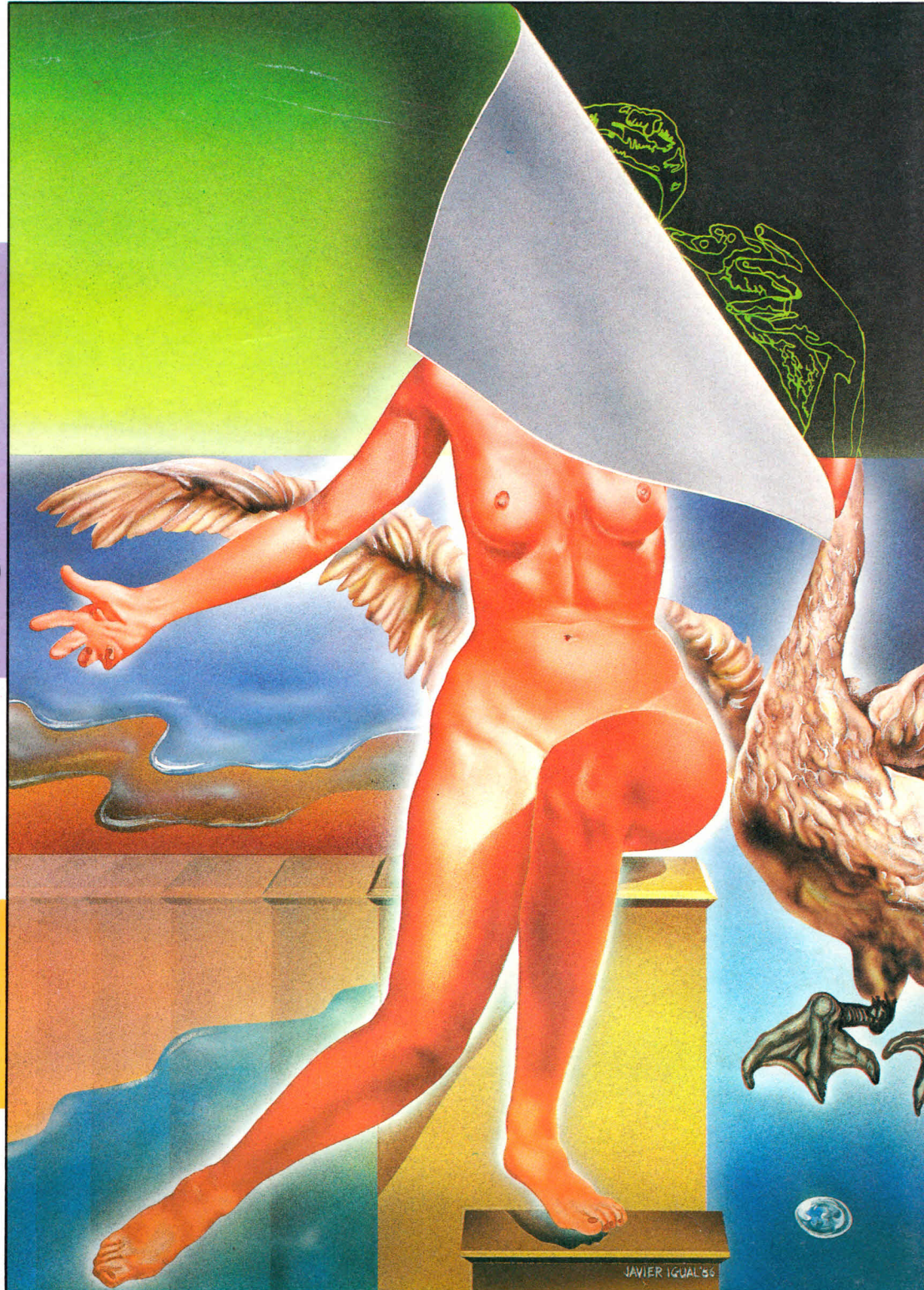


**SCROLL PIXEL
A PIXEL DE
CUALQUIER
ZONA DE LA
PANTALLA.
¡EN LAS CUATRO
DIRECCIONES!**

**PUZZLES POR
ORDENADOR.
¿A QUE NO LE GANAS
A TU AMSTRAD?**

**ORDENADORES
PARALELOS:
"THE CONECCION
MACHINE"
ESTA AQUI**

**El arte de tomar
decisiones con
elegancia en Basic**



CADA MES UN HOBBY EN TU KIOSCO
¡Ya está a la venta el número 3!



Un nuevo hobby por descubrir
te está esperando en el kiosco.
No dejes pasar la ocasión
de sumergirte entre las páginas
de la única revista que sabe satisfacer tu idea
de lo que es un hobby.

MICROHOBBY

AMSTRAD

Sumario

Año II • Número 49 • 5 al 18 de Agosto de 1986
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas. + 10 ptas. sobretasa aérea
Ceuta y Melilla, 155 ptas.

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.ª Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

Fernando Chaumel

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sopena

Robert Chatwin

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Alberto Suñer

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Chema Sacristán

Portada

Javier Igual

Ilustradores

J. Igual, J. Pons, F. L. Frontán,

J. Septien, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

María Andrino

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

Marta García

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortés

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

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

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

ROTEDEC, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, 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 solidaria de las

opiniones vertidas por sus

colaboradores en los artículos

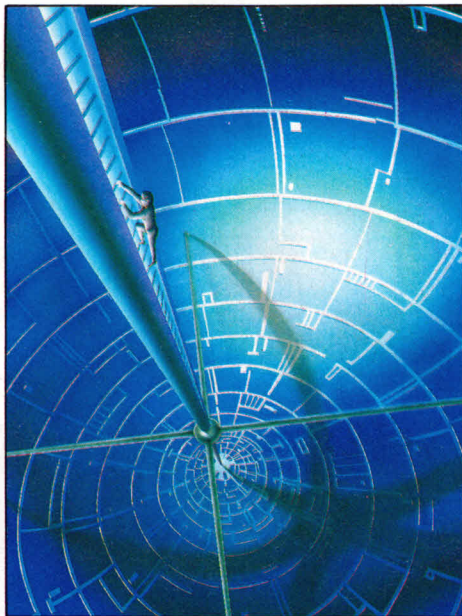
firmados. Reservados todos los

derechos.

6

Primeros pasos

Dentro de nuestro estudio de las interioridades del Amstrad Basic, estamos ya listos para introducirnos en la comprensión y dominio de una de las instrucciones de bifurcación y decisión más sofisticadas y elegantes que existen, y que sólo los dialectos Basic que podríamos llamar de «élite» poseen: la orden **«ON GOSUB»**.



16

ProgramAcción

De nuevo los comandos RSX vienen en nuestro auxilio para implementar algo que todos los programadores, sin excepción, y especialmente los de juegos, agradecerán en gran medida. Cómo hacer **«scroll»** de cualquier porción de la pantalla en las cuatro direcciones.



20 Banco de pruebas

Saludamos desde aquí a un viejo conocido: el programa procesador de textos **«Tasword»**. **AMSTRAD SEMANAL** lo examinó en su día, pero ha aparecido una nueva versión, casi integra en código máquina y capaz de manejar el segundo banco de memoria del 6128; deja 64 Kbytes libres para texto, y creemos que merece la pena probarlo una vez más.

10 Código Máquina

Nuestro colaborador Alberto Suñer nos va a deleitar de nuevo con una de sus magníficas explicaciones de un programa en máquina, paso a paso, y su equivalente en Basic, acerca de un asunto que poca gente conoce bien: cómo se manipula el teclado en código máquina.

14

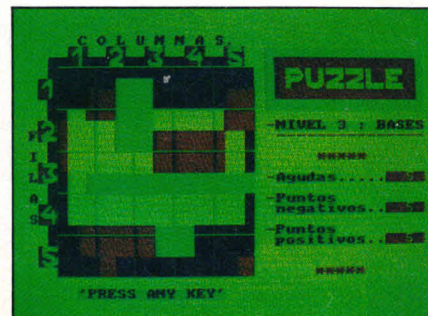
Mainframes

Todos hemos oído hablar, a menudo en las páginas de la revista, de la Quinta Generación japonesa de ordenadores, y de que la mayor dificultad para su advenimiento es el hardware, o era, porque la **«Conexión Machine»** acaba de nacer.

26

Serie Oro

Si tienes un Amstrad, vacaciones y unos ratos perdidos para relajarte a tus anchas, y ejercitar tus neuronas, **PUZZLEMIND** es tu juego.



GIGANTES del Basket

DA

LA

TALLA

Gigantes te espera cada martes en tu kiosco. Una revista imprescindible para todo aficionado. Gigantes del Basket te ofrece las últimas noticias, las mejores fotos, las opiniones que te interesan, las entrevistas que esperas, las anécdotas que te divierten y los comentarios que te gusta conocer. Pídelas todas las semanas, porque Gigantes da la talla.



UTENSILIOS, UTENSILIOS

De la mano de Mycrobyte nos llegan una serie de «**artefactos**» que, en mi opinión, son imprescindibles para trabajar a pleno rendimiento y con gran comodidad en tu ordenador.

Mycrobyte parece ser una de las propias empresas que se ha molestado en abordar, y tratar de resolver, dos problemas que al final concluyen en lo mismo:

- la ergonomía
- el concepto de «**entorno de trabajo**».

Respecto al primero, está muy de moda en estos días, y viene a decir algo muy simple: ya que usted tiene que usar un ordenador, hágalo con la mayor comodidad posible.

El segundo expresa un aserto parecido: puestos a manejar un **Amstrad**, organice racionalmente su espacio físico de trabajo, de forma que el aprovechamiento y funcionalidad de lo que le rodee sea máxima.

Nuestro más caluroso aplauso al filtro de carbono de 14 pulgadas que presenta Mycrobyte; está pensado para colocarse sobrepuesto a la pantalla de un monitor de fósforo verde y es realmente efectivo: el agotamiento visual se ve reducido en gran medida, y resulta hasta placentero mirar el normalmente brillante y dañino monitor.

¡Cuidado! Tampoco hay que concluir que el usuario asiduo de un **Amstrad** fósforo verde vaya a quedar inevitablemente ciego a plazo fijo. Simplemente decimos que es un hecho demostrado que los monitores de ordenador dañan la vista, y que es una precaución sensata protegerse de brillos y «luminotecnias».

A la hora de organizar con eficacia su espacio de trabajo, pocas cosas más útiles hay que una sencilla plataforma de plástico sobre la que se coloca la impresora, de tal modo que el ominoso «taco» de papel de la misma, se ubica justo debajo de ella, en el hueco dejado por el soporte y la mesa.

Esto puede parecer el cuento de Caperucita, pero al menos en mi mesa de trabajo, normalmente atesta-

da de papeles, artículos, fotos, etc, ha sido una especie de bendición Pascal: aparentemente ineficaz pero resultona. No me atrevería a decir que «yo no puedo estar sin él»; sin embargo, creo que es un elemento que debiera estar, sin discusión, en cualquier mesa de trabajo que incluya impresora con probabilidades de estorbar.

Más cosas. ¡Ah, sí! Para mi gusto, el «**periférico estrella**» de esta noticia es... ¡una percha!

Cuando lo vi por primera vez, una especie de tira de plástico blanco, retorcida de tal modo que nada tiene

Primera PLANA

co que sujeta un indeterminado número de papeles a dicha percha, de tal modo que es imposible no advertirlos.

Entre usted y yo, así recordé que tenía que escribir esta noticia.

Cuando una revista, a través de uno de sus redactores, alaba una serie de productos, inevitablemente surgen las suspicacias, sobre todo si el escritor se siente entusiasmado y sorprendido de aquello sobre lo que está escribiendo.

Bien, me siento entusiasmado y sorprendido. Como el juicio de uno, a pesar de que se base, como en este caso, en el continuado uso de los productos, siempre es subjetivo y discutible, los lectores y usuarios juzgarán. De cualquier forma, creo que ya



que envidiar a la más primorosa cinta de Moebius, mi perplejidad fue considerable; a punto estuve de llamar a Mycrobyte para encargarme un par de trajes a buen precio. Afortunadamente, un examen un poco más atento reveló mi torpeza, porque la «percha» es una gran idea que resuelve el difícil problema de hacer obvio lo importante e inaplazable en una mesa de trabajo: esta tira posee un velcro que se pega al ordenador y, además, incluye un clip de plásti-

es hora de que alguien se diera cuenta que existen más cosas acerca del **Amstrad** aparte del software, por muy bueno que sea.

Tema de precios: Ahí va:

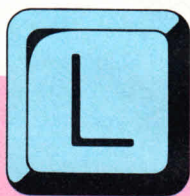
- filtro carbono: 4.900+IVA
- portadocumentos (percha): 714+IVA
- soporte impresora: 4.975+IVA

MICROBYTE está en:
San Gerardo, 59. Madrid 35.
Tel.: (91) 656 50 02

MAS RAPIDO Y ELEGANTE: ON GOSUB

En un gran número de programas se ofrece al usuario la posibilidad de elegir una de las opciones que aparecen en la pantalla. Es lo que se llama «MENU».

El Amstrad tiene preparada en su memoria una instrucción que nos va a permitir gestionar o tratar la elección que hemos hecho de un modo rápido y lleno de claridad y sencillez. Se trata de ON GOSUB.



La construcción y presentación en la pantalla de lo que hemos llamado «MENU» no tiene por qué darle ya grandes problemas. Conociendo el manejo de los comandos LOCATE, PRINT, CLS, etc., podremos adaptar cualquiera a nuestras necesidades y gustos, así como hacer una presentación en la pantalla que sea estética, clara y agradable.

Por ejemplo, mediante el Programa 1 conseguiremos visualizar un «menú» que nos ofrece la posibilidad de escoger una entre cuatro operaciones aritméticas.

```
10 REM PROGRAMA 1
20 CLS
30 PRINT TAB(10)"MENU DE OPERACIONE
S"
40 LOCATE 10,5:PRINT"1.- SUMAR"
50 LOCATE 10,9:PRINT"2.- RESTAR"
60 LOCATE 10,13:PRINT"3.- MULTIPLIC
AR"
70 LOCATE 10,17:PRINT"4.- DIVIDIR"
80 LOCATE 10,21:INPUT "ELIGE UNA OP
CION (1/4)",opcion
```

La parte del programa que se encarga de recoger la opción elegida es la línea 80. Su misión es primero informarnos de que existe la posibilidad de decidimos por una u otra, y después recoge nuestra elección una vez la hayamos teclado y pulsemos RETURN.

Y, ¿dónde la guarda? Se trata de una instrucción INPUT, así que controle la variable que hay al final de la línea, «opcion», y verá como allí está. Si quiere comprobarlo basta con que añada al Programa:

```
90 PRINT opcion
```

y verá que todo ha sucedido tal y como se lo hemos comentado.

Pero ésta no es la utilidad de los «menús». Aunque después de la ejecución del Programa anterior, con la línea 90 incluida, nos aparezca en la pantalla la opción elegida, la ver-

dad es que hasta ahora no hemos conseguido casi nada.

Para que sea útil, el Programa deberá encaminarse hacia una subrutina que trate, o ejecute de una manera en concreto, la acción que hayamos elegido. Recuerde que la opción se encuentra almacenada en la variable «opcion».

Queremos decir con esto que deberá analizar el valor que contenga «opcion» y dependiendo del que sea saltará a una línea de programa diferente a partir de la cual se ejecutará la rutina correspondiente.

Tradicionalmente esto se haría mediante una serie de sentencias del tipo:

```
IF opcion=número THEN GOSUB línea
```

Supongamos que la rutina que nos va a resolver todas las sumas que queremos se encuentra a partir de la línea 1000, la de restar a partir de la 2000, la de multiplicar de la 3000 y la de dividir de la 4000. Obtendríamos una serie de rutinas que muy esquemáticamente podría ser algo parecido al Programa 2.

```
900 REM PROGRAMA 11
1000 REM SUMAR
1010 CLS
1020 PRINT"*** AQUI SE SUMA ***"
1030 RETURN
2000 REM RESTAR
2010 CLS
2020 PRINT"*** AQUI SE RESTA ***"
2030 RETURN
3000 REM MULTIPLICAR
3010 CLS
3020 PRINT"*** AQUI SE MULTIPLICA ***"
3030 RETURN
4000 REM DIVIDIR
4010 CLS
4020 PRINT"*** AQUI SE DIVIDE ***"
4030 RETURN
```

Antes de continuar le proponemos que mezcle (con la instrucción MERGE) los dos programas anteriores, o bien se los teclee todo seguido, uno a continuación de otro de un tirón



y respetando sus respectivos números de línea.

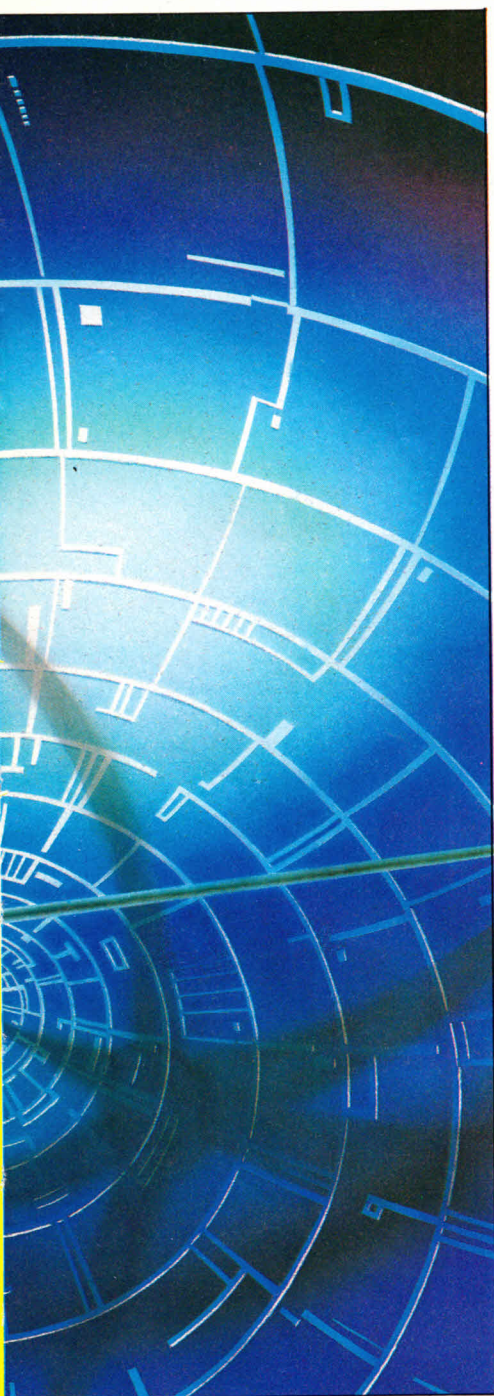
Una vez lo hayamos hecho, si suponemos que la operación a realizar es la suma, tendremos que echar una ojeada al menú y seguir sus instrucciones, es decir, pulsar la tecla número 1 y a continuación RETURN.

La línea 80 además de permitirnos introducir datos desde el teclado, también asigna el valor que nosotros hayamos elegido en la variable «opcion».

Y ahora llega el momento en el que tendríamos que dirigirnos a la subrutina de sumar, ya que ésta es la opción elegida. Para ello bastaría con añadir al programa la siguiente línea.

```
90 GOSUB 1000
```

y hacia allí iríamos.



Pero de esta manera siempre estaríamos sumando dos o más números, puesto que el salto a la subrutina de sumar se hace de un modo **«incondicional»**, o sea, el programa salta todas las veces que lo ejecutemos a la misma rutina, ya que no analizamos ninguna condición que nos diga si debemos o no hacerlo.

Vemos que el problema estaría ahora en la forma de encaminar nuestra elección hacia alguna de ellas. Pero habíamos convenido en hacerlo mediante instrucciones IF... THEN, ¿no es así?

Pues manos a la obra. En vez de la línea anterior, añadamos al programa esta otra orden:

```
90 IF opción=1 THEN GOSUB 1000
```

En ella primero analizamos si el valor que contiene la variable **«opción»** es igual a 1 y en caso afirmativo el ordenador ejecuta lo que sigue a la palabra clave THEN: un salto a la rutina encargada de sumar.

Y, ¿cuándo ocurrirá así? Pues cuando, como hemos dicho anteriormente, hayamos pulsado la tecla del 1 siguiendo los consejos del Menú. **«Opción»** contiene un 1 y al cumplirse la condición de la línea 90, el programa va a la rutina que comienza en la 1000.

A continuación se ejecutarán todas las instrucciones contenidas entre las líneas 1000 y 1030. Nosotros hemos colocado en ellas una especie de síntesis de lo que allí ocurriría: sólo nos aparecerá un mensaje en la pantalla. En realidad nos deberían pedir los sumandos, realizar la operación y ofrecer o visualizar los resultados. Pero eso se lo dejamos como tarea. ¡No se olvide hacerla a su forma y medida! Verá como saca unas muy buenas conclusiones de sus propios experimentos.

Siguiendo por la subrutina llegamos a la línea:

```
1030 RETURN
```

Al ejecutarla, el ordenador devuelve el control al punto del programa situado inmediatamente detrás del lugar desde donde se invocó la subrutina, es decir, al final de la línea 90, y a partir de allí continuará su ejecución secuencial.

Pero, ¿qué pasa cuando no hayamos elegido la opción 1 del menú? El ordenador llegará, como antes, a la línea 90. Ahora no se cumple la condición, ya que en **«opción»** estará almacenado un valor diferente de 1, y por tanto no se ejecutará lo que hay colocado después de THEN, sino que saltará a la línea siguiente.

Nuestro programa tendrá que tratar este nuevo caso, así que tendremos que añadirle nuevas instrucciones para que así lo haga.

Si tecleamos un 2, la opción que hemos elegido en esta ocasión es la operación **«resta»**. En la variable **«opción»** tendremos guardado el número 2.

La rutina que trata la resta es la que comienza a partir de la línea 2000. Piense un poquito. ¿Cómo tendría que ser la que siga a la 90 para que analice si **«opción»** tiene un 2 y que en caso de ser así el programa salte a la rutina de restar?

Seguro que se parece a:

```
100 IF opción=2 THEN GOSUB 2000
```

Y de la misma forma podemos generalizar esto al resto de operaciones y añadir al programa que tenemos entre manos las siguientes líneas:

```
110 IF opción=3 THEN GOSUB 3000
```

Y

```
120 IF opción=4 THEN GOSUB 4000
```

y, por ejemplo, cuando pulsemos oelijamos la opción 3 del menú el **Amstrad**, al llegar

Primeros PASOS

a la línea 110, encaminará la ejecución del programa hacia la rutina de la línea 3000 y se pondrá tranquilamente a multiplicar, tal como su programador se lo ha indicado.

Tenga en cuenta una cosa. En este último caso, cuando **«opción»** contenga un 3, el programa pasa también por las líneas 90, 100 y 120, pero, dado que no se cumple ninguna de las condiciones allí recogidas, no se ejecuta la parte de la instrucción que sigue a THEN.

Una vez que hayamos realizado la operación elegida, si quiere que el ordenador vuelva a ofrecer el menú al usuario para que éste continúe efectuando operaciones, tendrá que escribir una nueva línea al programa. Bastará con hacer un salto, de nuevo, a la primera instrucción que nos confeccione el menú:

```
140 GOTO 20
```

Haga todas las pruebas que quiera con este programa (el original más las líneas añadidas) y cuando esté seguro de su buen funcionamiento, sávelo rápidamente en una cinta o en un disco.

Pero aunque trabaje correctamente y haga todo lo que le pidamos, el programa así construido presenta algunos inconvenientes que quizá nos originen grandes problemas.

Por ejemplo, el valor de la variable **«opción»** no podrá ser modificado en alguna de las subrutinas que realizan las operaciones propiamente dichas. En nuestro caso, esto no va a ocurrir nunca, pero quizá en alguna ocasión será necesario hacerlo.

Cuando así ocurra, al retornar (tras RETURN) a una de las líneas comprendidas entre la 90 y la 120, pudiera ser que se cumpliera ahora alguna de las condiciones restantes, con lo que el programa saltaría de nuevo a una opción no deseada.

Para que lo vea de una manera más gráfica, añada la línea:

```
1025 opción=3
```

y ejecutémole de nuevo, pulsando la tecla 1 y RETURN para que la opción elegida sea la que hace sumas, o al menos así lo indica, y saltemos a la rutina de la línea 1000.

En ella cambiamos el valor de **«opción»** (un 1) por 3, según la nueva línea 1025.

Al retornar el programa principal lo hacemos a la 100 para seguir linealmente.

Recuerde que **«opción»** contiene ahora un 3. Así que al llegar a la línea 110 resulta que se cumple la condición allí recogida y la ejecución salta a la rutina de multiplicar.

Resumiendo, resulta que hemos elegido la opción 1 (SUMAR) y el programa pasará erróneamente por dos operaciones: sumar y multiplicar. Y todo ha sido debido a que hemos

alterado el valor de la variable «**opción**» escogido. ¿Más claro?

Otro inconveniente radica en el hecho de que si existe un número elevado de opciones, el programa contendría una cantidad de líneas semejantes a la 90 muy grande, y por tanto estaría ocupando una parte de la memoria que pudiera resultar necesaria para otras cosas.

Además la velocidad de ejecución del programa se vería disminuida en gran manera, ya que tendría que recorrer necesariamente todas las líneas que investigan si la variable «**opción**» lleva un valor u otro para tratarlo convenientemente. No importa que encuentre la que le corresponde, ¡tiene que recorrerlas todas!

Por ello es casi necesario encontrar una instrucción que nos resuelva en parte, o totalmente, estos inconvenientes.

La mayoría de los lenguajes científico-técnicos, poseen unas instrucciones o sentencias (del tipo CASE) que solucionan situaciones similares. Dependiendo del valor almacenado en una determinada variable, el control de ejecución pasará a un punto u otro del programa. Es de lo que se trata, ¿no?

Amstrad incorpora a su BASIC un comando de gran potencia con el que todos los problemas que antes enunciábamos serán un poco más pequeños y además los programas en los que esté contenida, estarán dotados de una mayor velocidad de ejecución.

Y esta instrucción es:

ON (en, con, sobre) GOSUB (ir a la rutina)

Naturalmente, cuando el ordenador se encuentre con esta instrucción, con poner la preposición ON no es suficiente, ya que tendremos que indicarle «**sobre**» qué, «**en**» qué o «**con**» qué criterio tendrá que decidir ir a una subrutina o a otra.

Y para decirselo emplearemos una variable numérica o una expresión del mismo tipo que el micro pueda evaluar. Según el valor que toma esta variable (o expresión), el programa se encaminará hacia la rutina asociada a dicho valor.

Por tanto, con lo que tenemos hasta ahora tampoco sería suficiente. Tras la palabra clave **GOSUB** habrá que especificar cada uno de los números de las líneas del programa a las que deseemos que salte la ejecución dependiendo del valor de la expresión.

Con esto bastaría. Por tanto, el formato completo de la sentencia sería:

ON expresión GOSUB lista de números de línea

¿Cómo decide ahora a cuál de las líneas indicadas debe ir? El **Amstrad**, al ejecutar una instrucción ON... GOSUB procede de la siguiente forma:

Primero evalúa la expresión que hayamos colocado entre ON y GOSUB, es decir, calcula el valor numérico que toma. En caso de ser una variable es evidente que la evaluación



es bastante sencilla: es suficiente con tomar el valor que contiene.

A continuación puede tomar varios caminos. Si el resultado de los cálculos es cero, el ordenador hace caso omiso de los números de línea que hay detrás de **GOSUB** y la ejecución del programa continúa secuencialmente por la línea siguiente.

Para comprobarlo, teclee el Programa 3, que no debe presentarle ninguna dificultad:

```
10 REM PROGRAMA III
20 opción=0
30 ON opción GOSUB 100
40 PRINT"NO HAGO EL SALTO"
50 END
100 PRINT"HE HECHO EL SALTO"
110 RETURN
```

Cuando lo ejecute, en la pantalla le aparecerá un mensaje indicando qué es lo que hace el ordenador:

No hago el salto

¿Por qué es así? Observe, en la línea 20 damos a la variable «**opción**» el valor 0. Al hacer la evaluación de la expresión (en este caso variable solamente) en la línea 30 nos encontramos que el resultado es cero, ya que esto es lo que vale «**opción**».

Entonces, según hemos dicho, el programa continuará por la línea 40, la que nos visualiza el mensaje, para que después el programa termine con el END de la 50. ¡No ha pasado por la rutina de la línea 100!

Sustituya la 20 por:

20 opción=20

Ahora el valor de la expresión es superior al número de líneas del programa citadas tras el **GOSUB**. ¿Qué pasará? Ejecútelos.

Aparentemente vemos que las consecuencias son las mismas que las obtenidas en el caso anterior, aunque ahora el resultado de la evaluación de «**opción**» sea distinto de cero.

Pues, créanos. Verdaderamente ocurre lo mismo en ambos casos. Así que podemos resumir estos dos caminos en uno solo. Cuando el valor que toma «**opción**» sea igual a cero o mayor que la cantidad de números de línea colocados detrás del **GOSUB**, el **Amstrad** actúa de la misma manera: no ocurre nada y la ejecución continúa linealmente en la instrucción siguiente.

Veamos los casos de verdad interesantes que serán aquellos en los que el ordenador salte a una de las subrutinas cuya línea de comienzo sea una de las colocadas en esta nueva instrucción ON GOSUB.

Si tras evaluar la expresión ésta toma el valor 1, al llegar a una instrucción del tipo:

ON... GOSUB

el programa saltará a la subrutina que empieza en la línea que esté colocada en primer lugar tras el **GOSUB**.

Sustituya en el Programa anterior la línea 20 por:

20 opción=1

para forzar a que la variable «**opción**» valga 1 cuando sea evaluada en la línea 30.

Vea que si ejecuta el programa, con este cambio, sí pasaremos por la subrutina que comienza en la línea 100, ya que ahora sí se visualiza:

«He hecho el salto»

aunque después, al continuar el programa por la línea 40 tras recorrer la rutina, parezca que no ha sido así por el mensaje contradictorio que nos aparece a continuación. No le haga caso, analice este último programa detenidamente y vea y comprenda por qué es así.

Cuando el resultado de la evaluación sea 2, el ordenador se encaminará hacia el número

de línea colocado en segundo lugar tras el **GOSUB**, si es que existe, y así sucesivamente.

Generalizando podríamos decir que si el valor de la variable o expresión está comprendido entre 1 y el número de líneas que hayamos citado al final de ON... GOSUB, la ejecución del programa saltará a la línea situada (tras GOSUB, claro) en la posición indicada por el valor de la variable.

Le proponemos que con el Programa 4 realice todas las pruebas que crea oportunas hasta que le queden las ideas suficientemente claras.

```

10 REM PROGRAMA IV
20 CLS
30 INPUT "VALOR DE LA VARIABLE: ",o
PRINT
40 ON opcion GOSUB 100,200,300
50 INPUT "PULSA RETURN PARA CONTINUAR",pepe
60 GOTO 20
100 PRINT"HAS TECLADO EL 1"
110 RETURN
200 PRINT"HAS TECLADO EL 2"
210 RETURN
300 PRINT"HAS TECLADO EL 3"
310 RETURN

```

En la línea 40 se encuentra el ON... GOSUB. Observe que al final de esta línea sólo hay tres de comienzo de subrutinas, así que para saltar a alguna de ellas, la variable «**opción**» necesitará tener un valor comprendido entre 1 y 3.

Pruebe también los valores cero y superiores a 3 y vea los resultados.

En el caso de teclear un valor negativo o superior a 255, el tratamiento no es el mismo. Si el ordenador encuentra que «**opción**» está fuera del margen 0-255 nos contesta con un mensaje de error que deja las cosas suficientemente claras:

Improper argument in 40

Experimente estos valores en el Programa anterior.

Y para despedirnos volvamos al Programa de las operaciones que resultó de la mezcla que hicimos al principio del artículo.

Manteniendo una estructura semejante formaremos ahora el Programa 4. Hemos sustituido todos los IF... THEN por un solo ON GOSUB.

¿Cuál de los dos le parece más sencillo? Vea cómo funciona para cualquier valor que le tecleemos. Esperamos que no encuentre ninguna dificultad en ver cómo trabaja.

Naturalmente el uso de ON GOSUB no se limita únicamente a programas con menús, pero éste es un ejemplo muy claro e ilustrativo para comprender el funcionamiento de este nuevo comando.

En posteriores artículos se analizarán otros de un funcionamiento parecido: ON GOTO, ON ERROR GOTO, y compañía nos esperan. ¿Les encontraremos?...

Primeros PASOS

```

10 REM PROGRAMA V
20 CLS
30 PRINT TAB(10)"MENU DE OPERACIONES"
40 LOCATE 10,5:PRINT"1.- SUMAR"
50 LOCATE 10,9:PRINT"2.- RESTAR"
60 LOCATE 10,13:PRINT"3.- MULTIPLICAR"
70 LOCATE 10,17:PRINT"4.- DIVIDIR"
80 LOCATE 10,21:PRINT"5.- TERMINAR"
90 LOCATE 10,25:INPUT "ELIGE UNA OPCION (1/4)",opcion
100 ON opcion GOSUB 1000,2000,3000,4000,5000
110 INPUT "PULSA RETURN PARA CONTINUAR",pepe
120 GOTO 20
1000 REM SUMAR
1010 CLS
1020 PRINT"*** AQUI SE SUMA ***"
1030 RETURN
2000 REM RESTAR
2010 CLS
2020 PRINT"*** AQUI SE RESTA ***"
2030 RETURN
3000 REM MULTIPLICAR
3010 CLS
3020 PRINT"*** AQUI SE MULTIPLICA ***"
3030 RETURN
4000 REM DIVIDIR
4010 CLS
4020 PRINT"*** AQUI SE DIVIDE ***"
4030 RETURN
5000 REM FIN
5010 CLS
5020 END

```



¿Te falta algún número?

Si estás interesado en algún número de los ya publicados por Microhobby Amstrad, realiza hoy mismo tu pedido porque ya hay algunos ejemplares agotados.

No pierdas la oportunidad de disponer de la mejor obra publicada sobre ordenadores Amstrad. En todos sus números encontrarás interesantes artículos de iniciación, pokes, trucos, curso de código máquina, etc...

¡No te pierdas detalle!
Recorta o copia el cupón que aparece cosido en las páginas de la revista.

LA LECTURA DEL TECLADO

Entraremos hoy en el estudio de otro capítulo importante dentro de la programación en código máquina, el manejo del teclado. Veremos cómo se puede comprobar si una tecla está pulsada o no.



En el **Amstrad** el teclado se lee cincuenta veces por segundo y se construye un buffer de teclado en el cual se indica qué teclas están o no están pulsadas.

Para descifrar el teclado, existen tres niveles de operación: el primero de ellos se encarga de leer el teclado, el siguiente nivel convierte cada una de las teclas pulsadas a números de teclas y el último de estos niveles convierte dichos números en caracteres ASCII.

En el presente capítulo hemos preparado dos programas que se encargan de chequear qué teclas se han pulsado. En concreto se observa la pulsación de las teclas '1', '2' ó '3'.

Cada uno de estos programas utiliza un sistema de lectura de teclado distinto. Uno de ellos utiliza el comando:

INKEY\$

el cual toma el valor ASCII de la tecla que se ha pulsado y, por lo tanto, distingue el carácter de la tecla pulsada ya sea mayúscula o minúscula.

El otro sistema de lectura de teclado se realiza mediante el comando:

INKEY

que no trabaja con caracteres ASCII sino con números de tecla.

Vamos a ver ahora cómo escribiríamos cada uno de los programas mencionados anteriormente en código máquina.

Estudiaremos en primer lugar cómo leer caracteres ASCII desde el lenguaje máquina. Para ello utilizaremos una rutina que nos proporciona el firmware.

Dicha rutina se encarga de tomar del buffer de teclado el carácter de la última tecla pulsada y éste se coloca en el registro A.

TOMA UN CARACTER DEL BUFFER DE TECLADO. #BB1B

Toma el código ASCII del carácter que se ha pulsado desde teclado. Dicha rutina no espera a que se pulse una tecla.

Condiciones de entrada.

No necesita ninguna condición.

Condiciones de salida.

Si se ha pulsado una tecla nos devolverá los siguientes resultados:

Flag Carry verdadero.

El registro A contiene el valor ASCII de la tecla pulsada.

Si no se ha pulsado ninguna tecla el resultado será:

Flag Carry falso.

Se corrompe el registro A.

En cualquier caso se corrompen todos los flags y se preservan los demás registros.

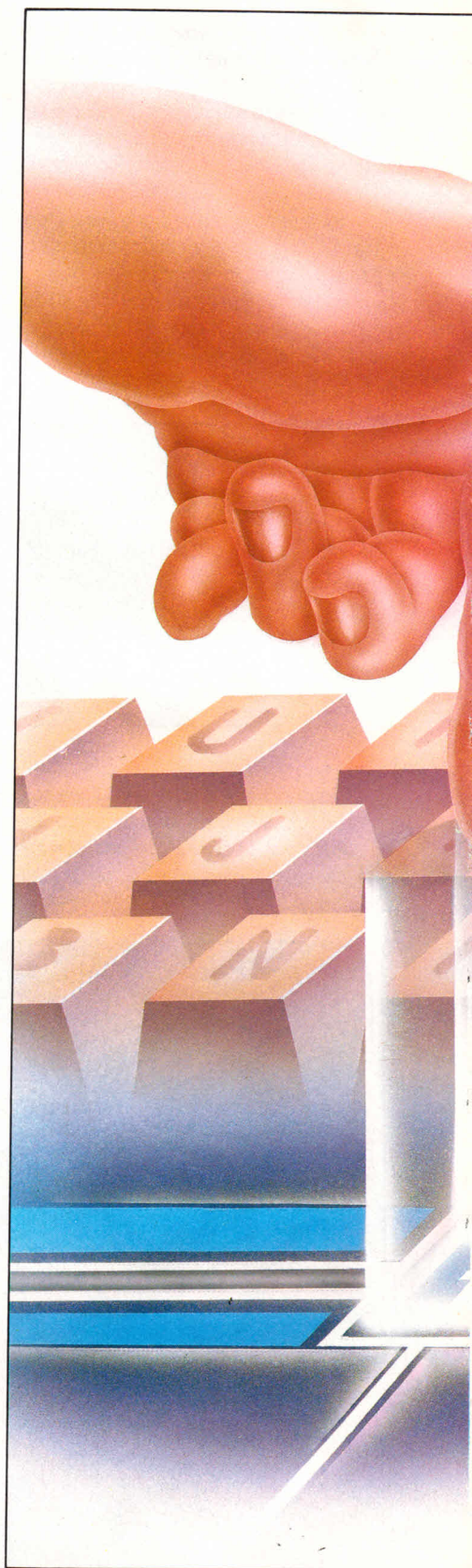
Una vez realizada esta llamada deberemos comprobar si se han pulsado alguna de las teclas que se han elegido. Primero comprobamos si se ha pulsado la tecla '1'.

CP "1"

JR Z, PRINT

si dicha tecla se ha pulsado, entonces el flag Z estará a cero y se realizará la impresión en pantalla correspondiente.

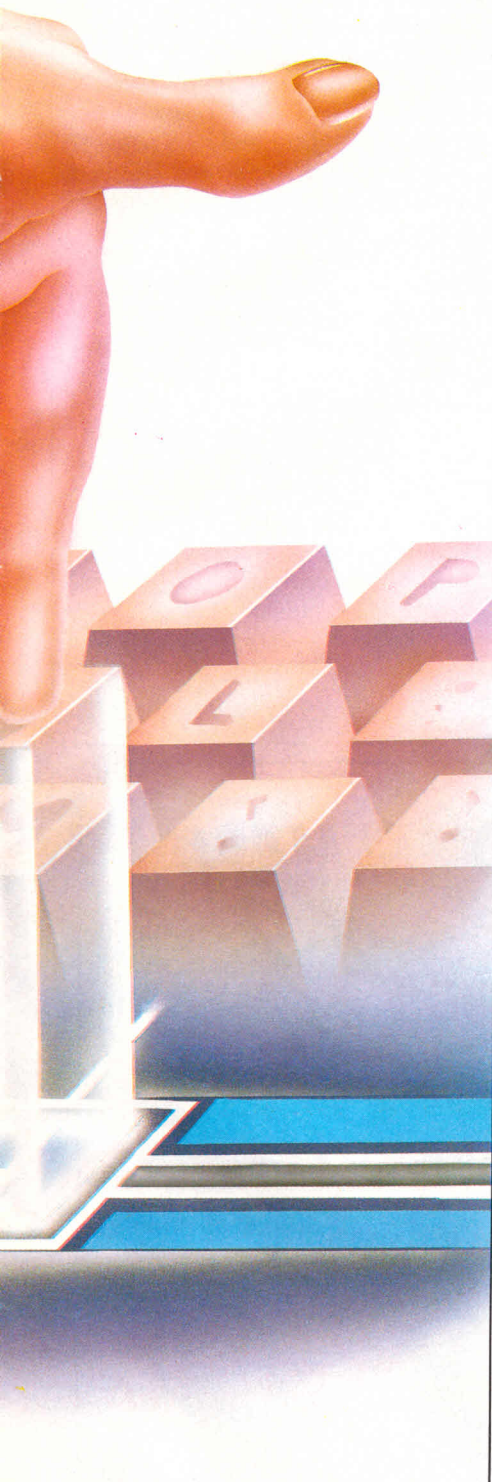
A continuación realizaremos la misma operación con las restantes teclas.



Cuando se entra en la rutina de impresión, lo primero que hacemos es preservar el registro AF:

PUSH AF

para que una vez impreso el mensaje podamos recuperarlo e imprimirlo en pantalla, ya que el acumulador



```

Hissoft GENA3.1 Assembler. Page 1.
Pass 1 errors: 00
A000 10 ORG #A000
A000 CD1BBB 20 TEC: CALL #BB1B
A003 FF31 30 CP "1"
A005 2B0A 40 JR 7,PRINT
A007 FE32 50 CP "2"
A009 2B06 60 JR 7,PRINT
A00B FE33 70 CP "3"
A00D 2B02 80 JR 7,PRINT
A00F 1BEF 90 JR TEC
A011 F5 100 PRINT: PUSH AF
A012 210CA 110 LD HL,#0A0C
A015 CD75BB 120 CALL #BB75
A01B 112CA0 130 LD DE, TXT
A01B 1A 140 PRBUC: LD A, (DE)
A01C FEEF 150 CP 255
A01E 2B06 160 JR 7,PINT1
A020 CDSABB 170 CALL #BBSA
A023 13 180 INC DE
A024 1BF5 190 JR PRBUC
A026 F1 200 PINT1: POP AF
A027 CDSABB 210 CALL #BBSA
A02A 1BD4 220 JR TEC
A02C 50554C53 230 TXT: DEFM "PULSADA LA TECLA "
A03D FF 240 DEFB 255

```

```

Pass 2 errors: 00
PINT1 A026 PRBUC A01B PRINT A011
TEC A000 TXT A02C
Table used: 69 from 139

```

```

10 MODE 1 MA049A CA
20 IF INKEY$="1" THEN A=1:GOSUB 100
30 IF INKEY$="2" THEN A=2:GOSUB 100
40 IF INKEY$="3" THEN A=3:GOSUB 100
50 GOTO 20
100 LOCATE 10,12:PRINT "PULSADA LA
TECLA";A
110 RETURN

```

```

10 FOR N=#A000 TO #A03F
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<&#192F THEN PRINT "ERROR
EN DATAS"
60 DATA 205,27,187,254,49,40,10
70 DATA 254,50,40,6,254,51,40
80 DATA 2,24,239,245,33,12,10
90 DATA 205,117,187,17,44,160,26
100 DATA 254,255,40,6,205,90,187
110 DATA 19,24,245,241,205,90,187
120 DATA 24,212,80,85,76,83,65
130 DATA 68,65,32,76,65,32,84
140 DATA 69,67,76,65,32,255,0

```

Para leer números de teclas desde lenguaje máquina, se utiliza otra rutina del firmware, la cual indica si se ha pulsado la tecla que se le ha indicado previamente en el acumulador.

CHEQUEA SI SE HA PULSADO UNA TECLA. #BB1E

Mira si se ha pulsado una tecla determinada o si se ha activado el joystick.

Condiciones de entrada.

El acumulador debe contener el número de tecla que se desea chequear.

Condiciones de salida.

Si se ha pulsado la tecla el flag de cero es falso.

Si no se ha pulsado la tecla el flag de cero es verdadero.

En cualquier caso se devuelven los siguientes valores:

El flag Carry falso.

El registro C contiene el estado de la tecla SHIFT (MAYS).

Los registros A y HL se corrompen, los demás son preservados.

Así pues, para chequear cada una de las teclas, deberemos cargar en el acumulador el número de tecla deseada y llamar a la rutina anteriormente explicada.

contiene el carácter leído desde teclado:

```

POP AF
CALL #BB5A

```

de esta forma en pantalla se imprimirá el número que se haya pulsado desde teclado.

Código MAQUINA

Dado que en primer lugar se desea mirar si se ha pulsado la tecla '1' (tecla número 64), deberemos cargar en el registro A dicho valor:

```
LD A, 64
```

CALL #BB1E

a continuación deberemos comprobar si dicha tecla se ha pulsado, para ello observamos el estado del flag cero:

```

Hissoft GENA3.1 Assembler. Page 1.
Pass 1 errors: 00
A000 10 ORG #A000
A000 3E01 20 LD A,1
A002 CD0EBC 30 CALL #BCE
A005 3E40 40 TEC: LD A,64
A007 CD1EBB 50 CALL #BB1E
A00A 2B04 60 JR 7,PAS1
A00C 3E31 70 LD A,"1"
A00E 1B1A 80 JR PRINT
A010 3E41 90 PAS1: LD A,65
A012 CD1EBB 100 CALL #BB1E
A015 2B04 110 JR 7,PAS2
A017 3E32 120 LD A,"2"
A019 1B0F 130 JR PRINT
A01B 3E39 140 PAS2: LD A,57
A01D CD1EBB 150 CALL #BB1E
A020 2BE3 160 JR 7,TEC
A022 3E33 170 LD A,"3"
A024 1B04 180 JR PRINT
A026 1BD0 190 JR TEC
A028 1BDB 200 JR TEC
A02A F5 210 PRINT: PUSH AF
A02B 210CA 220 LD HL,#0A0C
A02E CD75BB 230 CALL #BB75
A031 1145A0 240 LD DE, TXT
A03A 1A 250 PRBUC: LD A, (DE)
A03E FEEF 260 CP 255
A037 2B06 270 JR 7,PINT1
A039 CDSABB 280 CALL #BBSA
A03C 13 290 INC DE
A03D 1BF5 300 JR PRBUC
A03F F1 310 PINT1: POP AF
A040 CDSABB 320 CALL #BBSA
A043 1BC0 330 JR TEC
A045 50554C53 340 TXT: DEFM "PULSADA LA TECLA "
A05A FF 350 DEFB 255

```

```

Pass 2 errors: 00
PAS1 A010 PAS2 A01B PINT1 A03F
PRBUC A034 PRINT A02A TEC A005
TXT A045
Table used: 91 from 155

```

```

10 MODE 1 MA049B CA
20 IF INKEY(64)=0 THEN A=1:GOSUB 100
0
30 IF INKEY(65)=0 THEN A=2:GOSUB 100
0
40 IF INKEY(57)=0 THEN A=3:GOSUB 100
0
50 GOTO 20
100 LOCATE 10,12:PRINT "PULSADA LA
TECLA";A
110 RETURN

```

```

10 FOR N=#A000 TO #A057
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<&#1FBC THEN PRINT "ERROR
EN DATAS"
60 DATA 62,1,205,14,188,62,64
70 DATA 205,30,187,40,4,62,49
80 DATA 24,26,62,65,205,30,187
90 DATA 40,4,62,50,24,15,62
100 DATA 57,205,30,187,40,227,62
110 DATA 51,24,4,24,221,24,219
120 DATA 245,33,12,10,205,117,187
130 DATA 17,69,160,26,254,255,40
140 DATA 6,205,90,187,19,24,245
150 DATA 241,205,90,187,24,192,80
160 DATA 85,76,83,65,68,65,32
170 DATA 76,65,32,84,69,67,76
180 DATA 65,32,255,0,0,0,0

```

JR Z, PAS1

así pues si Z está a cero, indica que no se ha pulsado, por el contrario si Z está a uno, dicha tecla se habrá pulsado desde el teclado.

Ahora deberemos imprimirla en pantalla, para ello cargaremos en el acumulador el carácter ASCII de dicha tecla y llamaremos a la rutina PRINT para que se imprima en pantalla el número correspondiente:

```
LD A, "1"  
JR PRINT
```

Esta misma operación deberá efectuarse con las teclas restantes, indicando en el acumulador el número de cada una de dichas teclas.

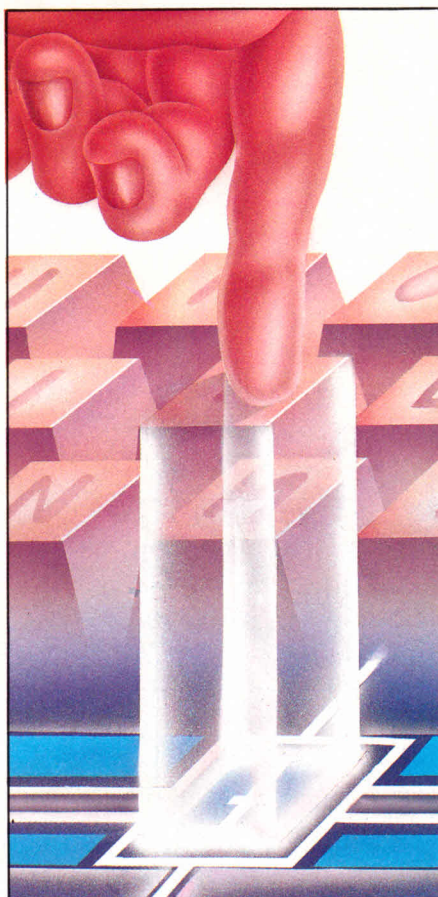
La rutina de impresión es exactamente igual a la utilizada en el anterior programa.

Aparte de las dos rutinas mencionadas para leer el teclado, existe otra que se encarga de leer el estado de los joysticks.

Dicha rutina es:

```
MIRA EL ESTADO DE LOS JOYS-  
TICKS. #BB24
```

Lee el estado de los joysticks.
Condiciones de entrada.



No necesita.

Condiciones de salida.

El registro H contiene el estado del joystick 0 y el registro L el estado del joystick 1.

Dado que en el buffer de teclado se van almacenando las teclas que pulsamos desde el ordenador, hay momentos en que en dicho buffer hay almacenadas varios valores, por lo que es muy posible que cuando deseemos leer una tecla en concreto, se produzca la lectura de otra que se había almacenado anteriormente.

Para evitar este fenómeno, existe en el firmware del **Amstrad** una rutina que se encarga de vaciar el buffer de teclado y dejarlo preparado para leer la próxima tecla que se pulse desde el teclado. Dicha rutina es la siguiente:

```
RESETEA EL BUFFER DE TECLA-  
DO. #BB03
```

Reiniciará los buffers.

Condiciones de entrada.

No son necesarias.

Condiciones de salida.

Se corrompen los registros AF, BC, DE y HL. Los demás registros se preservan.

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	
CARA 1	CARA 2
DISCMAN	AMSCAL*
EL MURO	EJECUTIVO
BEGIN 371 A 377	RAM DISC 128
MACROCOMPILER*	COMPANT
ANALISIS 38	ANALIS 37, 39 Y 40
	IDEAS 37 A 39*
* Sólo 664 y 6128.	



SAI COMBAT

ZAFIRO
CHIP

¡EL DEFINITIVO!

- 8 niveles de dificultad desde cinturón blanco a cinturón negro.
- A partir del octavo Dan puedes llegar a ser "Sai Master".
- 16 Movimientos diferentes.
- Joystick o Teclado.



LOS MEJORES MOVIMIENTOS EN EL
MEJOR PROGRAMA DE ARTES
MARCIALES

Spectrum

Amstrad

Amstrad Disk



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

MIRRORSOFT

Editado, fabricado y distribuido en España
bajo la garantía Zafiro. Todos los derechos
reservados.

THE CONNECTION MACHINE

Una máquina que, partiendo de un concepto totalmente revolucionario en el diseño de ordenadores, deja anticuado el concepto clásico de ordenador, para entrar en una nueva informática.



Un cubo de plástico de metro y medio de lado contiene en su interior el último logro de la técnica en el diseño de ordenadores.

Tras sus paneles de color negro y los miles de luces rojas destelleantes que cubren su superficie, se encuentra **THE CONNECTION MACHINE**, un ordenador que viene a revolucionar el mundo de los mainframes.

En su reciente debut público, el primer modelo de la prodigiosa máquina hizo una demostración de sus asombrosas capacidades, enfrentándose a complejos problemas que habrían mantenido ocupado durante horas a un ordenador normal.

Los siguientes tiempos y operaciones, dan idea de sus prestaciones.

En **solamente 1/20 de segundo**, es capaz de revisar el contenido de 16.000 artículos de prensa: tres meses de noticias de un periódico diario.

En **dos segundos**, transforma una imagen estereoscópica transmitida por dos cámaras de televisión, en un mapa bidimensional de curvas de contorno.

Tres minutos, es el tiempo que emplea esta poderosa máquina para diseñar el circuito de un chip de ordenador conteniendo 4.000 transistores.

En palabras de Daniel Hillis, el diseñador de ordenadores de 29 años de edad y cofundador de Thinking Machines Corp. que tiene el privilegio de ser el padre de tan portentosa criatura: **«Los ordenadores convencionales son respecto a la Connection Machine, lo mismo que una bicicleta a un avión supersónico.»**

La comparación usada por Hillis no está nada lejos de la realidad; su extraña máquina es capaz de operar a velocidades que sobrepasan el billón de instrucciones por segundo, aproximadamente la velocidad de proceso de un superordenador Cray X-MP, pero a la cuarta parte de precio.

Por otra parte, la Connection Machine ofrece la esperanza de resolver problemas en visión-máquina e inteligencia artificial, para los que los superordenadores de hoy están, por desgracia, pobremente equipados.

Según Says Stephen Squires, interlocutor de la Agencia de su investigación de proyectos avanzados de la defensa (DARPA), la oficina del Departamento de Defensa, que ha suministrado 4.7 millones de dólares para el desarrollo de los ordenadores ha dicho: «La máquina de Hillis ha sobrepasado todas nuestras expectativas.»

¿Cómo consigue la Connection Machine su impresionante velocidad? El secreto reside en su elevado número de procesadores, y en su radicalmente nuevo concepto de arquitectura interna, gracias a la cual está dotada de la suficiente flexibilidad como para realizar un gran número de operaciones simultáneamente.

La mayoría de los ordenadores construidos en los años 40, estaban diseñados para hacer una cosa a un tiempo, siguiendo el modelo concebido por John von Newman y sus colegas en 1945.

Este consistía en una unidad central de proceso de alta velocidad, conectada a una serie de células de memoria.

En palabras de Hillis:

«La arquitectura de dos cuerpos, mantiene increíblemente ocupado a la parte de silencio dedicada al proceso, que solamente ocupa el 3 por 100 del total del área de silicio em-



pleada en el ordenador; el otro 97 por 100 (dedicado exclusivamente al banco de memoria) permanece ocioso.»

De esta forma, mientras los ordenadores han crecido en capacidad, el diseño ha ido paulatinamente haciéndose más ineficaz.

Mientras es fácil expandir la memoria, la tarea de aumentar la capacidad de procesador es muy difí-



han seguido este camino, incrementando su número en valores que van desde varias unidades de proceso, hasta cientos de ellas, dejando que todas compartan en paralelo el trabajo de manipular el caudal de datos.

Connection Machine evita el problema del cuello de botella utilizando una cantidad hasta ahora inimaginable de procesadores: 65.536 unidades, las cuales, actuando simultáneamente, pueden manejar cantidades masivas de datos.

Tan importante como el número de procesadores es el hecho de que cada uno tiene asociado su pequeño banco de memoria, con lo que conseguimos que el proceso y la memoria, que antes estaban separados por un estrecho canal, estén ahora integrados en una pieza de silicio del tamaño de una uña.

Llegando aún más lejos en su revolucionaria arquitectura, cada procesador puede ser conectado directa e indirectamente con cada uno de los demás, mediante lo que podemos considerar como un auténtico sistema telefónico en miniatura, constituido por 4.096 estaciones interruptoras y una red de 24.576 líneas de comunicación, que pueden ser programadas y reprogramadas sin tener que cambiar la constitución interna del ordenador.

Estas conexiones reprogramables son las directas responsables del nombre de la máquina. Para una tarea determinada, los procesadores son electrónicamente seleccionados para adaptarse a la natural estructura de los datos a tratar.

Para simular la estructura de un componente de ordenador, integrado por 20.000 transistores, la máquina asignaría un procesador a cada transistor, consiguiendo que en vez de actualizar el estado de esos 20.000 interruptores uno por uno, como en el más puro estilo Von Neumann, el software de la Connection Machine hace que los 20.000 procesadores seleccionados para este proceso se actualicen todos a la vez.

Desafortunadamente, programar una máquina de estas características requiere una lógica conceptual que incluso muchos de los científicos de ordenadores, encuentran dificultades a la hora de ponerla en práctica.

Según el informe elaborado por DARPA, solamente uno de cada tres programadores del Departamento de Defensa pueden aceptar el reto.

En palabras de Larry Smarr, director del Centro Nacional de Aplicaciones para Superordenadores, de la Universidad de Illinois:

«En nuestra universidad tenemos una experiencia de 40 años, diseñando software para ordenadores de un solo procesador, sin embargo, el software para estas máquinas es complicado y extremadamente laborioso de escribir.»

Afortunadamente, la Connection Machine tiene poderosísimos entusiastas, entre los que cabe destacar a Marvin Minsky, perteneciente al MIT (Massachusetts Institut of Technology), pionero en la investigación en inteligencia artificial, y Claude Shannon, padre de la teoría estadística de la información.

La semana pasada, la compañía de Hillis había recibido un total de pedidos de siete de sus nuevos ordenadores, los cuales se encuentran en una escala de precios que van desde el millón de dólares para las unidades más baratas, hasta los tres millones.

Los organismos que han encargado las revolucionarias máquinas, avalan por su reconocido prestigio el brillante futuro que aguarda a Thinking Machine Corporation; dos ordenadores han sido solicitados por cada uno de los siguientes organismos: M.I.T., Perkin-Elmer y DARPA, y uno más ha sido encargado por la Universidad de Yale.

Pero la compañía de Hillis no es la única en este campo; Intel y Floating Point Systems, están construyendo ordenadores de procesadores paralelos llegando a límites tan ambiciosos como los de Thinking Machine.

Por otra parte, un par de compañías de reciente creación Encore y Sequent, están intentando captar un mercado orientado hacia máquinas paralelas más modestas.

Mientras tanto, equipos de investigación a lo largo de todos los Estados Unidos están experimentando con diseños aún más radicales.

Entre ellos, AT&T Bell está desarrollando circuitos para ordenadores que imitan la acción de los billones de neuronas que componen el cerebro humano.

Y la cosa no queda ahí, hay un gran camino por recorrer y nos encontramos en la hora de experimentar; los resultados obtenidos cambiarán por completo los conceptos de la informática clásica.

El futuro está a la vuelta de la esquina.

cil, dando como resultado el que gigantescas máquinas estén forzadas a conducir su flujo de datos por una estrecha senda, conocida en el mundo de los científicos de ordenadores por el «cuello de botella de Von Neumann».

Una forma de ensanchar el cuello de botella es añadir más procesadores. En los pasados cinco años, decenas de diseñadores en este campo

SCROLL PIXEL A PIXEL A IZQUIERDA, DERECHA, ARRIBA Y ABAJO

Hasta el momento hemos tenido la oportunidad de aprender a realizar «scrolls» de pantalla carácter a carácter, gracias a algunos artículos aparecidos en esta misma revista. Ahora le toca el turno al scroll pixel a pixel.



Hemos creído conveniente abordar este tema, ya que consideramos que una rutina de este tipo puede ser mucho más atractiva en la realización de presentaciones e incluso en la programación de juegos, porque produce una rotación de la pantalla de una forma mucho más suave que la pueda producir un scroll carácter a carácter.

Para la facilidad en el manejo de dicha rutina, hemos considerado oportuno crear varios comandos RSX con los cuales podremos acceder a cada uno de los «scroles» posibles, que son los siguientes:

- Scroll pixel a pixel a derecha
- Scroll pixel a pixel a izquierda
- Scroll pixel a pixel arriba
- Scroll pixel a pixel abajo

Comandos RSX de scroll

Vamos ahora a estudiar cada uno de dichos comandos y cómo se deben utilizar. En primer lugar tenemos el que se encarga de producir un scroll hacia la derecha:

IROLLD,X,Y,A,B

Los parámetros que aparecen tienen la siguiente significación:

X. Coordenada vertical de la esquina superior izquierda del bloque de pantalla sobre el que se va a efectuar el scroll.

Y. Coordenada horizontal de la esquina superior derecha del bloque de pantalla sobre el que se va a efectuar el scroll.

A. Altura del bloque sobre el que se efectúa el scroll.

B. Anchura del bloque sobre el que se efectúa el scroll.

Así pues, dicho comando efectúa un scroll a la derecha del bloque de pantalla indicado por los anteriores parámetros.

El comando encargado de efectuar el scroll a izquierda de la pantalla es el siguiente:

IROLLI,X,Y,A,B

Su ejecución provoca un scroll a la izquierda del bloque de pantalla delimitado por los parámetros que lo acompañan, que tiene el mismo significado que los explicados anteriormente.

El siguiente comando es el encargado de efectuar el scroll hacia arriba:

IROLLU,X,Y,A,B

cuyos parámetros son idénticos a los anteriores.

Por último, tenemos la instrucción que se encargará de producir un scroll abajo de la pantalla:

IROLLA,X,Y,A,B

Pixel a pixel: ¡en serio!

Debemos tener en cuenta que la ejecución de cada uno de los anteriores comandos provoca el desplazamiento de un pixel del bloque de pantalla indicado por los parámetros; así pues, si deseamos efectuar el desplazamiento de un carácter completo, deberemos ejecutar dichos comandos 8 veces.

Veamos, por ejemplo, cómo deberíamos confeccionar un programa en Basic que se encargará de desplazar hacia la derecha un bloque de pantalla cuya coordenada vertical fuera la 15, con una coordenada horizontal de 5, y cuya altura y anchura fueran 10 y 20 respectivamente. Para producir el efecto que deseamos deberíamos escribir el siguiente comando:

IROLLD,15,5,10,20

su ejecución produciría el desplazamiento de un pixel hacia la derecha de la pantalla. Ahora bien, si lo que se desea en realidad es producir un desplazamiento de tres caracteres por



ejemplo, entonces se debería escribir un programa como el siguiente:

```
10 FOR N=1 TO 24  
20 IROLLD,15,5,10,20  
30 NEXT
```

Para aquéllos que no les haya quedado claro el funcionamiento de dichos comandos, pueden revisar el programa demostración que acompaña a este artículo.

Dicho programa efectúa un scroll en todos los sentidos de una pantalla de presentación que hemos preparado a tal efecto.

La rutina en código máquina, instala en primer lugar los comandos mencionados anteriormente en el sistema, para que éstos sean utilizables desde Basic, de la forma habitual:



Lo explicado anteriormente serviría por ejemplo para producir un scroll en el modo 2 de **Amstrad**, pero como hemos dicho anteriormente, nosotros deseamos trabajar en modo 1.

Pues bien, como todos sabemos, el tratamiento del color que lleva el **Amstrad**, implica el emparejamiento de bits dentro de un mismo byte. Así, en modo 1, el emparejamiento de bits es el siguiente:

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

por lo tanto, la rotación se debe producir de la forma que hemos explicado anteriormente, pero teniendo además en cuenta que cuando se trasvase el bit 7 de un byte a otro, deberemos además trasvasar el bit con el que se encuentra emparejado (en este caso será el bit 3).

Del mismo modo, cuando se trasvase el bit 0, se deberá pasar con él, el bit 4. De esta forma se podrá conseguir un scroll correcto de la pantalla.

En lo referente a los «scrolls» arriba o abajo, el trabajo resulta mucho más sencillo, ya que en este caso lo que debemos desplazar son bytes, con lo cual nos ahorramos el trabajo de rotación de bits.

Dado que estas dos últimas operaciones no trabajan con bits, su funcionamiento será válido en cualquiera de los modos de pantalla con que puede trabajar el **Amstrad**.

Arriba y abajo

El método es muy sencillo, ya que lo único que se hace es provocar un LDIR de unas direcciones a otras de pantalla, con lo cual se logra producir el efecto de desplazamiento pixel a pixel.

Sólo queda por explicar los mensajes de error que se nos pueden mostrar en la utilización de los nuevos comandos; éstos son los siguientes:

FALTAN PARAMETROS: este mensaje nos indica que hemos omitido alguno de los parámetros necesarios para dicho comando.

FUERA DE RANGO: indica que los parámetros establecidos provocan un scroll fuera de los límites de la pantalla.

ERROR EN PARAMETROS: indica que alguno de los parámetros vale cero. (Ninguno de ellos puede ser cero).

Veamos por último cuáles son los pasos que debemos realizar para el correcto funcionamiento de los comandos RSX mencionados anteriormente.

En primer lugar deberemos copiar el listado ensamblador que aparece junto a este artículo, para lo cual deberemos estar en disposición de algún ensamblador.

Para aquellos que no posean dicha herramienta de programación, y deseen pasar a cinta o disco dichos comandos RSX, aparece también un cargador Basic.

PROGRAMACION

Cómo manipular el cargador

Los que elijan esta segunda opción, deberán teclear dicho cargador y ejecutarlo; si una vez hecho esto no aparece ningún mensaje de error, se procederá a su grabación de la forma siguiente:

SAVE "SCROLL",B,&A000,700

Una vez tengamos salvado dicho programa en disco o cinta, y deseemos ejecutarlo, deberemos escribir un programa cargador en Basic con el siguiente:

10 MEMORY &9FFF
20 LOAD "SCROLL",&A000
30 CALL &A000

Cuando dicho programa se haya ejecutado, estaremos en disposición de utilizar los nuevos comandos RSX.

LISTADO DE ENSAMBLADO

```

A888      18      ORG  #A888
A888      0189A8  28      LD  BC,TABLA
A883      212CA8  38      LD  HL,SPACE
A886      C3D1BC  48      JP  #BCD1
A889      17A8   58      TABLA: DEFW #NAME
A88B      C38A8  68      JP  SCROLL
A88E      C38A8  78      JP  SCROLL1
A811      C3F8A8  88      JP  SCROLLU
A814      C37FA1  98      JP  SCROLLD
A817      524F4C4C 108     NAME: DEFW "ROLL"
A81B      C4     118     DEFB "D"+#B8
A81C      524F4C4C 128     DEFW "ROLL"
A828      C9     138     DEFB "I"+#B8
A821      524F4C4C 148     DEFW "ROLL"
A825      D5     158     DEFB "D"+#B8
A826      524F4C4C 168     DEFW "ROLL"
A82A      C1     178     DEFB "A"+#B8
A82B      88     188     DEFB 0
A82C      88     198     SPACE: DEFS 4
          288     ;SCROLL DE PANTALLA EN MODO 1
          218 ;
          228 ;
A838      FE84   238     SCROLL: CP  4
A832      C267A2  248     JP  NZ,BADCOM
A835      CD24A2  258     CALL BUGS
A838      D8     268     RET  C
A839      D07E88  278     LD  A,(IX+#)
A83C      3219A2  288     LD  (COLIM),A
A83F      D07E82  298     LD  A,(IX+2)
A842      3218A2  308     LD  (FILAS),A
A845      D06E84  318     LD  L,(IX+4)
A848      D06686  328     LD  H,(IX+6)
A84B      CDE9A8  338     CALL CALCU
A84E      3A18A2  348     LD  A,(FILAS)
A851      47     358     LD  B,A
A852      C5     368     BUC2: PUSH BC
A853      E5     378     PUSH HL
A854      8E88   388     LD  C,B
A856      E5     398     BUC1: PUSH HL
A857      AF     408     XOR  A
A858      3283A8  418     LD  (STORE+1),A
A85B      3A19A2  428     LD  A,(COLIM)
A85E      47     438     LD  B,A
A85F      CD77A8  448     BUC:  CALL SCROL
A862      23     458     INC  HL
A863      18FA   468     DJNZ BUC
A865      E1     478     POP  HL
A866      118888  488     LD  DE,2848
A869      19     498     ADD  HL,DE
A86A      8D     508     DEC  C
A86B      79     518     LD  A,C
A86C      28E8   528     JR  NZ,BUC1
A86E      E1     538     POP  HL
A86F      115888  548     LD  DE,B8
A872      19     558     ADD  HL,DE
A873      C1     568     POP  BC
A874      18DC   578     DJNZ BUC2
A876      C9     588     RET
A877      7E     598     SCROLL: LD  A,(HL)
A878      E611   608     AND  17
A87A      17     618     RLA
A87B      17     628     RLA
A87C      17     638     RLA
A87D      57     648     LD  D,A
A87E      7E     658     LD  A,(HL)
A87F      E4EE   668     AND  238
A881      1F     678     RRA
A882      F688   688     STORE: OR  0
A884      77     698     LD  (HL),A
A885      7A     708     LD  A,D

```

se prepara la tabla de saltos junto con el nombre de los nuevos comandos.

Funcionamiento de las rutinas de scroll

Vamos a estudiar ahora cómo funcionan cada una de las rutinas de scroll.

Un desplazamiento pixel a pixel se produce debido a la rotación de los bits que componen un byte, colocándose el bit 0 de un byte, en el bit 7 del siguiente, o bien en el bit 7 de dicho byte se coloca en el bit 0 del anterior, según sea el scroll a izquierda o derecha.

```

A086 3203A0 718 LD (STORE+1),A
A089 C9 728 RET
A08A FE84 738 SCROLL: CP 4
A08B C267A2 748 JP NZ,BADCOM
A08F CD24A2 758 CALL BUGS
A092 D8 768 RET C
A093 D07E00 778 LD A,(IX+0)
A096 6F 788 LD L,A
A097 321BA2 798 LD (COLUM1),A
A09A D07E82 808 LD A,(IX+2)
A09D 321AA2 818 LD (FILAS1),A
A0A8 D07E84 828 LD A,(IX+4)
A0A3 85 838 ADD A,L
A0A4 3D 848 DEC A
A0A5 6F 858 LD L,A
A0A6 D06686 868 LD H,(IX+6)
A0A9 CDE800 878 CALL CALCU
A0AC 3A1AA2 888 LD A,(FILAS1)
A0AF 47 898 LD B,A
A0B8 C5 908 SAL2: PUSH BC
A0B1 E5 918 PUSH HL
A0B2 BE88 928 LD C,8
A0B4 E5 938 SALL: PUSH HL
A0B5 AF 948 XOR A
A0B6 32E1A0 958 LD (ALMAC+1),A
A0B9 3A1BA2 968 LD A,(COLUM1)
A0BC 47 978 LD B,A
A0BD CDD5A0 988 SALL: CALL SCRON
A0C8 2B 998 DEC HL
A0C1 18FA 1008 JR NZ,SAL
A0C3 E1 1018 POP HL
A0C4 110000 1028 LD DE,2048
A0C7 19 1038 ADD HL,DE
A0C8 8D 1048 DEC C
A0C9 79 1058 LD A,C
A0CA 28E8 1068 JR NZ,SALL
A0CC E1 1078 POP HL
A0CD 115000 1088 LD DE,80
A0D0 19 1098 ADD HL,DE
A0D1 C1 1108 POP BC
A0D2 18DC 1118 DJNZ SAL2
A0D4 C9 1128 RET
A0D5 7E 1138 SCRON: LD A,(HL)
A0D6 E480 1148 AND 136
A0D8 1F 1158 RRA
A0D9 1F 1168 RRA
A0DA 1F 1178 RRA
A0DB 57 1188 LD D,A
A0DC 7E 1198 LD A,(HL)
A0DD E477 1208 AND 119
A0DF 17 1218 RLA
A0E0 F600 1228 ALMAC: OR 8
A0E2 77 1238 LD (HL),A
A0E3 7A 1248 LD A,D
A0E4 32E1A0 1258 LD (ALMAC+1),A
A0E7 C9 1268 RET
A0E8 EB 1278 CALCU: EX DE,HL
A0E9 2180BF 1288 LD HL,MC000-88
A0EC 42 1298 LD B,D
A0ED 1600 1308 LD D,0
A0EF 1D 1318 DEC E
A0F0 19 1328 ADD HL,DE
A0F1 115000 1338 LD DE,80
A0F4 19 1348 S_BUC: ADD HL,DE
A0F5 18FD 1358 DJNZ S_BUC
A0F7 C9 1368 RET
A0F8 FE84 1378 SCROLL: CP 4
A0FA C267A2 1388 JP NZ,BADCOM
A0FD CD24A2 1398 CALL BUGS
A100 D8 1408 RET C
A101 D07E00 1418 LD A,(IX+0)
A104 321CA2 1428 LD (COLUM2),A
A107 8400 1438 LD B,0
A109 4F 1448 LD C,A
A10A ED431EA2 1458 LD (PASO1),BC
A10E D07E82 1468 LD A,(IX+2)
A111 321DA2 1478 LD (FIL2),A
A114 D06E84 1488 LD L,(IX+4)
A117 D06686 1498 LD H,(IX+6)
A11A CDE800 1508 CALL CALCU
A11D E5 1518 PUSH HL
A11E 110000 1528 LD DE,2048
A121 19 1538 ADD HL,DE
A122 D1 1548 POP DE
A123 3A1DA2 1558 LD A,(FIL2)
A126 47 1568 LD B,A
A127 C5 1578 DUC1: PUSH BC
A128 E5 1588 PUSH HL
A129 05 1598 PUSH DE
A12A E5 1608 PUSH HL
A12B 8687 1618 LD B,7
A12D C5 1628 DUC: PUSH BC
A12E E5 1638 PUSH HL
A12F 05 1648 PUSH DE
A130 ED4B1EA2 1658 LD BC,(PASO1)
A134 ED88 1668 LDIR
A136 E1 1678 POP HL
A137 110000 1688 LD DE,2048
A13A 19 1698 ADD HL,DE
A13B EB 1708 EX DE,HL
A13C E1 1718 POP HL
A13D 05 1728 PUSH DE
A13E 110000 1738 LD DE,2048
A141 19 1748 ADD HL,DE
A142 D1 1758 POP DE
A143 C1 1768 POP BC
A144 18E7 1778 DJNZ DUC
A146 E1 1788 POP HL
A147 05 1798 PUSH DE
A148 110000 1808 LD DE,1968
A14B 37 1818 SCF
A14C 3F 1828 CCF
A14D ED52 1838 SBC HL,DE
A14F D1 1848 POP DE
A150 ED4B1EA2 1858 LD BC,(PASO1)
A154 ED88 1868 LDIR
A156 E1 1878 POP HL
A157 115000 1888 LD DE,80
A15A 19 1898 ADD HL,DE
A15B EB 1908 EX DE,HL
A15C E1 1918 POP HL
A15D 05 1928 PUSH DE
A15E 115000 1938 LD DE,80
A161 19 1948 ADD HL,DE
A162 D1 1958 POP DE
A163 C1 1968 POP BC
A164 18C1 1978 DJNZ DUC1
A166 EB 1988 EX DE,HL
A167 115000 1998 LD DE,80
A16A 37 2008 SCF
A16B 3F 2018 CCF
A16C ED52 2028 SBC HL,DE
A16E 110038 2038 LD DE,2048*7

```

```

A171 19 2048 ADD HL,DE
A172 E5 2058 PUSH HL
A173 D1 2068 POP DE
A174 13 2078 INC DE
A175 ED4B1EA2 2088 LD BC,(PASO1)
A179 8B 2098 DEC BC
A17A 3600 2100 LD (HL),0
A17C ED88 2110 LDIR
A17E C9 2120 RET
A17F FE84 2130 SCROLL: CP 4
A181 C267A2 2140 JP NZ,BADCOM
A184 CD24A2 2150 CALL BUGS
A187 D8 2160 RET C
A188 D07E00 2170 LD A,(IX+0)
A18B 3222A2 2180 LD (COLUM3),A
A18E 8600 2190 LD B,0
A198 4F 2200 LD C,A
A191 ED4320A2 2210 LD (PASO2),BC
A195 D07E82 2220 LD A,(IX+2)
A198 67 2230 LD H,A
A199 3223A2 2240 LD (FIL3),A
A19C D06E84 2250 LD L,(IX+4)
A19F D07E86 2260 LD A,(IX+6)
A1A2 84 2270 ADD A,H
A1A3 3D 2280 DEC A
A1A4 67 2290 LD H,A
A1A5 CDE8A0 2300 CALL CALCU
A1A8 110038 2310 LD DE,2048*7
A1AB 19 2320 ADD HL,DE
A1AC E5 2330 PUSH HL
A1AD 110000 2340 LD DE,2048
A1B8 37 2350 SCF
A1B1 3F 2360 CCF
A1B2 ED52 2370 SBC HL,DE
A1B4 D1 2380 POP HL
A1B5 3A23A2 2390 LD A,(FIL3)
A1B8 47 2400 LD B,A
A1B9 C5 2410 MIS1: PUSH BC
A1BA E5 2420 PUSH HL
A1BB D5 2430 POP BC
A1BC E5 2440 PUSH HL
A1BD 8687 2450 LD B,7
A1BF C5 2460 MIS: PUSH BC
A1C0 E5 2470 PUSH HL
A1C1 D5 2480 PUSH DE
A1C2 ED4B20A2 2490 LD BC,(PASO2)
A1C6 ED88 2500 LDIR
A1C8 E1 2510 POP HL
A1C9 110000 2520 LD DE,2048
A1CC 37 2530 SCF
A1CD 3F 2540 CCF
A1CE ED52 2550 SBC HL,DE
A1D0 EB 2560 EX DE,HL
A1D1 E1 2570 POP HL
A1D2 D5 2580 PUSH DE
A1D3 110000 2590 LD DE,2048
A1D6 37 2600 SCF
A1D7 3F 2610 CCF
A1D8 ED52 2620 SBC HL,DE
A1DA D1 2630 POP HL
A1DB C1 2640 POP BC
A1DC 18E1 2650 DJNZ MIS
A1DE E1 2660 POP HL
A1DF D5 2670 PUSH DE
A1E8 110007 2680 LD DE,1968
A1E3 37 2690 SCF
A1E4 3F 2700 CCF
A1E5 19 2710 ADD HL,DE
A1E6 D1 2720 POP DE
A1E7 ED4B20A2 2730 LD BC,(PASO2)
A1EB ED88 2740 LDIR
A1ED E1 2750 POP HL
A1EE 115000 2760 LD DE,80
A1F1 ED52 2770 SBC HL,DE
A1F3 EB 2780 EX DE,HL
A1F4 E1 2790 POP HL
A1F5 05 2800 PUSH DE
A1F6 115000 2810 LD DE,80
A1F9 ED52 2820 SBC HL,DE
A1FB D1 2830 POP HL
A1FC C1 2840 POP BC
A1FD 180A 2850 DJNZ MIS
A1FF EB 2860 EX DE,HL
A200 115000 2870 LD DE,80
A203 37 2880 SCF
A204 3F 2890 CCF
A205 19 2900 ADD HL,DE
A206 110038 2910 LD DE,2048*7
A209 ED52 2920 SBC HL,DE
A20B E5 2930 PUSH HL
A20C D1 2940 POP HL
A20D 13 2950 INC DE
A20E ED4B20A2 2960 LD BC,(PASO2)
A212 0B 2970 DEC BC
A213 3600 2980 LD (HL),0
A215 ED88 2990 LDIR
A217 C9 3000 RET
A218 3818 FILAS: DEFS 1
A219 3028 COLUM: DEFS 1
A21A 3038 FILAS1: DEFS 1
A21B 3048 COLUM2: DEFS 1
A21C 3058 COLUM3: DEFS 1
A21D 3068 FILA2: DEFS 1
A21E 3078 PASO1: DEFS 2
A21F 3088 PASO2: DEFS 2
A222 3098 COLUM3: DEFS 1
A223 3108 FILA3: DEFS 1
A224 D07E00 3118 BUGS: LD A,(IX+0)
A227 A7 3120 AND A
A228 2835 3130 JR 2,PARAM
A22A D07E82 3140 LD A,(IX+2)
A22D A7 3150 AND A
A22E 282F 3160 JR 2,PARAM
A230 D07E84 3170 LD A,(IX+4)
A233 A7 3180 AND A
A234 2827 3190 JR 2,PARAM
A236 D07E86 3200 LD A,(IX+6)
A239 A7 3210 AND A
A23A 2823 3220 JR 2,PARAM
A23C D07E90 3230 LD A,(IX+8)
A23F 57 3240 LD D,A
A240 D07E84 3250 LD A,(IX+4)
A243 82 3260 ADD A,D
A244 FE50 3270 CP 0
A246 380F 3280 JR NC,RANGO
A248 D07E82 3290 LD A,(IX+2)
A24B 57 3300 LD D,A
A24C D07E86 3310 LD A,(IX+6)
A24F 82 3320 ADD A,D
A250 FE19 3330 CP 25
A252 3803 3340 JR NC,RANGO
A254 37 3350 SCF
A255 3F 3360 CCF

```

PROGRAMA CARGADOR

```

10 FOR N=&A000 TO &A2AE
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<75656 THEN PRINT "ERROR
EN DATAS"
60 DATA 1,9,160,33,44,160,195
70 DATA 209,188,23,160,195,48,160
80 DATA 195,138,160,195,248,160,195
90 DATA 127,161,82,79,76,76,196
100 DATA 82,79,76,76,201,82,79
110 DATA 76,76,213,82,79,76,76
120 DATA 193,0,0,0,0,254
130 DATA 4,194,103,162,205,36,162
140 DATA 216,221,126,0,50,25,162
150 DATA 221,126,2,50,24,162,221
160 DATA 110,4,221,102,6,205,232
170 DATA 160,58,24,162,71,197,229
180 DATA 14,8,229,175,50,131,160
190 DATA 58,25,162,71,205,119,160
200 DATA 35,16,250,225,17,0,8
210 DATA 25,13,121,32,232,225,17
220 DATA 80,0,25,193,16,220,201
230 DATA 126,230,17,23,23,23,87
240 DATA 126,230,238,31,246,0,119
250 DATA 122,50,131,160,201,254,4
260 DATA 194,103,162,205,36,162,216
270 DATA 221,126,0,111,50,27,162
280 DATA 221,126,2,50,26,162,221
290 DATA 126,4,133,61,111,221,102
300 DATA 6,205,232,160,58,26,162
310 DATA 71,197,229,14,8,229,175
320 DATA 58,225,160,58,27,162,71
330 DATA 205,213,160,43,16,250,225
340 DATA 17,0,8,25,13,121,32
350 DATA 232,225,17,0,0,25,193
360 DATA 16,220,201,126,230,136,31
370 DATA 31,31,87,126,230,119,23
380 DATA 246,0,119,122,50,225,160
390 DATA 201,235,33,176,191,66,22
400 DATA 0,29,25,17,80,0,25
410 DATA 16,253,201,254,4,194,103
420 DATA 162,205,36,162,216,221,126
430 DATA 0,50,28,162,6,0,79
440 DATA 237,67,30,162,221,126,2
450 DATA 50,29,162,221,110,4,221
460 DATA 102,6,205,232,160,229,17
470 DATA 0,8,25,209,58,29,162
480 DATA 71,197,229,213,229,6,7
490 DATA 197,229,213,237,75,30,162
500 DATA 237,176,225,17,0,8,25
510 DATA 235,225,213,17,0,8,25
520 DATA 209,193,16,231,225,213,17
530 DATA 176,7,55,63,237,82,209
540 DATA 237,75,30,162,237,176,225
550 DATA 17,80,0,25,235,225,213
560 DATA 17,80,0,25,209,193,16
570 DATA 193,235,17,80,0,55,63
580 DATA 237,82,17,0,56,25,229
590 DATA 209,19,237,75,30,162,11

```

600 DATA 54,8,237,176,201,254,4
 610 DATA 194,103,162,205,36,162,216
 620 DATA 221,126,0,50,34,162,6
 630 DATA 0,79,237,67,32,162,221
 640 DATA 126,2,103,50,35,162,221
 650 DATA 110,4,221,126,6,132,61
 660 DATA 103,205,232,160,17,0,56
 670 DATA 25,229,17,0,8,55,63
 680 DATA 237,82,209,58,35,162,71
 690 DATA 197,229,213,229,6,7,197
 700 DATA 229,213,237,75,32,162,237
 710 DATA 176,225,17,0,8,55,63
 720 DATA 237,82,235,225,213,17,0
 730 DATA 8,55,63,237,82,209,193
 740 DATA 16,225,225,213,17,176,7
 750 DATA 55,63,25,209,237,75,32
 760 DATA 162,237,176,225,17,80,0
 770 DATA 237,82,235,225,213,17,80
 780 DATA 0,237,82,209,193,16,186
 790 DATA 235,17,80,0,55,63,25
 800 DATA 17,0,56,237,82,229,209
 810 DATA 19,237,75,32,162,11,54
 820 DATA 0,237,176,201,0,0,0
 830 DATA 0,0,0,0,0,0,0
 840 DATA 0,0,221,126,0,167,40
 850 DATA 53,221,126,2,167,40,47
 860 DATA 221,126,4,167,40,41,221
 870 DATA 126,6,167,40,35,221,126
 880 DATA 0,87,221,126,4,130,254
 890 DATA 80,48,15,221,126,2,87
 900 DATA 221,126,6,130,254,25,48
 910 DATA 3,55,63,201,33,139,162
 920 DATA 205,111,162,55,201,33,154
 930 DATA 162,205,111,162,55,201,33
 940 DATA 121,162,205,111,162,55,201
 950 DATA 126,254,255,200,205,90,187
 960 DATA 35,24,246,70,65,76,84
 970 DATA 65,78,32,80,65,82,65
 980 DATA 77,69,84,82,79,83,255
 990 DATA 70,85,69,82,65,32,68
 1000 DATA 69,32,82,65,78,71,79
 1010 DATA 255,69,82,82,79,82,32
 1020 DATA 69,78,32,80,65,82,65
 1030 DATA 77,69,84,82,79,83,255
 1040 DATA 0

PROGRAMA EJEMP.

```
10 MODE 1:BORDER 0:INK 0,0:PAPER 0:
  INK 1,24:PEN 1
20 LOAD "SCROLL", &A000
30 CALL &A000
40 LOCATE 15,10:PRINT "MICROHOBBY"
50 LOCATE 16,12:PRINT "AMSTRAD"
60 GOSUB 300
70 INK 1,24
80 GOSUB 200
90 GOSUB 250
100 GOSUB 200
110 GOSUB 300
120 INK 1,6
130 GOSUB 200
140 GOSUB 250
150 GOSUB 200
160 GOSUB 300
170 INK 1,6
180 WHILE INKEY="" :WEND
190 END
200 FOR N=1 TO 90
210 :ROLLD,10,1,1,72
220 :ROLLI,12,1,1,72
230 NEXT
240 RETURN
250 FOR N=1 TO 16
260 :ROLLA,10,50,3,28
270 :ROLLU,10,5,3,28
280 NEXT
290 RETURN
300 FOR N=0 TO 26:INK 1,N:FOR X=1 TO
  20:NEXT X:NEXT N
310 RETURN
```

Sin duda alguna

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

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

SIN DUDA ALGUNA está abierta a todos.

AMSTRAD BILINGÜE

Estoy a punto de comprarme un **Amstrad** pero tengo la duda de cuál de las versiones del **Amstrad CPC 6128** comprar, la inglesa o la española. Y os formulo estas dos preguntas:

—Las ampliaciones de memoria, lápiz óptico, modem, tabletas gráficas, lector de tarjetas EPROM y periféricos en general, de procedencia todos ellos de Gran Bretaña, ¿son compatibles con las versiones españolas de los ordenadores **Amstrad**?

—En la versión inglesa, ¿es posible generar los siguientes caracteres del castellano y catalán (Ñ, ñ, ç, ¿, ü, à)?
José Fco. García (Barcelona)

Sobre tu pregunta te podemos decir que, las diferencias que hay entre el 6128 español e inglés son únicamente de teclado, los periféricos, modem, etc., son perfectamente compatibles, lo que ocurre es que al ser el teclado diferente es posible que para hacer funcionar correctamente un programa con el 6128 español haya que definir determinadas teclas, cosa ésta totalmente posible.

Osea, que la elección es tuya, aunque ten en cuenta que en España, teóricamente, ya no se debería vender la versión inglesa.

ACERCA DEL 472

Soy un poseedor de un 472 y me gustaría saber varias cuestiones:

1) ¿Qué significa (¿el comando?) «» (micro **AMSTRAD** n.º 31, el ratón casero, línea 3140)?

2) ¿Qué tecla debo utilizar en un 472, o por qué la puedo sustituir?

3) ¿Cuántas «K» se come un floppy disk y una impresora (por ejemplo, una Printer 130), y qué tipo de ampliación de memoria debo utilizar?

Constantino Roda

El símbolo al que te refieres no es sino el de exponenciación y está en el teclado pero así: «↑». Lo que ocurre es que en los listados sale diferente porque la impresora carece de ese símbolo.

El manejo de una unidad de disco implica la necesidad de utilizar un poco más de 1K (1.284 bytes) de memoria. En lo concerniente a la ampliación de memoria, lamentamos comunicarte que el 472 no puede utilizarlas.

¿CUANTA MEMORIA?

1.—Me gustaría saber si existen ampliaciones de memoria RAM para el CPC—464, y si así es, de cuántos Kb y el precio de la misma. Idem para la memoria ROM.

2.—¿Se pueden acoplar otras unidades de disco al CPC—464 además de la de 3?

Por último, darles la enhorabuena por la gran calidad de los programas publicados últimamente en la revista y rogarles que sigan publicando más programas de utilidad, como por ejemplo un programa de Hoja de Cálculo sencillo.

Agradeciéndoles de antemano la atención que presten a mis dudas.

Angel Niño González (León)

Por supuesto que existen ampliaciones de memoria RAM para el CPC—464, lo que ya no tenemos tan claro, aunque creemos que sí, es que haya ampliaciones de memoria ROM. La ampliación de memoria RAM es de 64K, en cuanto al precio, no te podemos decir una cifra concreta, pero está alrededor de 15.000 ptas.

Referente a tu segunda pregunta, la respuesta es sí, pudiendo ser de 3 1/2" y de 5 1/4" las posibles unidades a acoplar.

TASWORD 128

Podemos escribir a lápiz, a bolígrafo, e incluso a máquina, pero la forma más eficiente y fácil de realizar un escrito perfecto, es utilizar el ordenador.

G

racias a los programas dedicados a la escritura, podemos hacer cosas tan asombrosas que cualquier otro método queda totalmente eclipsado por las ventajas del ordenador.

Los mayores problemas de la escritura tradicional tienen su origen en que lo que está escrito ya no se puede rectificar, que forma que es imposible hacer ningún cambio si no nos gustan los resultados.

La mayor rectificación que se puede realizar en una máquina de escribir electrónica es borrar una o varias letras, lo cual permite enmendar algún error, pero no modificar sensiblemente el texto.

Con la llegada de la escritura informatizada, todo ha cambiado radicalmente. Gracias a los procesadores de texto, nuestro ordenador trabaja con las palabras como si de piezas independientes se tratara, ofreciendo una serie de posibilidades de alterar su disposición y estructura que parecen increíbles.

Esta facilidad de mover el texto por pantalla, de alterar las líneas, párrafos y páginas como nos venga en gana, es la responsable de la absoluta supremacía del procesador de textos, sobre cualquier otro medio de escritura.

Hasta los grandes fabricantes de máquinas de escribir reconocen las ventajas de la escritura electrónica, y dotan a sus modelos más evolucionados de pantalla, memoria y discos de almacenamiento.

¿Qué podemos hacer con un procesador?

Supongamos que tenemos que realizar un escrito, donde sabemos de qué vamos a hablar, pero no las palabras exactas.

Una vez superado el trauma del papel en blanco, las frases afluyen a nuestra mente con una velocidad que es difícil atrapar en el teclado.

Nuestros dedos saltan de tecla en tecla, con un ritmo desenfadado que ni nosotros mismos sabemos cuándo va a acabar.

Cuando queremos darnos cuenta, tenemos en la pantalla varias líneas de texto, conteniendo los primeros acordes de nuestra incipiente obra literaria.

Tras el párrafo de introducción, nuevas líneas van imprimiéndose en la pantalla, nuestro cerebro no deja de trabajar y las ideas se agolpan antes de ser traducidas a letras.

Poco a poco, nuestros dedos intentan descongestionar el caudal cerebral que nos desborda. Tras unos minutos de tecleo, la pantalla tiene un aspecto totalmente distinto, en nuestro frenesí creador hemos escrito casi dos pantallas de texto.

En ellas están contenidas las ideas, y los principales puntos sobre los que versará nuestro escrito.

Después de una reposada lectura, nos damos cuenta de que algunos puntos bien merecen un desarrollo más extenso, incluso ahora hemos recordado una idea que no puede ser ignorada.

Si estuviésemos escribiendo en papel, el próximo paso sería coger otro par de folios y comenzar a escribir basándonos en las dos primeras hojas, ya inservibles.

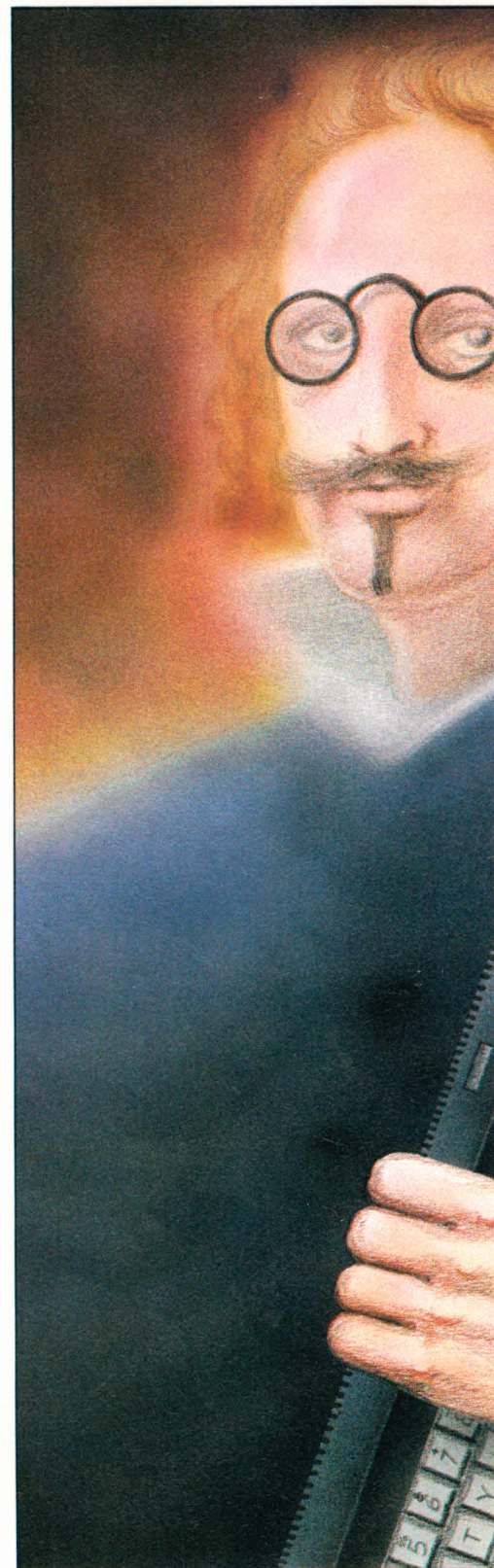
Gracias al procesador, para añadir esa nueva idea, en el sitio deseado, sólo tenemos que abrir espacio entre las líneas primitivas y comenzar a escribir.

Para desarrollar los puntos en los que nos habíamos quedado cortos, el mismo proceso nos permite darles la longitud necesaria.

Una vez que tenemos desarrolladas todas las ideas, y que les hemos dado la longitud deseada, es la hora de volver a leer. La nueva lectura, nos revela que precisamente el orden en el que hemos expuesto nuestro parecer no es el adecuado.

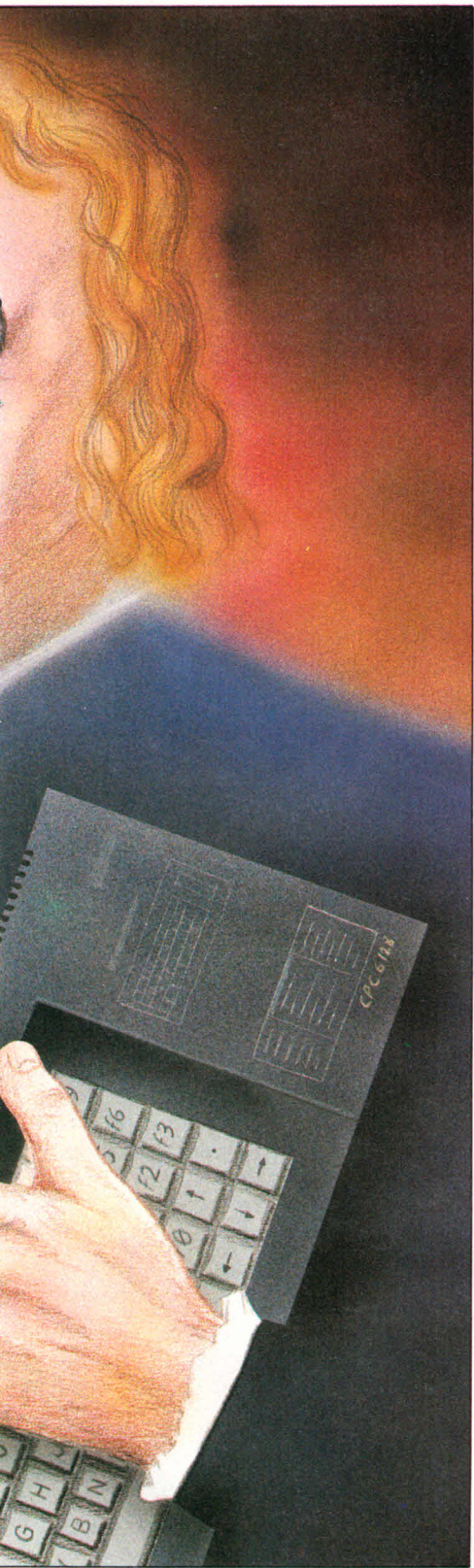
Nuevamente el procesador nos resuelve la papeleta cambiando al lugar idóneo las porciones de texto que haga falta.

Efectuados los cambios, leemos cuidadosamente para detectar esas comas que siempre se olvidan, esas palabras de ortografía errónea, y demás despistes que la precipitación y el tecleo hayan dejado en el escrito.



Borramos lo que haga falta, cambiamos las palabras necesarias, introducimos los puntos y comas requeridos, y corregimos la ortografía.

Una nueva lectura nos confirmará que nuestro texto está listo para imprimir, un toque de tecla nos dará un escrito ordenado en la exposición de las ideas, con una pulcra ejecución, perfecta ortografía y ni una sola enmienda o tachadura.



¿Cuántas veces tendríamos que haberlo repetido a máquina?

Tasword en castellano

El principal problema del software, es que como se produce en Inglaterra, lógicamente está hecho en inglés, produciendo graves pro-

blemas para la gente que no conoce este idioma.

Para evitar estos problemas y con el objeto de vender más copias, las casas que distribuyen el producto en España, traducen los manuales, consiguiendo de esta forma una solución intermedia; manuales en castellano e instrucciones en pantalla en inglés.

Asombrosamente, en la versión de tasword para España, llegamos a la solución total; estamos en posesión de un extenso manual de instrucciones en español, y lo que es más importante, de un programa en el que todos los mensajes, menús y demás instrucciones en pantalla están en castellano.

Otro gran problema de los procesadores de texto es el molesto caso de la ñ; la ortografía hispana posee esta artística letra única en el mundo, la cual por supuesto no está incluida en los códigos ASCII.

En Tasword el problema está resuelto; tenemos a nuestra disposición los signos de la ñ; la interrogación al principio de frase y las vocales acentuadas; con estas mejoras el programa queda totalmente adaptado a nuestra lengua.

Sólo hemos de reseñar un pequeño fallo entre tanta maravilla, con la inclusión de la ñ y demás signos típicos del castellano: ha desaparecido del teclado el signo de los dos puntos (en la versión inglesa del 6128, no en la castellana), nadie es perfecto.

Instrucciones y manual

El programa viene contenido en un estuche de grandes dimensiones. Cuando extraemos la carpeta nos encontramos con un extenso manual de 71 páginas, en el cual se encuentran ampliamente explicadas todas las funciones y posibilidades del mismo.

Los entusiastas de la lectura pasarán gratas horas de práctica, siguiendo el manual y consiguiendo en pocas sesiones un dominio total del mismo.

Los que no están tan inclinados a la lectura, tienen una posibilidad que les permitirá manejar el mismo sin usar apenas el manual.

En el disco del programa, vienen incluidos un tutor y varios textos explicativos. Estos se cargan como si fuese un texto cualquiera, y su lectura y la realización de los pequeños ejemplos que nos proponen, nos permite hacernos con el completo dominio del programa, sin tener que leer el manual.

Sin más preámbulos, extraemos el disco de la funda y lo insertamos en el ordenador; **RUN «TASWORD»** hace que éste se cargue en memoria en varios segundos.

Concluida la carga aparece ante nuestros ojos la página, a cuyo pie se encuentran los datos de línea, columna, justificación derecha, palabra partida, fin de página, tecla de ayuda y juego de caracteres, encima de los cua-

Banco de PRUEBAS

les tenemos la reglilla de márgenes y tabuladores.

Personalización del programa

La primera operación que conviene realizar es personalizar el programa; con ella modificaremos los colores de la pantalla y definiremos los márgenes adecuados al formato de nuestro texto.

El cambio de color de la pantalla es un hecho que agradecerán los poseedores de monitores en color; los colores iniciales son papel negro con letras blancas, los cuales producen un gran contraste para la vista, al ser opuestos a los textos escritos sobre papel.

Los márgenes los definiremos de manera que las líneas de texto tengan el número de caracteres deseado; inicialmente el número de caracteres fijado por línea es de 80, pudiendo establecerse éste entre 1 y 128 (la pantalla admite un máximo de 80, por lo cual sobrepasar este número nos obligará a realizar molestos scrolls).

El menú principal, en su opción de personalizar el programa nos permite realizar este cambio tan necesario: en él podemos elegir color del papel, borde 1, borde 2, tinta 2, tinta 1, tinta 2, forma del cursor, margen derecho y margen izquierdo.

En la elección de colores, para evitar el contraste (*fuertemente perjudicial para la vista*) que existe entre pasar la vista continuamente desde un papel blanco con impresión negra, a una pantalla negra con letra blanca, se puede usar el color 13 (blanco) para el papel y 0 (negro) para la tinta 1, la tinta 2 puede ser el 3 (roja), pues sólo aparece en el menú principal indicando el número de líneas, palabras, caracteres y demás datos estadísticos del texto escrito.

Una vez realizados los cambios necesarios en el programa procedemos a su grabación, para poder cargarlo en memoria sin efectuar más modificaciones cada vez que vayamos a usarlo.

El menú principal posee una opción de grabación que resulta idónea para este fin. Con ella además podremos obtener copias de seguridad del programa, con las cuales podemos prevenir cualquier accidente que se produzca en el master.

Funciones de pie de página

La pantalla inicial está presidida por la cabecera de ayuda, la cual ocupa casi un tercio de pantalla; puede eliminarse teniendo de esta forma la pantalla completa para texto.

En la parte inferior de la pantalla, bajo la regilla de tabulación y ancho de líneas, se encuentran las opciones básicas, y el indicador de si cada uno de ellas se encuentra activada o no.

A la izquierda se nos marca el número de línea en la que nos encontramos; tras ella, el número de columnas; después las cuatro funciones de justificación, ruptura de palabras, modo de inserción y final de página.

Ahora procedemos a fijar las opciones que más se adaptan a nuestro estilo de texto y forma de escribir.

En primer lugar fijamos la opción de ruptura de palabras; con ella activada el programa comprueba si la última palabra de la línea cabe íntegra en ella, en caso contrario la pasa a la de abajo sin que ésta quede partida.

Con la ruptura desactivada, la última palabra de la línea quedaría partida en dos, en caso de que no entrase íntegra en la línea.

La siguiente opción es la de justificación; ésta nos permite ordenar los espacios entre las palabras que componen una línea, de forma que en la primera y última columnas de la misma siempre se encuentre alguna letra.

La utilización de la justificación da a nuestros textos un aspecto francamente uniforme y agradable a la vista. La distribución de las palabras en la línea ocupando toda ésta separadas a espacios iguales, evita esos espacios en blanco a la derecha de la misma, que por decirlo de alguna forma descompensan un poco el ritmo de las letras.

Esta opción está íntimamente ligada a la de ruptura de palabras, pues sirve de complemento indispensable cuando no queremos que aparezcan palabras partidas.

La siguiente función, es una de las más útiles e importantes en un procesador de textos: estamos hablando de la inserción.

Debido a que nadie es perfecto y que el ser humano se equivoca con frecuencia, es fundamental que entre ciertas palabras podamos intercalar aquello que se nos haya olvidado.

Supongamos que tras una palabra queremos poner una coma, que otra queramos dejarla en plural en vez de en singular, que nos falta una palabra entre otras dos, etc.

Todo esto se resuelve de forma inmediata con la inserción, con ella podemos partir una línea por el punto deseado y a partir de éste escribir, volviendo a ensamblar la línea después con la justificación.

Nuestro procesador permite dos modos distintos de justificación: la normal que mediante un toque de ENTER crea una línea en blanco en la posición del cursor, y la automática.

Justificación automática

Sin duda, la operación más importante de esta versión de Tasword es la justificación automática. Con ella podemos intercalar las palabras que deseemos en el texto con sólo colocar el cursor en posición y empezar a escribir.

Su uso ofrece la ventaja de que ordena el texto que viene a continuación, moviendo hacia abajo las palabras que sobran de la línea, es decir, que además justifica automáticamente el texto restante.

La comodidad que ofrece este modo de inserción y la facilidad con que ésta operación se realiza, hacen que esta versión de Tasword, deje a las anteriores bastante anticuadas.

La última opción es la del indicador de fin de página. Una vez definidas las líneas que constituyen cada página, esta opción activada hace aparecer una línea de puntos al final de cada una, con lo cual obtenemos una importante guía visual de la estructura de nuestro escrito.

Fijadas las funciones básicas, sólo nos queda empezar a escribir, en pantalla irán apareciendo las letras que tecleamos, ordenándose en las líneas según las funciones que tengamos activadas.

Movimiento del cursor

Con una cantidad de texto suficientemente grande, podemos observar las ventajas, los distintos tipos de movimiento del cursor a lo largo de las líneas y palabras.

Este puede saltar de letra a letra, de palabra a palabra, de línea a línea, de una pantalla a la siguiente, al principio y al final del texto.



Los errores cometidos en nuestra operación de tecleo, son fáciles de subsanar con la tecla de borrado: podemos borrar una letra, una palabra, una línea, e incluso si el error es de demasiada cuantía, podemos borrar todo el texto.

Si no tenemos activado el modo automático de inserción, podemos insertar una letra, una línea, o varias donde nos interese.

Otras posibilidades en la escritura de líneas, son la de mover el texto a la derecha, a la izquierda, centrar una línea, justificar un párrafo entero, o solamente una línea.

Podemos adaptar el formato de las líneas a nuestras necesidades alternando los márgenes.





Búsqueda de palabras

Para los escritores noveles que, sin darse cuenta, repiten a lo largo de un texto cientos de veces la misma palabra, Tasword ofrece una solución que sacará de apuros a más de uno, haciendo que su estilo literario mejore de una forma ostensible.

Con la búsqueda de palabras podemos localizar a lo largo del texto todas las veces que hemos utilizado determinada palabra, y sustituirla según convenga o no por ser la elegida a este fin.

Para el manejo del disco, el menú principal nos ofrece la posibilidad de cargar, grabar o mezclar cualquier tipo de textos que tengamos en el disco.

Cada vez que llegamos al menú, Tasword nos da la información más relevante del texto que estamos procesando: en ella nos especifica el número de líneas, el de palabras, el número de caracteres y la cantidad de memoria disponible.

Para pasar al papel el contenido de la pantalla, este programa ofrece gran variedad de posibilidades.

Papel continuo, hojas sueltas, doble espacio, triple número de líneas por página, pie de página, cabecera, imprimir número de página, etc. consiguiendo que nuestro trabajo sea totalmente perfecto.

El resultado final

Tasman software, en su última versión de tasword, ha conseguido un producto francamente notable, en el que además de aprovechar el total de la memoria del 128, ha mejorado ampliamente sus anteriores procesadores.

La inclusión de la inserción automática, que en los anteriores programas se echaba en falta, dota de una inusitada potencia al programa.

La adaptación a la ortografía del castellano con eles vocales acentuadas, son un detalle que nunca dejaremos de agradecer.

La posibilidad de recuperar la última línea borrada, constituye otra innovación respecto a las anteriores versiones.



nes, lo cuales se pueden suprimir, colocar al lado derecho, o situar al izquierdo.

De forma similar podemos alterar las marcas de tabulación, de manera que el movimiento rápido a lo largo de las líneas se realice con soltura. Podemos borrar todos los existentes, restituir los primitivos, añadir los que deseemos o borrar los que nos sobren.

Página de ayuda

Un capítulo importante, es la página de ayuda; en ella se encuentran concentradas todas las funciones y las teclas que las activan. Para sacar esta página por pantalla, basta con pulsar una tecla (ESC); después, un toque de ENTER nos devuelve a la misma posición del texto en la que nos encontrábamos.

Otra innovación respecto a anteriores versiones, introducida para el uso en España del programa, es la inclusión de la ñ en el procesador y las vocales acentuadas.

Para la repetición de algunas de esas palabras que siempre aparecen en cualquier escrito, o de las cabeceras de notas, cartas, etc, podemos programar las teclas de función para que contengan las palabras deseadas; después, a toque de tecla aparecerá en la posición del cursor el texto programado.

El movimiento de bloques de texto, es otra de las grandes ventajas que pone en manos del usuario un procesador de textos. Con éste podemos desplazar a la posición requerida, la porción de texto que hallamos seleccionado, como si de cortar y pegar se tratara. Podemos copiar el bloque, trasladarlo de un lugar a otro y borrar un trozo determinado.

Banco de PRUEBAS

El programa es de un manejo sencillo y fácil de aprender, no obstante, para sacarle el máximo partido, dominando la cantidad de posibilidades que ofrece, habremos de dedicarle varias sesiones, además de realizar una delicada lectura del manual.

Las versiones del Tasword

Tasword ha ido evolucionando en el tiempo de la misma forma que los modelos de **Amstrad**.

Desde la aparición del 464, hasta la llegada del 6128, el aumento de memoria y la inclusión del disco en el conjunto del ordenador, han hecho que el programa haya sufrido un proceso paralelo de perfeccionamiento y aumento de sus posibilidades, dando como resultado una versión que se adapta perfectamente a las características del 6128.

La creadora del tasword es la casa británica Tasman Software, la cual está especializada en hacer software para **Amstrad**. La versión que tenemos ante nosotros, es el resultado de un proceso de evolución y transformación del programa original, nacido en 1983.

La primera versión de Tasword estaba realizada para el 464, con una capacidad de texto equivalente a 15.800 caracteres, quedando el resto de la memoria ocupada por el programa, con ella no podían manejar más de 200 líneas de texto.

La segunda versión, realizada en 1984 pensando exclusivamente en la utilización de disco, en ésta se produce un aumento de la capacidad a 21.800 caracteres, sacando el menú principal de la memoria y cogiéndolo del disco, con lo cual podíamos tener hasta 350 líneas de texto en memoria...

La versión final; Tasword 128 está realizada para aprovechar las superiores prestaciones del 6128, consiguiendo una capacidad de 36.800 caracteres, e introduciendo nuevas opciones que le hacen infinitamente superior a las anteriores.

El uso del segundo banco de memoria es el principal responsable de la superior potencia y capacidad de este programa; con ello se ha conseguido un importante aumento de la cantidad de texto que se puede tener en memoria, pudiendo llegar hasta las 500 líneas.

Un proceso evolutivo que ha ido mejorando el programa de una forma continua y constante.

SOMOS MAYORISTAS

MICRO-1

PRECIOS INCLUIDO IVA

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 274 53 80

(Metro O'Donnell o Goya)
Aparcamiento gratuito en Felipe II

Ofertas en software: 2 programas al precio de 1 y además regalo fin de curso una calculadora completamente gratis. ¡¡Asombroso!!
¿Verdad?

BAT MAN _____	2.300	ptas.	KUNG-FU MASTER _____	2.300	ptas.
ROCK'N LUCHA _____	2.300	ptas.	SABOTEUR _____	2.300	ptas.
YIER AR KUNG FU _____	2.300	ptas.	PING PONG _____	2.300	ptas.
THE WAY OF THE TIGER _____	2.300	ptas.	MILLION II _____	0000	ptas.
WEST BANK _____	2.100	ptas.	OLE TORO _____	2.300	ptas.
CAMELOT WARRIORS _____	2.300	ptas.	TURBO ESPRIT _____	2.100	ptas.
RAMBO _____	2.300	ptas.	SABRE WULF _____	1.650	ptas.
WORLD CUP (DISCO) _____	3.300	ptas.	BATALLA DE LOS PLANETAS _____	2.100	ptas.
MILLION II (DISCO) _____	3.300	ptas.	SABOTEUR-COMBAT LINX DISCO _____	3.300	ptas.
RAMBO-MATCH DAY (DISCO) 3.300 ptas.					

SOFTWARE DE REGALO (OFERTA 2x1)
DECATHLON BEACH HEAD SOUTHERN BELLE
DRAGONTORC

LAPIZ OPTICO
3.295 PTAS.
CASSETTE ESPECIAL ORDENADOR
5.295 PTAS.

SINTETIZADOR DE VOZ EN
CASTELLANO
7.650 PTAS.
AMPLIACION DE MEMORIA ANTA 64 K.3
12.500 PTAS.

IMPRESORAS
20% DE DESCUENTO SOBRE P.V.P.

TAPA DE METACRILATO 464	895
CABLE CENTRONICS	3.175
CABLE SEGUNDA UNIDAD D.	1.790
CABLE SEPARADORES 6128	1.975
INTERFACE RS232	9.265
CINTA VIRGEN C15	69

CABLE AUDIO	795
CABLE ADAPTADOR 2 JOYSTICK	2.390
CABLE SEPARADORES 464	1.390
CABLE SEPARADOR 8256	2.900
CABLE RS232	2.500
DISKETTES 3''	990

PRECIOS EXCEPCIONALES PARA TU AMSTRAD
CPC-464, CPC-6128, PCW-8256

¡¡LLEGARON LAS REBAJAS DE VERANO A MICRO !!

OFERTAS EN JOYSTICKS	
QUICK SHOT I	1.395
QUICK SHOT II	1.695
QUICK SHOT V	1.695

PEDIDOS CONTRA REEMBOLSO SIN NINGUN GASTO DE ENVIO TEL. (91) 274 53 80
O ESCRIBIENDO A: MICRO-1. C/ DUQUE DE SESTO, 50. 28009 MADRID.

Tiendas y distribuidores grandes descuentos.
Dirigirse a Dipromsa. C/ Galatea, 25. Tel. (91) 742 20 19 ó 742 79 68

Un número de película
A partir del día 1 estreno en toda España
 Batman, Bomb Jack, Commando...
 todos tus héroes preferidos juntos en este número,
 para divertirse a tope.

MICRO
Manía
 Año 11 - N° 15 Sólo para adictos 300 Ptas.

Canarias, Ceuta y Melilla 285 ptas.



**BATMAN
 UN SUPERHEROE
 DE PELICULA**

HOBBY PRESS

SPECTRUM	ALIEN HIGHWAY	MSX	NIGHT SHADE
	SPINDIZZY		y además
AMSTRAD	FRANKIE GOES TO HOLLYWOOD	SPECTRUM MSX	Gunfight y Starquake
	BOMB JACK		AMSTRAD COMMODORE
	COMMANDO		

Utensilios y Cachivaches

UNA AUTENTICA ORQUESTA SINFONICA

HISTORIA DE SOFT *Aventura en el castillo del Conde Drácula*

Pídelo en tu kiosco

PUZZLEMIND

Puzzlemind es un programa muy poco corriente, y reúne todas las cualidades para merecer este apelativo: es muy original, muy bonito y muy adictivo, con varios niveles de destreza para que todos, desde el despistado al «memorión», tengan su rato de diversión.

Además, para los programadores lo digo, está escrito en un Basic muy paciente y laborioso, que merece la pena estudiar, tanto para aprender lo que se debe o no se debe hacer, según los gustos.

Desde luego, Puzzlemind es un programa raro, bastante difícil de resolver, y que puede ser la respuesta a esas tardes terribles de agosto, en pleno abandono estival, cuando todo el mundo menos tú, «El Gran Insomne», está entregado a filosóficas disquisiciones con la almohada. ¡Caray!, la siesta.



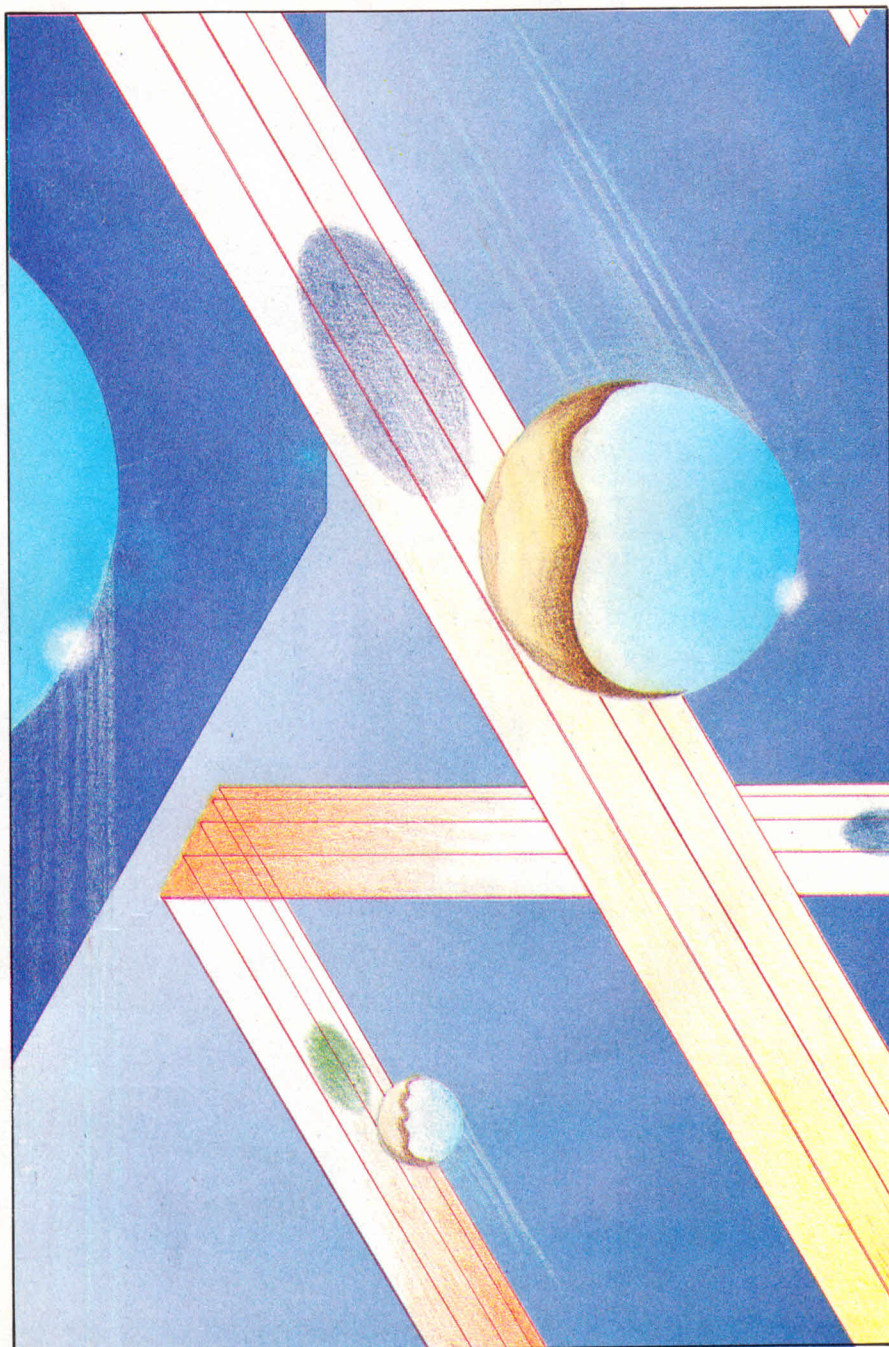
El programa que mando es un juego, concretamente un **PUZZLE**. Tiene tres niveles de dificultad y un número limitado de ayudas por nivel. Es un juego muy sencillo que pone a prueba la capacidad de «retentiva» del jugador y su habilidad para visualizar el conjunto de piezas y colocarlas en el lugar correcto, sabiendo distribuir a lo largo del juego el número de consultas al **PUZZLE** de ayudas para no quedarse colgado y poder completarlo con la máxima puntuación.

El propio programa se encarga de explicar su funcionamiento y la utilización del teclado en las instrucciones y en las bases.

El programa es compatible con todos los CPC. Lo he grabado por las dos caras por aquéllos de los «duendes». No me queda más que felicitaros por vuestra revista que en muchos casos me ha ayudado a comprender mejor el funcionamiento de mi **Amstrad**, muy aclarativa. Gracias.

Esperando que mi programa guste, se despide:

Marize Benayas Pazos



TABLAS Y VARIABLES

T	Tabla con el puzzle completo para mostrar las bases de cada nivel.
PG	Tabla con las coordenadas de las piezas de muestra y elección.
PP	Tabla con las coordenadas de las piezas que el jugador ha de colocar en el puzzle.
M	Tabla con el puzzle completo para las ayudas.
NP	Tabla donde se guardan los números de las piezas aleatoriamente.
HL	Número de ayudas según el nivel.
PNEG	Errores, puntos negativos.
PPOS	Aciertos, puntos positivos.
TPG	Total de elementos de la tabla PG.
TPP	Total de elementos de la tabla PP.
TT	Total de elementos de las tablas T y M.
PUNT	Contador de puntuación obtenida.
ACIE	Contador de aciertos.
NI	Elección de nivel de dificultad.

PROGRAMA

1-36	Portada (presentación).
37-74	Instrucciones y elección del nivel. Sentencias ON ERROR... y ON BREAK... Límites del temporizador EVERY.
75-148	Datos de los niveles y los diferentes puzzles.
149-189	Control y llenado de tablas.
190-196	Fin de la carga.
197-225	Inicializa y muestra las bases.
226-271	Definición de letras.
272-280	Definición de gráficos.
281-296	Impresión.
297-316	Cuadrícula de las piezas y los puzzles.
317-332	Pista inicial.
333-345	Inicio del juego.
346-374	PROGRAMA PRINCIPAL. Elección de las piezas o ayudas.
375-391	Rutina de ayudas.
392-394	Rutina de tratamiento de errores y ruptura del programa.
395-403	Temporizador EVERY. Fin del programa.
404-414	... has perdido, despedida.
415-425	... has ganado, felicidades.

Serie ORO

```

49 LOCATE 8,16:PEN 2:PRINT "[":PEN
1:PRINT "N":PEN 2:PRINT "]":LOCATE 1
5,16:PEN 1:PRINT "Si no quieres col
ocar":LOCATE 15,17:PRINT "esa pieza
y deseas":LOCATE 15,18:PRINT "qu
e te muestre otra."
50 LOCATE 8,20:PEN 2:PRINT "[":PEN
1:PRINT "A":PEN 2:PRINT "]":LOCATE 1
5,20:PEN 1:PRINT "Si necesitas ayuda
."
51 LOCATE 11,23:PEN 2:PRINT "[":PEN
1:PRINT "PRESIONA UNA TECLA":PEN 2
:PRINT "]"
52 WHILE INKEY$="" :FOR ap=1 TO 300:
INK 2,1:NEXT:FOR de=1 TO 300:INK 2,
11:NEXT:WEND
53 BORDER 0:INK 0,0:SYMBOL AFTER 32
:WINDOW 2,39,5,24:PAPER 0:CLS:WINDO
W 1,40,1,25
54 LOCATE 3,10:PEN 3:PRINT "*":PEN
1:PRINT "Tienes TRES niveles de dif
icultad:"
55 LOCATE 12,13:PEN 2:PRINT "[":PEN
1:PRINT "1":PEN 2:PRINT "] " :PEN
1:PRINT ".....FACIL."
56 LOCATE 12,15:PEN 2:PRINT "[":PEN
1:PRINT "2":PEN 2:PRINT "] " :PEN
1:PRINT "....DIFICIL."
57 LOCATE 12,17:PEN 2:PRINT "[":PEN
1:PRINT "3":PEN 2:PRINT "] " :PEN
1:PRINT ".MUY DIFICIL."
58 LOCATE 12,23:PEN 2:PRINT "[":PEN
1:PRINT "QUE NIVEL QUIERES":PEN 2
:PRINT "]"
59 a$=INKEY$:IF a$>"1" AND a$<"2"
AND a$<"3" THEN FOR ap=1 TO 100:I
F a$="1" AND a$="2" AND a$="3" THEN
60 ELSE INK 2,1:NEXT:FOR de=1 TO 1
00:IF a$="1" AND a$="2" AND a$="3"
THEN 60 ELSE INK 2,11:NEXT:GOTO 59
60 ni=VAL(a$)
61 MODE 1:BORDER 0
62 xiz=1:xde=40:yar=1:yab=25:col=0
63 WINDOW xiz,xde,yar,yab:PAPER col
:CLS
64 xiz=xiz+3:xde=xde-3:yar=yar+3:ya
b=yab-3:col=col+1
65 IF xiz=13 THEN WINDOW 12,29,12,1
4:PAPER 0:CLS:GOTO 66 ELSE 63
66 DATA "C","A","R","G","A","N","D"
,"O"," ","D","A","T","O","S"
67 RESTORE 66
68 LOCATE 3,2:FOR x=1 TO 14:READ b$
:PAPER 0:PEN 1:PRINT b$;:FOR t=1 TO
300:NEXT:NEXT
69
70 ON BREAK GOSUB 394
71 ON ERROR GOTO 393
72 EVERY 50,1 GOSUB 396
73 KEY 140,"mode 1:border 0:ink 1,2
5:ink 2,10:ink 3,15:paper 0:pen 1:l
ist"+CHR$(13)
74 KEY DEF 7,0,140
75 REM

```

- NIVEL 1 -

T-----

```

76 DATA 6,8,5,6,7,8,10,17,21,13,9,2
0,7,13,13,11,12,5,16,3,13,23,18,19,
3,18,21,10,21,3,6,10,13,21,6
77 REM

```

-PG---Coordenadas venta

```

nas-----
78 DATA 29,36,10,17,33,36,12,15,29,
36,10,17,29,30,12,15,31,34,16,17,35
,36,12,17,29,36,10,17,29,30,12,17,3
1,36,16,17,29,36,10,17,29,36,16,17
79 DATA 29,36,10,17,29,30,16,17,29,

```

```

1 P U Z Z L E -PORTADA-
2 MODE 1:BORDER 0:INK 0,0:INK 1,25:
INK 2,11:INK 3,6:SYMBOL AFTER 32:LO
CATE 7,12:PEN 1:PAPER 0:PRINT "[ QUI
ERES INSTRUCCIONES? ]"
3 a$=UPPER$(INKEY$):IF a$=" "AND a$<
">"S" AND a$<>"N"THEN 3
4 IF a$="S" THEN 21 ELSE 53
5 GOTO 3
6 DATA 478,80,426,30,379,70,478,30,
379,50,478,50,379,80,0,30
7 DATA 426,80,379,30,358,23,358,23,
379,30,426,30,358,80,0,30
8 DATA 379,80,358,30,319,70,379,30,
319,50,379,50,319,80,0,30
9 DATA 358,80,319,30,284,23,284,23,
319,30,358,30,284,80,0,30
10 DATA 319,70,478,23,426,23,379,23
,358,23,319,30,284,80,0,20
11 DATA 284,70,426,23,379,23,358,23
,319,23,284,30,253,80,0,20
12 DATA 253,70,379,23,358,23,319,23
,284,23,253,30,239,80,0,20
13 DATA 239,50,253,30,284,30,358,60
,239,30,319,60,239,100,0,30
14 RESTORE 6:ENT 1,10,15,14:ENV 1,4
,1,1
15 FOR v=0 TO 63
16 READ no,du
17 IF INKEY$<>" " THEN RELEASE 1:RET
URN
18 SOUND 1,no,du,11,1,12
19 NEXT
20 GOTO 14
21 MODE 1
22 LOCATE 11,23:PAPER 0:PEN 1:PRINT
"[ POR FAVOR ESPERA ]":FOR pas=1 TO
5:FOR ap=1 TO 400:INK 1,25:NEXT:FO
R de=1 TO 400:INK 1,0:NEXT:NEXT
23 INK 1,0:INK 2,0:INK 3,0
24 MODE 1:te$="PUZZLE":x=1:y=1:xx=0
:yy=240:a=16*(x-1):b=399-(y-1)*16
25 FOR l=1 TO LEN(te$):a$=MID$(te$,
l,1):GOSUB 26:xx=xx+110:NEXT l:GOTO
32
26 p=xx:q=yy:LOCATE x,y:PRINT a$
27 FOR n=b TO b-16 STEP -2:FOR m=a
TO a+16 STEP 2
28 IF TEST(m,n) THEN GOSUB 31
29 p=p+10
30 NEXT m:q=q-10:p=xx:NEXT n:RETURN

```

```

31 FOR t=0 TO 7:MOVE p+10,q-t:DRAWR
B,0,3:MOVE p,q-t+8:DRAWR B,0,INT(R
ND*2)+1:NEXT:RETURN
32 WINDOW 1,40,1,7:PAPER 2:CLS:WIND
OW 2,39,3,8:PAPER 1:CLS:WINDOW 1,40
,17,25:PAPER 3:CLS:WINDOW 2,39,16,2
4:PAPER 2:CLS
33 DATA P," ",R," ",E," ",S," ",I,"
",O," ",N," ",A," ",U," ",N," ",
A," ",T," ",E," ",C," ",L," ",A
34 INK 1,25:INK 2,18:INK 3,6:LOCATE
1,25:PEN 3:PRINT CHR$(164):" Mariz
e Benayas, 1986"
35 WINDOW 1,40,1,25:LOCATE 3,20:RES
TORE 33:FOR i=1 TO 31:READ t$:PEN 1
:PRINT t$:FOR pas=1 TO 50:NEXT:NEX
T
36 GOSUB 14
37

```

INSTRUCCIONES

```

38 MODE 1:BORDER 0:WINDOW 1,40,1,25
:PAPER 0:CLS:PAPER 0:PEN 3:INK 1,25
39 FOR b=2 TO 12:LOCATE b,1:PRINT C
HR$(197):NEXT
40 FOR b=2 TO 3:LOCATE 13,b:PRINT C
HR$(198):LOCATE 29,b:PRINT CHR$(196
):NEXT
41 FOR b=14 TO 28:LOCATE b,4:PRINT
CHR$(197):NEXT
42 FOR b=30 TO 39:LOCATE b,1:PRINT
CHR$(197):NEXT
43 FOR b=2 TO 24:LOCATE 40,b:PRINT
CHR$(198):LOCATE 1,b:PRINT CHR$(196
):NEXT
44 FOR b=2 TO 19:LOCATE b,25:PRINT
CHR$(199):NEXT
45 PAPER 0:PEN 1:LOCATE 15,2:PRINT "
INSTRUCCIONES"
46 LOCATE 4,7:PEN 3:PRINT "*":PEN 1
:PRINT "Te ire mostrando una a una
todas":LOCATE 6,8:PRINT "las piezas
del PUZZLE."
47 LOCATE 4,10:PEN 3:PRINT "*":PEN
1:PRINT "Tu las elegirás pulsando :
"
48 LOCATE 8,13:PEN 2:PRINT "[":PEN
1:PRINT "S":PEN 2:PRINT "]":PEN 1:LO
CATE 15,13:PRINT "Si quieres colocar
la":LOCATE 15,14:PRINT "pieza most
rada."

```

36, 10, 17, 29, 36, 10, 17, 31, 34, 10, 17, 35
 36, 10, 17, 29, 36, 10, 17, 29, 30, 10, 17, 3
 1, 36, 10, 17, 29, 36, 10, 17, 29, 34, 10, 17
 B0 DATA 33, 36, 14, 17, 29, 36, 10, 17, 29,
 30, 10, 13, 29, 32, 14, 17, 29, 36, 10, 17, 33
 , 36, 12, 17, 29, 36, 10, 17, 29, 34, 12, 17, 3
 1, 34, 10, 11, 35, 36, 10, 17, 29, 36, 10, 17
 B1 DATA 31, 36, 10, 13, 29, 30, 10, 17, 29,
 36, 10, 17, 29, 32, 10, 13, 33, 36, 10, 17, 29
 , 36, 10, 17, 29, 32, 10, 17, 29, 36, 10, 17, 3
 3, 36, 10, 17, 29, 36, 10, 17, 29, 34, 12, 17
 B2 DATA 35, 36, 10, 11, 29, 34, 10, 11, 29,
 36, 10, 17, 29, 30, 10, 11, 31, 36, 14, 17, 29
 , 36, 10, 17, 33, 36, 10, 17, 29, 32, 14, 17, 2
 9, 36, 10, 17, 29, 32, 10, 17, 33, 36, 14, 17
 B3 DATA 29, 36, 10, 17, 33, 36, 10, 13, 29,
 36, 10, 17, 29, 34, 10, 13, 29, 36, 10, 17, 29
 , 36, 10, 17, 33, 36, 10, 13, 29, 36, 10, 17, 2
 9, 32, 10, 13
 B4 REM

--PP--Coordenadas venta

nas-----
 B5 DATA 4, 7, 4, 7, 6, 7, 5, 6, 8, 11, 4, 7, 8,
 8, 5, 6, 9, 10, 7, 7, 11, 11, 5, 7, 12, 15, 4, 7,
 12, 12, 5, 7, 13, 15, 7, 7, 16, 19, 4, 7, 16, 19
 , 7, 7
 B6 DATA 20, 23, 4, 7, 20, 20, 7, 7, 4, 7, 8, 1
 1, 8, 11, 8, 11, 9, 11, 8, 11, 11, 11, 8, 11, 12
 , 15, 8, 11, 12, 12, 8, 11, 13, 15, 8, 11, 16, 1
 9, 8, 11, 16, 19, 8, 11

B7 DATA 18, 19, 10, 11, 20, 23, 8, 11, 20, 2
 0, 8, 9, 20, 21, 10, 11, 4, 7, 12, 15, 6, 7, 13,
 15, 8, 11, 12, 13, 8, 10, 13, 15, 9, 10, 12, 12
 , 11, 11, 12, 15, 12, 15, 12, 15
 B8 DATA 13, 15, 12, 13, 12, 12, 12, 15, 16,
 19, 12, 15, 16, 17, 12, 13, 18, 19, 12, 15, 20
 , 23, 12, 15, 20, 21, 12, 15, 4, 7, 16, 19, 6, 7
 , 16, 19, 8, 11, 16, 19, 8, 10, 17, 19
 B9 DATA 11, 11, 16, 16, 8, 10, 16, 16, 12, 1
 5, 16, 19, 12, 12, 16, 16, 13, 15, 18, 19, 16,
 19, 16, 19, 18, 19, 16, 19, 14, 17, 18, 19, 20
 , 23, 16, 19, 20, 21, 16, 19, 22, 23, 18, 19
 90 DATA 4, 7, 20, 23, 6, 7, 20, 21, 8, 11, 20
 , 23, 8, 10, 20, 21, 12, 15, 20, 23, 16, 19, 20
 , 23, 18, 19, 20, 21, 20, 23, 20, 23, 20, 21, 2
 0, 21
 91 REM

- M-----

92 DATA 30, 38, 11, 16, 13, 33, 40, 20, 20,
 3, 37, 39, 14, 22, 3, 31, 32, 10, 18, 3, 27, 30
 , 16, 22, 6, 29, 30, 19, 22, 13, 28, 29, 10, 10
 , 7
 93 REM

--PG y PP--numero de la

pieza--
 94 DATA 1, 1, 2, 2, 2, 2, 3, 3, 4, 4, 5, 5, 6
 , 7, 7, 8, 8, 8, 9, 9, 10, 10, 10, 11, 11, 1
 2, 12, 12, 12, 13, 13, 13, 14, 14, 14, 15, 15,
 16, 16, 17, 17, 17, 18, 18, 18, 19, 19, 19
 , 20, 20, 20, 21, 21, 22, 22, 23, 24, 24, 25, 2
 5
 95 REM

--PG y PP--color piezas--

96 DATA 0, 3, 0, 3, 1, 3, 0, 3, 1, 0, 1, 0, 1, 0
 , 0, 1, 3, 0, 3, 1, 0, 1, 3, 0, 1, 3, 0, 2, 0, 2, 1,
 3, 0, 1, 3, 0, 1, 3, 0, 3, 0, 2, 0, 1, 3, 2, 0, 3, 3
 , 0, 3, 3, 0, 3, 3, 0, 2, 0, 1, 0, 0, 3, 0, 3
 97 REM

--PG y PP--coordenada ho

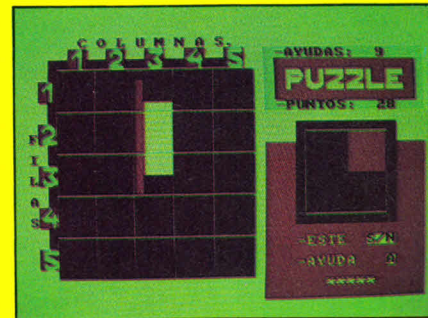
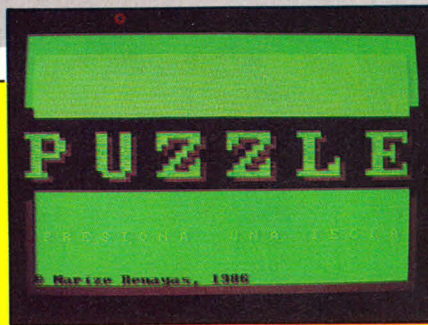
rizon--
 98 DATA 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 5, 1
 , 2, 2, 2, 3, 3, 4, 4, 4, 5, 5, 1, 1, 2, 2, 2, 2,
 2, 3, 3, 4, 4, 4, 5, 5, 1, 1, 2, 2, 2, 2, 3, 3, 3
 , 4, 4, 4, 5, 5, 5, 1, 1, 2, 2, 3, 4, 4, 5, 5
 99 REM

- NIVEL 2 -

100 DATA 4, 5, 6, 8, 3, 6, 20, 10, 19, 13, 11
 , 16, 6, 13, 3, 14, 17, 8, 16, 2, 12, 19, 13, 15
 , 7, 9, 15, 20, 22, 2, 11, 11, 20, 22, 3, 13, 17
 , 17, 20, 7, 8, 13, 5, 7, 7, 18, 22, 6, 7, 7
 101 REM

--PG--Coordenadas venta

nas-----
 102 DATA 29, 36, 10, 17, 29, 32, 14, 17, 29
 , 36, 10, 17, 29, 36, 12, 17, 29, 36, 10, 17, 2
 9, 32, 11, 17, 33, 36, 14, 17, 29, 36, 10, 17,
 29, 30, 14, 17, 33, 36, 14, 17, 29, 36, 10, 17
 103 DATA 29, 34, 14, 17, 29, 36, 10, 17, 29



, 32, 10, 11, 33, 36, 14, 17, 29, 36, 10, 17, 2
 9, 34, 14, 17, 35, 36, 10, 17, 29, 36, 10, 17,
 29, 32, 10, 17, 33, 36, 10, 17, 29, 36, 10, 17
 104 DATA 29, 32, 10, 17, 33, 36, 14, 17, 29
 , 36, 10, 17, 29, 30, 14, 17, 29, 36, 10, 17, 3
 3, 36, 10, 17, 29, 36, 10, 17, 29, 36, 10, 17,
 35, 36, 10, 13, 29, 36, 10, 17, 29, 36, 12, 17
 105 DATA 29, 32, 10, 11, 33, 36, 10, 11, 29
 , 36, 10, 17, 29, 36, 12, 17, 29, 32, 10, 11, 3
 3, 36, 10, 11, 29, 36, 10, 17, 29, 30, 10, 17,
 29, 36, 10, 17, 33, 36, 10, 17, 29, 36, 10, 17
 106 DATA 29, 36, 10, 17, 29, 36, 10, 17, 29
 , 32, 10, 17, 33, 36, 10, 11, 31, 36, 12, 17, 2
 9, 36, 10, 17, 33, 36, 10, 17, 29, 32, 10, 11,
 29, 32, 12, 17, 29, 36, 10, 17, 29, 30, 10, 17
 107 DATA 29, 36, 10, 17, 29, 36, 10, 17, 31
 , 34, 10, 15, 35, 36, 10, 15, 29, 36, 10, 17, 2
 9, 36, 10, 15, 31, 36, 10, 11, 29, 36, 10, 17,
 29, 32, 10, 11, 29, 36, 10, 17
 108 REM

--PP--Coordenadas venta

nas-----
 109 DATA 4, 7, 4, 7, 4, 5, 6, 7, 8, 11, 4, 7, 8
 , 11, 5, 7, 12, 15, 4, 7, 12, 13, 5, 7, 14, 15, 6
 , 7, 14, 19, 4, 7, 16, 16, 6, 7, 18, 19, 6, 7, 20
 , 23, 4, 7

110 DATA 20, 22, 6, 7, 4, 7, 8, 11, 4, 5, 8, 8
 , 6, 7, 10, 11, 8, 11, 8, 11, 8, 10, 10, 11, 11,
 11, 8, 11, 12, 15, 8, 11, 12, 13, 8, 11, 14, 15
 , 8, 11, 16, 19, 8, 11

111 DATA 16, 17, 8, 11, 18, 19, 10, 11, 20,
 23, 8, 11, 20, 20, 10, 11, 4, 7, 12, 15, 6, 7, 1
 2, 15, 8, 11, 12, 15, 8, 11, 12, 15, 11, 11, 12
 , 13, 12, 15, 12, 15, 12, 15, 13, 15

112 DATA 12, 13, 12, 13, 14, 15, 12, 12, 14
 , 19, 12, 15, 16, 19, 13, 15, 16, 17, 12, 12, 1
 8, 19, 12, 12, 20, 23, 12, 15, 20, 20, 12, 15,
 4, 7, 16, 19, 6, 7, 16, 19, 8, 11, 16, 19

113 DATA 8, 11, 16, 19, 12, 15, 16, 19, 12,
 13, 16, 19, 14, 15, 16, 16, 13, 15, 17, 19, 16
 , 19, 16, 19, 18, 19, 16, 19, 16, 17, 16, 16, 1
 6, 17, 17, 19, 20, 23, 16, 19, 20, 20, 16, 19

114 DATA 4, 7, 20, 23, 8, 11, 20, 23, 9, 10,
 20, 22, 11, 11, 20, 22, 12, 15, 20, 23, 12, 15
 , 20, 22, 13, 15, 20, 22, 16, 19, 20, 23, 16, 1
 7, 20, 20, 20, 23, 20, 23
 115 REM

- M-----

116 DATA 26, 26, 11, 12, 3, 28, 38, 13, 20,
 13, 31, 35, 11, 16, 3, 34, 36, 12, 18, 2, 32, 3
 7, 16, 17, 7, 30, 34, 21, 22, 2, 31, 31, 21, 22
 , 3, 33, 36, 19, 21, 7, 29, 33, 10, 11, 7, 37, 3
 9, 11, 11, 7
 117 REM

--PG y PP--numero de la

pieza--
 118 DATA 1, 1, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 6,
 6, 6, 7, 7, 7, 8, 8, 8, 9, 9, 10, 10, 11, 11, 1
 2, 12, 12, 13, 13, 13, 14, 14, 14, 14, 15,
 15, 16, 16, 17, 17, 18, 18, 18, 18, 19, 19, 19

, 19, 20, 20, 21, 22, 22, 22, 23, 23, 23, 24, 2
 4, 25
 119 REM

--PG y PP--color piezas

120 DATA 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
 3, 1, 0, 1, 3, 0, 3, 2, 0, 2, 1, 0, 1, 0, 1, 0, 1, 3
 , 0, 3, 3, 2, 0, 3, 2, 1, 0, 1, 0, 1, 0, 1, 0, 1, 2
 3, 0, 1, 2, 3, 0, 1, 0, 0, 2, 3, 0, 2, 3, 0, 3, 0
 121 REM

--PG y PP--coordenada h

orizon--
 122 DATA 1, 1, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 1,
 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 1, 1, 2, 2, 2
 , 3, 3, 3, 4, 4, 4, 4, 5, 5, 1, 1, 2, 2, 3, 3, 3,
 3, 4, 4, 4, 4, 5, 5, 1, 2, 2, 2, 3, 3, 4, 4, 5
 123 REM

- NIVEL 3 -

T-----

124 DATA 4, 6, 9, 13, 3, 5, 22, 8, 18, 13, 6,
 13, 19, 21, 7, 7, 13, 6, 8, 7, 8, 9, 22, 23, 3, 1
 0, 13, 5, 11, 2, 14, 17, 19, 21, 2, 21, 23, 6, 9
 , 7, 22, 23, 22, 23, 3, 14, 20, 10, 14, 7, 7, 23
 , 14, 15, 2
 125 REM

--PG--Coordenadas venta

nas-----
 126 DATA 29, 36, 10, 17, 35, 36, 14, 17, 29
 , 36, 10, 17, 33, 36, 12, 17, 29, 32, 14, 17, 2
 9, 36, 10, 17, 29, 32, 12, 17, 29, 36, 10, 17,
 29, 36, 10, 17, 30, 36, 14, 17, 29, 36, 10, 17
 127 DATA 31, 36, 10, 17, 35, 36, 10, 11, 29
 , 30, 12, 17, 29, 36, 10, 17, 33, 36, 10, 17, 2
 9, 32, 10, 11, 29, 32, 12, 17, 29, 36, 10, 17,
 29, 32, 10, 17, 33, 36, 10, 13, 33, 36, 14, 17
 128 DATA 29, 36, 10, 17, 29, 36, 10, 13, 29
 , 36, 14, 17, 29, 36, 10, 17, 29, 34, 10, 17, 3
 1, 36, 10, 13, 29, 30, 14, 17, 29, 36, 10, 17,
 31, 36, 10, 17, 35, 36, 14, 17, 29, 30, 10, 13
 129 DATA 29, 36, 10, 17, 29, 36, 10, 13, 29
 , 36, 14, 17, 29, 36, 10, 17, 29, 32, 10, 13, 3
 3, 36, 10, 13, 29, 36, 14, 17, 29, 36, 10, 17,
 29, 36, 10, 17, 33, 36, 10, 17, 29, 36, 10, 17
 130 DATA 29, 36, 14, 17, 31, 34, 10, 13, 29
 , 30, 10, 13, 29, 36, 10, 17, 31, 36, 10, 15, 3
 3, 36, 16, 17, 29, 36, 10, 17, 29, 36, 10, 15,
 29, 36, 16, 17, 29, 36, 10, 17, 29, 36, 10, 15
 131 DATA 29, 32, 16, 17, 33, 36, 16, 17, 29
 , 36, 10, 17, 29, 36, 10, 15, 29, 32, 16, 17, 2
 9, 36, 10, 17, 29, 34, 10, 15, 29, 36, 10, 17,
 33, 36, 10, 13, 29, 36, 10, 17, 29, 36, 10, 13
 132 DATA 29, 32, 14, 17, 29, 36, 10, 17, 29
 , 32, 10, 13, 33, 36, 10, 13, 29, 36, 10, 17, 2
 9, 32, 10, 13, 29, 36, 10, 17, 33, 36, 14, 17
 133 REM

--PP--Coordenadas venta

nas-----
 134 DATA 4, 7, 4, 7, 7, 7, 6, 7, 8, 11, 4, 7, 1
 0, 11, 5, 7, 8, 9, 6, 7, 12, 15, 4, 7, 12, 13, 5,
 7, 16, 19, 4, 7, 20, 23, 4, 7, 21, 23, 6, 7, 4, 7
 , 8, 11

135 DATA 5, 7, 8, 11, 7, 7, 8, 8, 4, 4, 9, 11,
 8, 11, 8, 11, 10, 11, 8, 11, 8, 9, 8, 8, 8, 9, 9,
 11, 12, 15, 8, 11, 12, 13, 8, 11, 14, 15, 8, 9,
 14, 15, 10, 11

136 DATA 16, 19, 8, 11, 16, 19, 8, 9, 16, 19
 , 10, 11, 20, 23, 8, 11, 20, 22, 8, 11, 21, 23,
 8, 9, 20, 20, 10, 11, 4, 7, 12, 15, 5, 7, 12, 15
 , 7, 7, 14, 15, 4, 4, 12, 13

137 DATA 8, 11, 12, 15, 8, 11, 12, 13, 8, 11
 , 14, 15, 12, 15, 12, 15, 12, 13, 12, 13, 14, 1
 5, 12, 13, 12, 15, 14, 15, 16, 19, 12, 15, 16,
 19, 12, 13, 16, 19, 14, 15, 20, 23, 12, 15

138 DATA 20, 23, 14, 15, 21, 22, 12, 13, 20
 , 20, 12, 13, 4, 7, 16, 19, 5, 7, 16, 18, 6, 7, 1
 9, 19, 8, 11, 16, 19, 8, 11, 16, 18, 8, 11, 19,
 19, 12, 15, 16, 19, 12, 15, 16, 18

139 DATA 12, 13, 19, 19, 14, 15, 19, 19, 16
 , 19, 16, 19, 16, 19, 16, 18, 16, 17, 19, 19, 2
 0, 23, 16, 19, 20, 22, 16, 18, 4, 7, 20, 23, 6,
 7, 20, 21, 8, 11, 20, 23, 8, 11, 20, 21

140 DATA 8, 9, 22, 23, 12, 15, 20, 21, 12, 1
 3, 20, 21, 14, 15, 20, 21, 16, 19, 20, 23, 16,
 17, 20, 21, 20, 23, 20, 23, 22, 23, 22, 23
 141 REM

- M-----

142 DATA 26, 26, 13, 15, 3, 27, 39, 12, 19,
 13, 28, 32, 20, 21, 7, 28, 29, 11, 12, 7, 29, 3
 0, 22, 23, 3, 30, 33, 10, 14, 2, 33, 35, 20, 21

```
,2,39,40,11,12,7,40,40,23,23,3,34,3
8,13,16,7,28,40,17,17,2
143 REM
```

-PG y PP--numero de la

```
pieza--
144 DATA 1,1,2,2,2,3,3,4,5,5,6,6,6,
6,7,7,7,8,8,8,8,9,9,9,10,10,10,
11,11,11,11,12,12,12,13,13,13,13,1
4,14,14,15,15,15,16,16,16,17,17,
17,18,18,18,19,19,19,20,20,21,21
,22,22,22,23,23,23,24,24,25,25
145 REM
```

-PG y PP--color piezas

```
146 DATA 0,3,0,2,3,0,2,0,0,3,0,1,3,
3,0,2,3,1,0,2,1,3,0,1,3,0,1,3,3,0,1
,2,3,0,1,2,0,1,3,2,0,3,2,0,2,1,3,0,
1,3,0,1,3,0,1,3,2,0,1,2,0,1,0,3,0,3
,3,0,3,2,0,2,0,3
147 REM
```

-PG y PP--coordenada h

```
148 DATA 1,1,2,2,2,3,3,4,5,5,1,1,1,
1,2,2,2,2,3,3,3,3,4,4,4,5,5,5,1,1
,1,1,2,2,2,3,3,3,3,4,4,4,5,5,5,1,
1,1,2,2,2,3,3,3,3,4,4,4,5,5,1,1,2,2
,2,3,3,3,4,4,5,5
149
```

CONTROL DE TABLAS

```
150 DIM t(11,5),pg(74,8),pp(74,8),m
(11,5),np(5,5)
151 IF ni=1 THEN tt=7:RESTORE 74:GO
SUB 156:tpg=64:RESTORE 78:GOSUB 159
:tpg=64:RESTORE 85:GOSUB 162:RESTOR
E 92:pg=64:GOSUB 165:RESTORE 94:GOS
UB 168:RESTORE 96:GOSUB 170:RESTORE
98:GOSUB 173:GOSUB 176:h1=15:pneg=
1:ppos=10:GOTO 180
152 IF ni=2 THEN tt=10:RESTORE 100:
GOSUB 156:tpg=65:RESTORE 102:GOSUB
159:tpg=65:RESTORE 109:GOSUB 162:RE
STORE 116:pg=65:GOSUB 165:RESTORE 1
18:GOSUB 168:RESTORE 120:GOSUB 170:
RESTORE 122:GOSUB 173:GOSUB 177:h1=
10:pneg=1:ppos=5:GOTO 180
153 IF ni=3 THEN tt=11:RESTORE 124:
GOSUB 156:tpg=74:RESTORE 126:GOSUB
159:tpg=74:RESTORE 134:GOSUB 162:RE
STORE 142:pg=74:GOSUB 165:RESTORE 1
44:GOSUB 168:RESTORE 146:GOSUB 170:
RESTORE 148:GOSUB 173:GOSUB 176:h1=
5:pneg=5:ppos=5:GOTO 180
154 GOTO 59
155
```

llenado T

```
156 FOR x=1 TO tt:FOR y=1 TO 5:READ
a:t(x,y)=a:NEXT NEXT
157 RETURN
158
```

llenado PG solo ventana

```
159 FOR x=1 TO tpg:FOR y=2 TO 5:REA
D b:pg(x,y)=b:NEXT NEXT
160 RETURN
161
```

llenado PP solo ventana

```
162 FOR x=1 TO tpp:FOR y=2 TO 5:REA
D c:pp(x,y)=c:NEXT NEXT
163 RETURN
164
```

llenado M

```
165 FOR x=1 TO tt:FOR y=1 TO 5:READ
d:m(x,y)=d:NEXT NEXT
166 RETURN
167
```

llenado PG-PP numero de p

```
168 FOR x=1 TO pg:READ d:pg(x,1)=d:
pp(x,1)=d:NEXT
169 RETURN
170
```

llenado PG-PP color piez

```
171 FOR x=1 TO pg:READ e:pg(x,6)=e:
pp(x,6)=e:NEXT
172 RETURN
173
```

llenado PG-PP coordenada

horiz.

```
174 FOR x=1 TO pg:READ f:pg(x,7)=f:
pp(x,7)=f:NEXT
175
```

llenado PG-PP coordenada

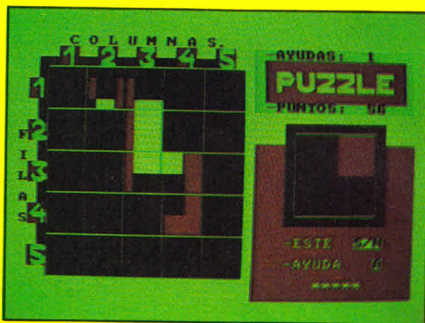
```
vert.
176 FOR x=1 TO 13:pg(x,8)=1:pp(x,8)
=1:NEXT:FOR x=14 TO 26:pg(x,8)=2:pp
(x,8)=2:NEXT:FOR x=27 TO 40:pg(x,8)
=3:pp(x,8)=3:NEXT:FOR x=41 TO 55:pg
(x,8)=4:pp(x,8)=4:NEXT:FOR x=56 TO
64:pg(x,8)=5:pp(x,8)=5:NEXT:RETURN
177 FOR x=1 TO 12:pg(x,8)=1:pp(x,8)
=1:NEXT:FOR x=13 TO 26:pg(x,8)=2:pp
(x,8)=2:NEXT:FOR x=27 TO 41:pg(x,8)
=3:pp(x,8)=3:NEXT:FOR x=42 TO 55:pg
(x,8)=4:pp(x,8)=4:NEXT:FOR x=56 TO
65:pg(x,8)=5:pp(x,8)=5:NEXT:RETURN
178 FOR x=1 TO 10:pg(x,8)=1:pp(x,8)
=1:NEXT:FOR x=11 TO 29:pg(x,8)=2:pp
(x,8)=2:NEXT:FOR x=30 TO 47:pg(x,8)
=3:pp(x,8)=3:NEXT:FOR x=48 TO 62:pg
(x,8)=4:pp(x,8)=4:NEXT:FOR x=63 TO
74:pg(x,8)=5:pp(x,8)=5:NEXT:RETURN
179
```

llenado NP -numero de 1

```
a pieza-
180 v=0:np(5,5)=25
181 FOR f=1 TO 5:FOR c=1 TO 5
182 b=INT(RND*24)+1:IF b=25 THEN 18
2
183 FOR x=1 TO 5:FOR y=1 TO 5
184 IF b=np(x,y) THEN 182
185 IF v=24 THEN 189
186 NEXT:NEXT
187 np(f,c)=b:v=v+1
188 NEXT:NEXT
189 punt=0:acie=0
190
```

Fin de la carga

```
191 DATA "F","I","N"," ","","D","E","
","L","A"," ","","R","A","R","G","A"
192 DATA "P","R","E","S","I","O","N"
,"A"," ","","U","N","A"," ","","T","E","
C","L","A"
193 LOCATE 2,2:PAPER 0:PRINT SPACE$
(15)
194 RESTORE 191:LOCATE 2,2:FOR za=1
TO 15:READ zu$:PEN 1:PRINT zu$:FO
```



```
R t=1 TO 100:NEXT:NEXT:FOR t=1 TO 1
000:NEXT
195 RESTORE 192:LOCATE 2,2:PRINT SP
ACE$(15)
196 LOCATE 1,2:FOR za=1 TO 18:READ
zu$:PEN 1:PRINT zu$:FOR t=1 TO 100
:NEXT:NEXT:CALL @BR18
197
```

INICIALIZA

```
198 INK 1,0:INK 2,0:INK 3,0
199 WINDOW#1,1,40,1,25:PAPER#1,13:C
LS#1
200 WINDOW 3,21,3,21:PAPER 0:CLS
201 WINDOW 4,23,4,23:PAPER 0:CLS
202 FOR x=1 TO tt
203 WINDOW t(x,1),t(x,2),t(x,3),t(x
,4):PAPER t(x,5):CLS
204 NEXT
205 GOSUB 298
206 WINDOW 25,40,2,7:PAPER 2:CLS
207 WINDOW 26,39,3,6:PAPER 3:CLS:WI
NDOW 1,40,1,25
208 GOSUB 227:GOSUB 273:GOSUB 282
```

Serie ORO

```
209 WINDOW#2,25,40,8,25:PAPER#2,13:
CLS#2
210 LOCATE#1,25,9:PAPER#1,13:PEN#1,
0:PRINT#1,"-";:PAPER#1,2:PRINT#1,"N
IVEL";ni;": ";:BASES"
211 LOCATE#1,26,10:PAPER#1,13:PEN#1
,3:PRINT#1,STRING$(15,"-")
212 LOCATE#1,30,12:PRINT#1,STRING$(
5,"*")
213 LOCATE#1,25,14:PAPER#1,13:PEN#1
,0:PRINT#1,"-";:PAPER#1,2:PRINT#1,"
Ayudas";:PAPER#1,13:PRINT#1,"...."
;:IF LEN(STR$(h1))=2 THEN PAPER#1,3
:PEN#1,1:PRINT#1," ";h1:ELSE PAPER#
1,3:PEN#1,1:PRINT#1,h1
214 LOCATE#1,25,16:PAPER#1,13:PEN#1
,0:PRINT#1,"-";:PAPER#1,2:PRINT#1,"
Puntos":LOCATE#1,26,17:PRINT#1,"neg
ativos";:PAPER#1,13:PRINT#1,"...";:P
APER#1,3:PEN#1,1:PRINT#1,"";:pneg
215 LOCATE#1,25,19:PAPER#1,13:PEN#1
,0:PRINT#1,"-";:PAPER#1,2:PRINT#1,"
Puntos":LOCATE#1,26,20:PRINT#1,"pos
itivos";:PAPER#1,13:PRINT#1,"...";
216 IF LEN(STR$(ppos))<3 THEN PAPER
#1,3:PEN#1,1:PRINT#1,"";:ppos ELSE
PAPER#1,3:PEN#1,1:PRINT#1,ppos
217 LOCATE#1,30,23:PAPER#1,13:PEN#1
,3:PRINT#1,STRING$(5,"*")
218 WINDOW#3,25,40,11,25:PAPER#3,3
219 WINDOW#4,28,37,9,18:PAPER#4,0
220 WINDOW#5,28,37,19,25:PAPER#5,3
221 WINDOW#6,26,40,9,23:PAPER#6,0
222 co$="C O L U M N A S":LOCATE#1
,6,1:PAPER#1,1:PEN#1,0:PRINT#1,co$
223 DATA "F"," ","","I"," ","","L"," ","","A
"," ","","S"
224 FOR zas=10 TO 18:READ pun$:LOCA
TE#1,1,zas:PAPER#1,1:PEN#1,0:PRINT#
1,pun$:NEXT
225 INK 1,25:INK 2,18:INK 3,6:BORDE
R 25:GOTO 292
226
```

Definicion de letra

```
227 SYMBOL AFTER 123
228 SYMBOL 124,&X1,&X11,&X111,&X111
1,&X11111,&X111110,&X111100,&X11000
229 SYMBOL 125,&X11100000,&X1111000
0,&X11110000,&X11110000,&X11110000
,&X11110000,&X11110000,&X11110000
230 SYMBOL 126,&X11110000,&X1111000
0,&X11110000,&X11110000,&X11110000
,&X11110000,&X11110000,&X1100000
231 SYMBOL 127,&X111,&X1111,&X11111
,&X111111,&X111100,&X111100,&X11000
,&X0
232 SYMBOL 128,&X11100000,&X1111000
0,&X11111000,&X1111100,&X1111100,&
X111100,&X111100,&X1111100
233 SYMBOL 129,&X11,&X1111,&X11111,
&X111111,&X111111,&X111111,&X111111
,&X11111
234 SYMBOL 130,&X11111100,&X1111100
0,&X11110000,&X11000000,&X11111100,
&X1111110,&X1111110,&X1111110
235 SYMBOL 131,&X11111,&X111111,&X1
11111,&X111111,&X0,&X0,&X1,&X11
236 SYMBOL 132,&X11111100,&X1111111
0,&X11111110,&X11111110,&X1111100,&
X1111000,&X11110000,&X11110000
237 SYMBOL 133,&X1,&X11000,&X111100
,&X111100,&X1111111,&X111111,&X11111
,&X1111
238 SYMBOL 134,&X11111000,&X11111110
0,&X11111110,&X1111110,&X11111110,&X1
111110,&X1111100,&X11110000
239 SYMBOL 135,&X0,&X1,&X11,&X111,&
X1111,&X11111,&X111110,&X111111
```

```

240 SYMBOL 136,&X11000000,&X1110000
0,&X11100000,&X11000000,&X10010000,
&X111100,&X111100,&X1111100
241 SYMBOL 137,&X11111111,&X11111111,&
X11111,&X0,&X0,&X0,&X0,&X0,&X0
242 SYMBOL 138,&X11111110,&X11111111
0,&X11111100,&X111100,&X111100,&X11
1100,&X111100,&X11000
243 SYMBOL 139,&X111111,&X111111,&X1
11111,&X111111,&X1111100,&X1111111,&X
111111,&X111111
244 SYMBOL 140,&X11111100,&X11111111
0,&X11111110,&X11111110,&X1100,&X0,
&X11000000,&X11111000
245 SYMBOL 141,&X111111,&X11,&X11000
,&X111100,&X11111111,&X111111,&X111111,
&X1111
246 SYMBOL 142,&X11111100,&X11111111
0,&X11111110,&X1111110,&X11111110,&X1
111110,&X11111100,&X11110000
247 SYMBOL 143,&X111111111,&X11111111
1,&X111111111,&X111111111,&X111111111,
&X11111000,&X111111111,&X111111111
248 SYMBOL 144,&X11110000,&X11111100
0,&X11111100,&X11111100,&X11111110,
&X111110,&X11111110,&X1111100
249 SYMBOL 145,&X11111000,&X11111100
0,&X111111000,&X11111000,&X11111000,
&X11111000,&X11111000,&X11111000
250 SYMBOL 146,&X111110,&X111110,&X
111110,&X111110,&X111110,&X111110,&X
111110,&X111110
251 SYMBOL 147,&X111111111,&X11111111
1,&X111111111,&X111111111,&X111111111,
&X0,&X0,&X1
252 SYMBOL 148,&X11111110,&X11111111
0,&X11111110,&X11111110,&X11111110,
&X111110,&X111110,&X11111110
253 SYMBOL 149,&X111111111,&X11111111
1,&X111111111,&X111111111,&X111111111,
&X0,&X0,&X1
254 SYMBOL 150,&X11111110,&X11111111
0,&X11111110,&X11111110,&X11111110,
&X111110,&X111110,&X11111110
255 SYMBOL 151,&X11111000,&X1111100
0,&X11111000,&X11111000,&X11111000,
&X11111000,&X11111000,&X11111000
256 SYMBOL 152,&X0,&X0,&X0,&X0,&X0,
&X0,&X0,&X0
257 SYMBOL 153,&X111111111,&X11111111
1,&X111111111,&X111111111,&X111111111,
&X11111000,&X111111111,&X111111111
258 SYMBOL 154,&X11111110,&X11111111
0,&X11111110,&X11111110,&X11111110,
&X0,&X11100000,&X11100000
259 SYMBOL 155,&X111111111,&X11111111
1,&X111111111,&X11111000,&X11111000,
&X11111000,&X11111000,&X11111000
260 SYMBOL 156,&X11111100,&X11111100
0,&X111110000,&X0,&X0,&X0,&X0,&X0
261 SYMBOL 157,&X11111000,&X11111100
0,&X11111100,&X111111111,&X111111111,
&X11111111,&X1111111,&X11111
262 SYMBOL 158,&X111110,&X111110,&X
111110,&X1111110,&X1111110,&X111110,
&X111100,&X11111000,&X11100000
263 SYMBOL 159,&X1111,&X111111,&X1111
111,&X111111111,&X111111111,&X11111111
1,&X111111111,&X111111111
264 SYMBOL 160,&X11111100,&X11111000
0,&X11000000,&X11111110,&X11111110,
&X11111110,&X11111110,&X11111110
265 SYMBOL 161,&X1111,&X1111,&X111111,
&X111111111,&X111111111,&X111111111,
&X111111111,&X111111111
266 SYMBOL 162,&X11111100,&X11111000
0,&X11000000,&X11111110,&X11111110,
&X11111110,&X11111110,&X11111110
267 SYMBOL 163,&X11111000,&X11111100
0,&X11111000,&X11111111,&X11111111,
&X11111111,&X11111111,&X11111111
268 SYMBOL 164,&X0,&X0,&X0,&X0,&X111111
0,&X11111110,&X11111110,&X11111110,
&X11111110
269 SYMBOL 165,&X111111111,&X11111111
1,&X111111000,&X111111111,&X111111111,
&X111111111,&X111111111,&X111111111
270 SYMBOL 166,&X11100000,&X11100000
0,&X0,&X11111110,&X11111110,&X111111
10,&X11111110,&X11111110
271 RETURN
272 *

```

Definicion de grafic

```

os
273 a1$=CHR$(124)+CHR$(125):a2$=" "
+CHR$(126)
274 b1$=CHR$(127)+CHR$(128):b2$=CHR
$(129)+CHR$(130)
275 c1$=CHR$(131)+CHR$(132):c2$=CHR
$(133)+CHR$(134)
276 d1$=CHR$(135)+CHR$(136):d2$=CHR
$(137)+CHR$(138)
277 e1$=CHR$(139)+CHR$(140):e2$=CHR
$(141)+CHR$(142)
278 a$=CHR$(143)+CHR$(144)+CHR$(145
)+CHR$(146)+CHR$(147)+CHR$(148)+CHR
$(149)+CHR$(150)+CHR$(151)+CHR$(152
)+CHR$(153)+CHR$(154)
279 b$=CHR$(155)+CHR$(156)+CHR$(157
)+CHR$(158)+CHR$(159)+CHR$(160)+CHR
$(161)+CHR$(162)+CHR$(163)+CHR$(164
)+CHR$(165)+CHR$(166)
280 RETURN
281 *

```

Impresion

```

282 PAPER 3:PEN 2
283 LOCATE 5,2:PRINT a1$:LOCATE 5,3
:PRINT a2$:LOCATE 2,5:PRINT a1$:LOC
ATE 2,6:PRINT a2$
284 LOCATE 9,2:PRINT b1$:LOCATE 9,3
:PRINT b2$:LOCATE 2,9:PRINT b1$:LOC
ATE 2,10:PRINT b2$
285 LOCATE 13,2:PRINT c1$:LOCATE 13
,3:PRINT c2$:LOCATE 2,13:PRINT c1$:
LOCATE 2,14:PRINT c2$
286 LOCATE 17,2:PRINT d1$:LOCATE 17
,3:PRINT d2$:LOCATE 2,17:PRINT d1$:
LOCATE 2,18:PRINT d2$
287 LOCATE 21,2:PRINT e1$:LOCATE 21
,3:PRINT e2$:LOCATE 2,21:PRINT e1$:
LOCATE 2,22:PRINT e2$
288 PAPER 3:PEN 1
289 LOCATE 27,4:PRINT a$
290 LOCATE 27,5:PRINT b$
291 RETURN
292 GOSUB 298:GOSUB 293:GOTO 317
293 LOCATE#1,6,25:PAPER#1,2:PEN#1,0
:PRINT#1,"PRESS ANY KEY"
294 IF INKEY$="" THEN 294
295 LOCATE#1,6,25:PAPER#1,13:PRINT#
1,SPACE$(15)
296 WINDOW 4,23,4,23:PAPER 0:CLS:CL
S#2:CLS#3:CLS#4
297 *

```

cuadrícula puzzle grande

```

298 FOR o1=47 TO 367 STEP 64
299 ORIGIN o1,32:DRAW 1,319
300 NEXT
301 FOR o2=32 TO 367 STEP 64
302 ORIGIN 47,o2:DRAW 319,1
303 NEXT
304 RETURN
305 *

```

cuadrícula ayudas

```

306 FOR o1=399 TO 639 STEP 48
307 ORIGIN o1,32:DRAW 1,239
308 NEXT
309 FOR o2=32 TO 287 STEP 48
310 ORIGIN 399,o2:DRAW 239,1
311 NEXT:RETURN
312 *

```

cuadrícula pieza muestra

```

313 ORIGIN 447,129:DRAW 1,128:ORIGI
N 576,129:DRAW 1,128
314 ORIGIN 447,129:DRAW 128,1
315 ORIGIN 447,256:DRAW 128,1
316 RETURN
317 *

```

pista inicial

```

318 a=INT(RND*24)+1
319 FOR x=1 TO 5:FOR y=1 TO 5
320 IF np(x,y)=a THEN 323
321 NEXT:NEXT
322 GOTO 318
323 np=np(x,y):FOR z=1 TO tpp
324 IF pp(z,1)=np THEN 327
325 NEXT
326 GOTO 318
327 WINDOW pp(z,2),pp(z,3),pp(z,4),
pp(z,5)

```

```

328 PAPER pp(z,6):CLS:GOSUB 297
329 IF z=tp THEN 331
330 IF pp(z+1,1)=np THEN z=z+1:GOTO
327
331 np(x,y)=0
332 punt=punt+ppos:acie=acie+1
333 *

```

inicio del juego

```

334 w=1
335 FOR x=w TO 5:FOR y=1 TO 5
336 IF np(x,y)=0 THEN 337 ELSE 339
337 NEXT:NEXT
338 IF acie=25 THEN 397
339 np=np(x,y):FOR z=1 TO tpp
340 IF pg(z,1)=np THEN z1=z:GOTO 34
2
341 NEXT:GOTO 338
342 WINDOW pg(z,2),pg(z,3),pg(z,4),
pg(z,5)
343 PAPER pg(z,6):CLS:GOSUB 312
344 IF z=tp THEN 346
345 IF pg(z+1,1)=np THEN z=z+1:GOTO
342
346 *

```

eleccion o ayuda

```

347 PAPER#5,3:CLS#5:LOCATE#1,28,20:
PAPER#1,3:PEN#1,1:PRINT#1,"-ESTE "
":PAPER#1,1:PEN#1,0:PRINT#1,"S/N":I
F h1<1 THEN h1=0:LOCATE#1,28,22:PAP
ER#1,3:PEN#1,1:PRINT#1,SPACE$(10):G
OTO 349
348 LOCATE#1,28,22:PAPER#1,3:PEN#1
,1:PRINT#1,"-AYUDA "":PAPER#1,1:P
EN#1,0:PRINT#1,"A":LOCATE#1,31,24:P
APER#1,3:PEN#1,1:PRINT#1,STRING$(5,
" ")
349 a$=UPPER$(INKEY$)
350 IF a$<>"S" AND a$<>"N" AND a$<>
"A" THEN 349
351 IF a$="N" THEN 352 ELSE 354
352 IF w=5 THEN w=1:GOTO 335
353 w=w+1:GOTO 335
354 IF a$="S" THEN 355 ELSE 376
355 v=1
356 sw=0:LOCATE#1,25,20:PAPER#1,3:P
EN#1,1:PRINT#1,"-":PRINT#1,"DAME C
OLUMNA "":PAPER#1,1:PEN#1,0:PRINT#1
,CHR$(241):PAPER#1,3:LOCATE#1,28,22
:PRINT#1,STRING$(11," ")
357 a$=INKEY$
358 IF VAL(a$)<1 OR VAL(a$)>5 THEN
357
359 IF sw=1 THEN b=VAL(a$):GOTO 362
360 LOCATE#1,25,20:PAPER#1,3:PEN#0,
1:PRINT#1,"-":PRINT#1,"DAME COLUMN
A "":PAPER#1,0:PEN#1,3:PRINT#1,CHR$
(241)
361 a=VAL(a$):sw=1:LOCATE#1,25,22:P
APER#1,3:PEN#1,1:PRINT#1,"-":PRINT
#1,"DAME FILA "":PAPER#1,1:PEN#1
,0:PRINT#1,CHR$(243):GOTO 357
362 LOCATE#1,25,22:PAPER#1,3:PEN#1,
0:PRINT#1,"-":PRINT#1,"DAME FILA
":PAPER#1,0:PEN#1,3:PRINT#1,CHR$
(243):FOR pa=1 TO 500:NEXT:PAPER#1,
3:LOCATE#1,25,20:PRINT#1,STRING$(15
," "):LOCATE#1,25,22:PRINT#1,STRIN
G$(15," ")
363 IF a=pp(z1,7) THEN 364 ELSE 371
364 IF b=pp(z1,8) THEN 365 ELSE 371
365 WINDOW pp(z1,2),pp(z1,3),pp(z1,
4),pp(z1,5)
366 PAPER pp(z1,6):CLS:GOSUB 297
367 IF z1=tp THEN 369
368 IF pp(z1+1,1)=np(x,y) THEN z1=z
1+1:GOTO 365
369 ENT 1,1,15,3:ENV 1,100,12,2:SOU
ND 1,85,100,11,1,1:np(x,y)=0:punt=p
unt+ppos:acie=acie+1:LOCATE#1,26,7:
PAPER#1,2:PEN#1,0:PRINT#1,"-PUNTOS:
":PEN#1,3:PRINT#1,punt:LOCATE#1,2
6,2:PEN#1,0:PRINT#1,"-AYUDAS: "":PE
N#1,3:PRINT#1,h1
370 GOTO 334
371 ENV 1,100,2,3:ENT 1,100,1,1:SOU
ND 1,284,200,11,1,1:punt=punt-pneg:
LOCATE#1,26,7:PAPER#1,2:PEN#1,0:PRI
NT#1,"-PUNTOS: "":PEN#1,3:PRINT#1,p
unt:LOCATE#1,26,2:PEN#1,0:PRINT#1,"
-AYUDAS: "":PEN#1,3:PRINT#1,h1

```

```

372 IF v=2 THEN LOCATE#1,30,20:PAPER#1,3:
PEN#1,1:PRINT#1,"TE HAS":LOCATE#1,28,21:
PRINT#1,"EQUIVOCADO":LOCATE#1,29,22:PAPER#1,6:
PEN#1,0:PRINT#1,"OTRA VEZ":FOR t=1 TO 3000:NEXT:
GOTO 347
373 PAPER#5,3:CLS#5:LOCATE#1,30,19:
PAPER#1,3:PEN#1,1:PRINT#1,"TE HAS":
LOCATE#1,28,20:PRINT#1,"EQUIVOCADO":
LOCATE#1,28,21:PRINT#1,"INTENTALO":
LOCATE#1,29,23:PAPER#1,6:PEN#1,0:P
RINT#1,"OTRA VEZ":FOR t=1 TO 3000:N
EXT
374 PAPER#5,3:CLS#5:v=v+1:GOTO 356
375 '

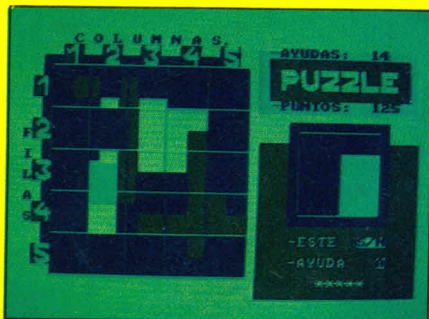
```

AYUDAS

```

376 IF a$="A" THEN 377 ELSE 349
377 CLS#2:CLS#6:h1=h1-1:LOCATE#1,26
,2:PAPER#1,2:PEN#1,0:PRINT#1,"-AYUD
AS: ";PEN#1,23:PRINT#1,h1:IF h1<0 T
HEN h1=0:GOTO 378 ELSE 380

```



```

378 PAPER#2,13:CLS#2:LOCATE#1,29,10
:PAPER#1,2:PEN#1,0:PRINT#1,"LO SIEN
TO":LOCATE#1,32,12:PRINT#1,"HAS":LO
CATE#1,30,14:PRINT#1,"AGOTADO"
379 LOCATE#1,31,16:PRINT#1,"TODAS":
LOCATE#1,32,18:PRINT#1,"LAS":LOCATE
#1,31,20:PAPER#1,3:PEN#1,1:PRINT#1,
"AYUDAS":GOTO 388
380 FOR m=1 TO tt
381 WINDOW m(m,1),m(m,2),m(m,3),m(m
,4)
382 PAPER m(m,5):CLS
383 NEXT
384 GOSUB 305
385 DATA "1","2","3","4","5"
386 RESTORE 385:PAPER#1,3:PEN#1,2:F
OR h=27 TO 39 STEP 3:READ g$:LOCATE
#1,h,8:PRINT#1,g$:NEXT
387 RESTORE 385:FOR h=10 TO 22 STEP
3:READ g$:LOCATE#1,25,h:PRINT#1,g$
:NEXT
388 LOCATE#1,26,24:PAPER#1,6:PEN#1,
0:PRINT#1,"PRESS ANY KEY"
389 b$=INKEY$:IF b$="" THEN 389
390 PAPER#2,13:CLS#2:CLS#3:CLS#4

```

```

LIME:END
394 SYMBOL AFTER 32:WINDOW 1,40,1,2
5:PAPER 8:PEN 1:END
395 '

```

Temporizador EVERY

```

396 IF acie=25 OR punt<0 THEN 397 E
LSE 403
397 PAPER#2,13:CLS#2:CLS#6:FOR m=1
TO tt:WINDOW m(m,1),m(m,2),m(m,3),m
(m,4):PAPER m(m,5):CLS:NEXT:GOSUB 3
05
398 IF punt<0 THEN punt=0:GOSUB 404
399 IF acie=25 THEN GOSUB 420
400 LOCATE#1,26,24:PAPER#1,6:PEN#1,
0:PRINT#1,"OTRA PARTIDA?":LOCATE#
1,2,24:PAPER#1,1:PEN#1,0:PRINT#1,"-
HAS GANADO":PAPER#1,2:PRINT#1,punt
+h1:PAPER#1,1:PRINT#1,"PUNTOS."
401 b$=UPPER$(INKEY$):IF b$<>"S" AN
D b$<>"N" THEN 401
402 IF b$="S" THEN ERASE t,m,pg,pp,
np:CLEAR:GOTO 1 ELSE BORDER 0:WINDO
W 1,40,1,25:PAPER 8:CLS:LOCATE 14,1
3:PRINT"- F I N -":LOCATE 1,1:END
403 RETURN
404 '

```

Adios con el coraz

```

on
405 FOR t=1 TO 2000:NEXT:ENT 1,100,
2,13:ENV 1,100,2,26
406 DATA 319,379,379,319,379,379,31
9,358,358,358,358,0,319,284,284,319
,253,284,319,358,379,379,379
407 RESTORE 406:FOR v=1 TO 23:READ
no:SOUND 1,no,40,11,1,3:FOR t=1 TO
200:NEXT:NEXT
408 RETURN
409 WINDOW#1,1,2,1,2
410 PAPER#1,1:CLS#1:LOCATE#1,1,1:PE
N#1,0:PRINT#1,"-"
411 PEN 0:LOCATE 25,25:PRINT "+"
412 WINDOW SWAP 1,0
413 FOR x=1 TO 1000:NEXT
414 GOTO 410
415 '

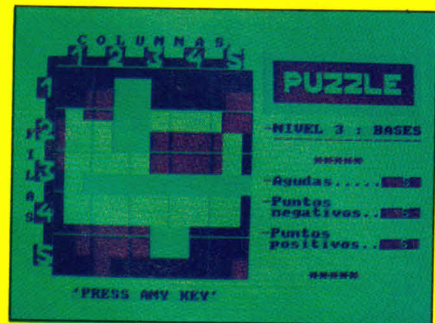
```

Felicidades !!

```

416 DATA 319,40,284,40,284,33,253,3
0,284,30,253,30,239,33,319,70,0,10
417 DATA 319,30,0,3,319,40,284,30,2
84,30,284,30,253,30,284,30,253,30,2
39,33,319,70,0,10
418 DATA 319,30,358,30,379,30,358,3
0,284,30,319,30,358,30,379,30,319,8
0,0,10
419 DATA 239,30,213,30,189,40,0,3,1
89,35,0,3,213,40,0,3,213,35,239,80,
0,10
420 RESTORE 416:ENV 1,2,1,2:ENT 1,0
,1,10:FOR t=1 TO 1000:NEXT
421 FOR x=1 TO 41
422 READ no,du
423 SOUND 1,no,du,11,1,1
424 NEXT
425 RETURN

```



```

391 GOTO 339
392 '

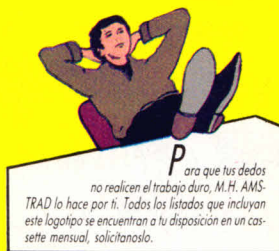
```

Tratamiento de erro

```

res Ruptura del p
rograma
393 WINDOW 1,40,1,25:PAPER 8:PEN 1:
PRINT"Error";ERR;"en linea";ERL:RES

```



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

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!

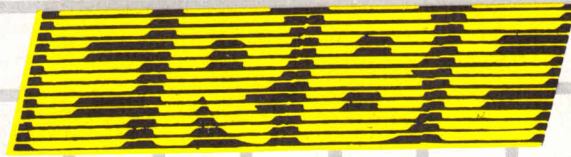
Adjuntando los siguientes datos:

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)

TU PUEDES FORMAR PARTE DE



¡Como lo oyes! Tú puedes estar informado antes que nadie de todas las novedades que aparecen en el mercado, tener acceso a programas exclusivos y a precios especiales, poder conseguir los trucos que te ayuden a salir de esa pantalla que se te resiste, regalos, pósters, sorteos, pegatinas... y un montón más de ventajas.

APUNTATE YA AL CLUB

No lo pienses más. Si te apuntas ahora al Club ERBE vas a recibir, además, como regalo de inscripción:

► El juego que más te guste de nuestro catálogo*



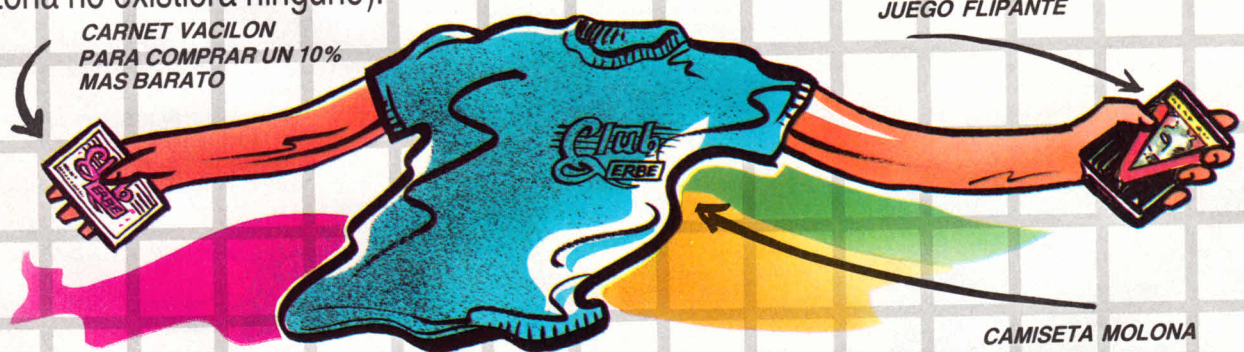
► Una fabulosa camiseta con el anagrama del Club.



► Tu carnet personal de socio, que te dará derecho al 10% de descuento en todos los juegos de ERBE que compres, durante un año, en cualquiera de los establecimientos que aquí al lado se relacionan (o a través del Club si en tu zona no existiera ninguno).



CARNET VACION
PARA COMPRAR UN 10%
MAS BARATO



JUEGO FLIPANTE

CAMISETA MOLONA

TODO POR SOLO 3.000 PTS.... UN AUTENTICO CHOLLO

Elige ya tu juego, corre al teléfono y llámanos al (91) 447 34 10 o escríbenos a ERBE SOFTWARE C/. Santa Engracia, 17 - 28010 Madrid para hacerte socio o pedir más información.

* QUEDAN EXCLUIDOS LOS PROGRAMAS EN DISQUETE

DISTRIBUIDORES AFILIADOS

ALAVA

DATAVI. Avda. Gasteiz, 29. VITORIA

ALMERIA

INFORMATICA-ELECTRONICA. Arapiles, 22.

ALICANTE

INFORTRONICA. Doctor Jiménez Díaz, 2. ELCHE.
MICRO CENTRO. César Ezquezábal, 45. ALICANTE.
MULTISYSTEM. San Vicente, 53. ALICANTE.
CODE-2000. Ramón y Cajal, 3. DENIA.
SILICON VALLEY. Glorieta, 4, semisótano. ELCHE (Alicante).

ASTURIAS

CASA-RADIO MIERES. Jerónimo Ibran, 11. MIERES.
SOVI ELECTRONICA. Cabrales, 31. GIJON.
BERNE. Menéndez Valdés, 13. GIJON.
RADIO-NORTE. Uria, 20. OVIEDO.
IMAGEN. Pablo Iglesias, 83. GIJON.
COMERCIAL ARANGO. Marcos de Termiello, 2. AVILES.
CUADRADO INFORMATICA. Toreno, 5. OVIEDO.

AVILA

DISCO-70. Plaza Sta. Teresa, 1.

BADAJOS

SONIDO RUBIO. Avda. Fdez. Calzadilla, 10. BADAJOZ.
RADIO GRAJERA. San José, 11. ALMENDRALEJO.

BALEARES

ERAGON. Falangista Laportilla, 2. P. MALLORCA.
PROCONT. Extremadura, 31. IBIZA.
COMPUSHOP. Via Alemania, 11. P. MALLORCA.
TEKNOS. Aragón, 30. P. MALLORCA.

CADIZ

PARODI-DISCOS. Novena, s/n.

CASTELLON

CASIO. S. A. San Vicente, 6. Avda. Rey Don Jaime, 74.
APARATOS. Mayor, 32. VINAROS.

LA CORUÑA

PHOTOCOPY. Juana de Vega, 29-31.

GERONA

CENTRAL FOTO. Ctra. de San Felú, 28. PLATJA D'ARO.

GRANADA

INFORMATICA-ELECTRONICA. Melchor Almagro, 8.

GUADALAJARA

ABI. Padre Félix Flores, 3.

GUIPUZCOA

SABA. Fuenterrabia, 14. SAN SEBASTIAN.
AMASONIK. Paseo Colón, 80-82. IRUN.

HUELVA

RADILUX. Concepción, 6.

LEON

MICRO BIERZO. Carlos I, 2. PONFERRADA.

LOGROÑO

COMPUTER PAPEL. Castroviejo, 19.

LUGO

MED INFORMATICA. Avda. Ramón Ferreiro, s/n.

MADRID

HIESA INFORMATICA. Camino Vinateros, 40.
INSTRUMENTOS MUSICALES ANGEL. Plaza España, 2 (local 9). LEGANES.
COMPUTIQUE. Embajadores, 90.

MALAGA

TODO INFORMATICA. Avda. Aurora, 14.
INFORMATICA EUROPA. Moreno Carbonero (Edif. Carbonero). FUENGIROLA.
ORGANIZACION EMPRESAS. Ricardo Soriano, 35. MARBELLA.
TELEVISION PIÑAS. Dr. Eusebio Ramirez, 2. SAN PEDRO ALCANTARA.

MURCIA

MEMORY SHOP. Lepanto, 1.

NAVARRA

MICROORDENADORES RAMAR. Navarro Villoslada, 7.
PAMPLONA.

ORENSE

ALMACENES MENDEZ. Capitán Cortés, 17.

PALENCIA

LA ESFERA. Mayor, 87.

SANTANDER

RADIO MARTINEZ. Doctor Jiménez Díaz, 13.

TOLEDO

CALCO. Angel de Alcázar, 56. TALAVERA DE LA REINA.

VALENCIA

RADIO COLON. Colón, 7.
ELECTRONICA MORANT. Jaime Torres, 12. GANDIA.

VALLADOLID

CHIPS AND TIPS. Plaza Tenerife, 11.
MICROLID. Gregorio Fernández, 6.

VIZCAYA

REMBAT. General Concha, 12. BILBAO.
EPROM 2. Juan XXIII, 3. SANTURCE.

ZARAGOZA

ADA COMPUTER. Independencia, 24-26.



Mercado común

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

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

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

Apartado de correos 54.062
28080 MADRID

¡ABSTENERSE PIRATAS!

Vendo Amstrad CPC 664 con monitor fósforo verde, en perfecto estado, y regalo programas comerciales valorados en más de 40.000 ptas. Todo por 85.000 ptas. Interesados llamar a Javier Rodríguez. Tel. 34 41 40. Ayala, 5 (nuevo), 9.º-1. 29002. Málaga.

Cambio programas en Barcelona más de 50 títulos como the way Exploding fist, Gremlins, Skyfox, Comando II, y muchos más. Interesados: Antonio Ropero Luceno. Avda. Can Serra, bloque X, 51, 3.º, 4.º Hospital de Llobregat (Barcelona). Tel. (93) 437 11 10. ¡Mandad lista

Ofrezco monitor en color **Amstrad** a cambio de monitor en fósforo verde **Amstrad** más diferencia. Interesados llamar al tel. 62 62 88 de Sevilla o escribir a León Carlos Alvarez Riaño. Genaro Parladé, 2, 2.º A. 410013. Sevilla.

Cambio monitor **Amstrad** en fósforo verde por monitor en color más 10.000 ptas. Mi monitor está comprado en Navidad y lleva la garantía, las 10.000 ptas. las pagaré contra reembolso. Mi dirección: Valeriano García Domínguez. Las Moreras, 157. 06006 Madrid.

Vendo impresora **BROTHER M-1009**, con Interface **CENTRONICS**, 80 columnas, 50 cps., 12 tipos de letras, posibilidad de 132 columnas por programa, etc. La vendo por 37.000 ptas. Está en perfecto uso y es a toda prueba. Avisos a: Rafael Rodrigo Rubio. Sierra Martes, 10-13.º Tel. (96) 349 38 63. Valencia.

Desearía contactar con usuarios de **Amstrad** (464, 664, 6128) para intercambio de juegos, información, utilidades, etc. Tengo más de 100. Interesados enviar relación de los que tienen al apartado de correos 50.272 de Madrid. Contestaré todas las cartas.

Mercado común

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

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

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A. AMSTRAD SEMANAL.**
Apartado de correos 54.062
28080 MADRID
¡ABSTENERSE PIRATAS!

Cambio o vendo juegos de disco o de cassette. Tengo unos 70 juegos. (Entre ellos Comando, Knight Lore, Manic Miner, Dun Darach, etc.) Interesados llamar a Fernando Jaén Julián. Rubén Darío, 17, 3.º, 3.º Tel. 33 10 05. (Preferentemente de Zaragoza.)

Poseedor de Amstrad estaría interesado en intercambios de programas con usuarios de cualquier tipo de **Amstrad**, en cassette o en disco. Interesados escribir a: Juan Carlos Benítez. Anselmo Clavé, 5, 3.º D. Palamós (Gerona).

Vendo el siguiente lote: Modulador TV CPC (464/472), totalmente nuevo, sin estrenar, en embalaje original. Fecha de compra: 30-5-86. 25 juegos a elegir entre más de 170 (enviaré lista). Libro «Hacia la inteligencia artificial con **Amstrad**». Todo por 10.000 ptas. El modulador es un regalo y no me interesa. Interesados escribir a: Manuel Angel Sánchez Costa. Avda. Amilcar Barca, 29, 2.º D. 11009 Cádiz.

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**)

Vendo impresora Seikosh GP 250 X, con interfaces serie y paralelo, 80 columnas, 64 caracteres programables y cuatro tipos de letra. Manual de instrucciones. Precio: 30.000 ptas. (negociables). Llamar al tel. 242 41 21 de Madrid. Preguntar por Juan.

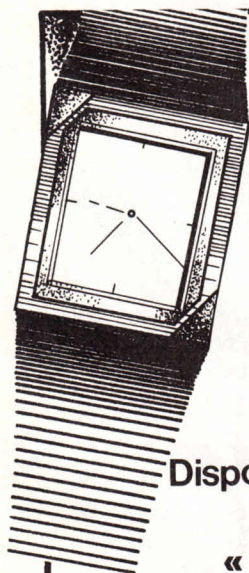
Ha surgido un club de usuarios del **Amstrad** en Zaragoza. Interesados en formar parte de él, escribir a: David Used. Mariana Pineda, 22, 6.º A. 50015 Zaragoza.

Vendo Amstrad 464 monitor fósforo verde 12 juegos y manuales. Perfecto estado, poco uso, todo por 50.000 ptas. interesados llamar al tel. 204 70 16 de lunes a viernes de 8 a 14 horas. Sábados de 8 a 13 horas. Preguntar por Javier (Madrid).

MASTER HARD

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

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



ii RECUPERA LAS ASIGNATURAS PENDIENTES !

■ SOCIALES ■ NATURALES ■ MATEMATICAS ■ LENGUAJE ■

DE : 5' 6' 7' y 8' EGB

Programa para ejercicios de ortografía **EDORTO**

Programa para ejercicios de atención y comprensión **EDACLE**

Programa para ejercitar el instrumento lector **EJEILE**

Disponible en MSX y AMSTRAD.. 1800 ptas Cassette — 2750 Disco

«PRECIO ESPECIAL» CURSO COMPLETO EGB... 3500 ptas.

PEDIDOS : **GOSLINE** CUARTELES, 43-1º-29002-MÁLAGA-TEL. 311877

... ..regalo de un reloj por cada cassette o disco

Ofites Informática

Presenta:
el lápiz al que gusta decir SI
mientras nuestros competidores dicen no
UNICO PARA AMSTRAD, CON PRECISION PIXEL

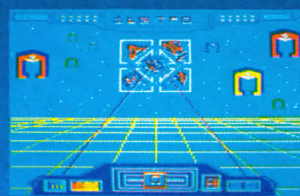
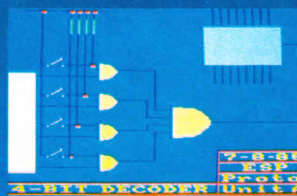
FUNCIONES	ESP	dk'tronics	OTROS
UNICO MENU DE PANTALLA	SI	NO	
ARRASTRE OBJETOS PANTALLA	SI	NO	
TRASLADO OBJETOS PANTALLA	SI	NO	
TRASLADO DE CURSOR	SI	NO	
CAJAS ELASTICAS	SI	SI	
LINEA ELASTICA	SI	SI	
TRIANGULO ELASTICO	SI	NO	
ELIPSE ELASTICO	SI	NO	
DIAMANTE ELASTICO	SI	NO	
POLIGONO ELASTICO	SI	NO	
HEXAGONO ELASTICO	SI	NO	
OCTOGONO ELASTICO	SI	NO	
CUBO ELASTICO	SI	NO	
PIRAMIDE ELASTICA	SI	NO	
CIRCUNFERENCIAS	SI	SI	
CIRCULOS RELLENOS	SI	NO	
CAJAS RELLENAS	SI	NO	
ELIPSES RELLENAS	SI	NO	
CUNAS	SI	NO	
SIMULADOR DE CORTES	SI	NO	
DISEÑO DE ZOOM	SI	SI	
IMAGEN ESPEJO E INVERTIDA	SI	NO	
FONDO DE REFERENCIA	SI	NO	
REJILLA DE FONDO	SI	NO	
OPCION DISPLAY X, Y	SI	NO	
RELLENADO CON COLOR	SI	SI	
LAVADO DE COLOR	SI	NO	
VOLCADO PANTALLA RESIDENTE	SI	NO	
DIBUJO DE BORDES EN 3 D	SI	NO	
TEXTO	SI	SI	
9 TAMAÑOS DE BROCHA	SI	NO	
18 TOBERAS MOSTRADORAS	SI	NO	
4 MEZCLAS BASICAS	SI	NO	
VARIADOR DE MEZCLAS	SI	NO	
SOMBREADO DE MEZCLAS XOR	SI	NO	
FICHERO ICONOS RESIDENTES	SI	NO	
FICHERO RELLENOS RESIDENTES	SI	NO	
26 COLORES DE PAPEL	SI	NO	
PALETA DE 15 TONOS DE COLOR	SI	NO	
POSICIONAMIENTO DE PUNTO	SI	SI	
RAYOS DESDE UN PUNTO FIJO	SI	NO	
DIBUJO REFLEJADO (ESPEJO)	SI	NO	
FUNCION HOME	SI	NO	
CONTROL DESDE TECLADO	SI	SI	
CONTROL CON JOYSTICK	SI	NO	
DISPONIBLES MODOS 1 Y 2	SI	?	

Compare con otros lápices



**TRADUCIDO
AL ESPAÑOL**

**ESTOS SON
ALGUNOS EJEMPLOS
DE LOS GRAFICOS QUE VD.
PODRA REALIZAR CON NUESTRO
LAPIZ OPTICO**



**DE VENTA EN LOS MEJORES COMERCIOS
DE INFORMÁTICA**

Si Vd. tiene alguna dificultad para obtener el lápiz óptico, puede dirigirse a:

DISPONIBLE PARA:

- CPC 464 CASSETTE 4.900 Ptas.
- CPC 464-664 DISCO 6.900 Ptas.
- CPC 6128 DISCO 6.900 Ptas.

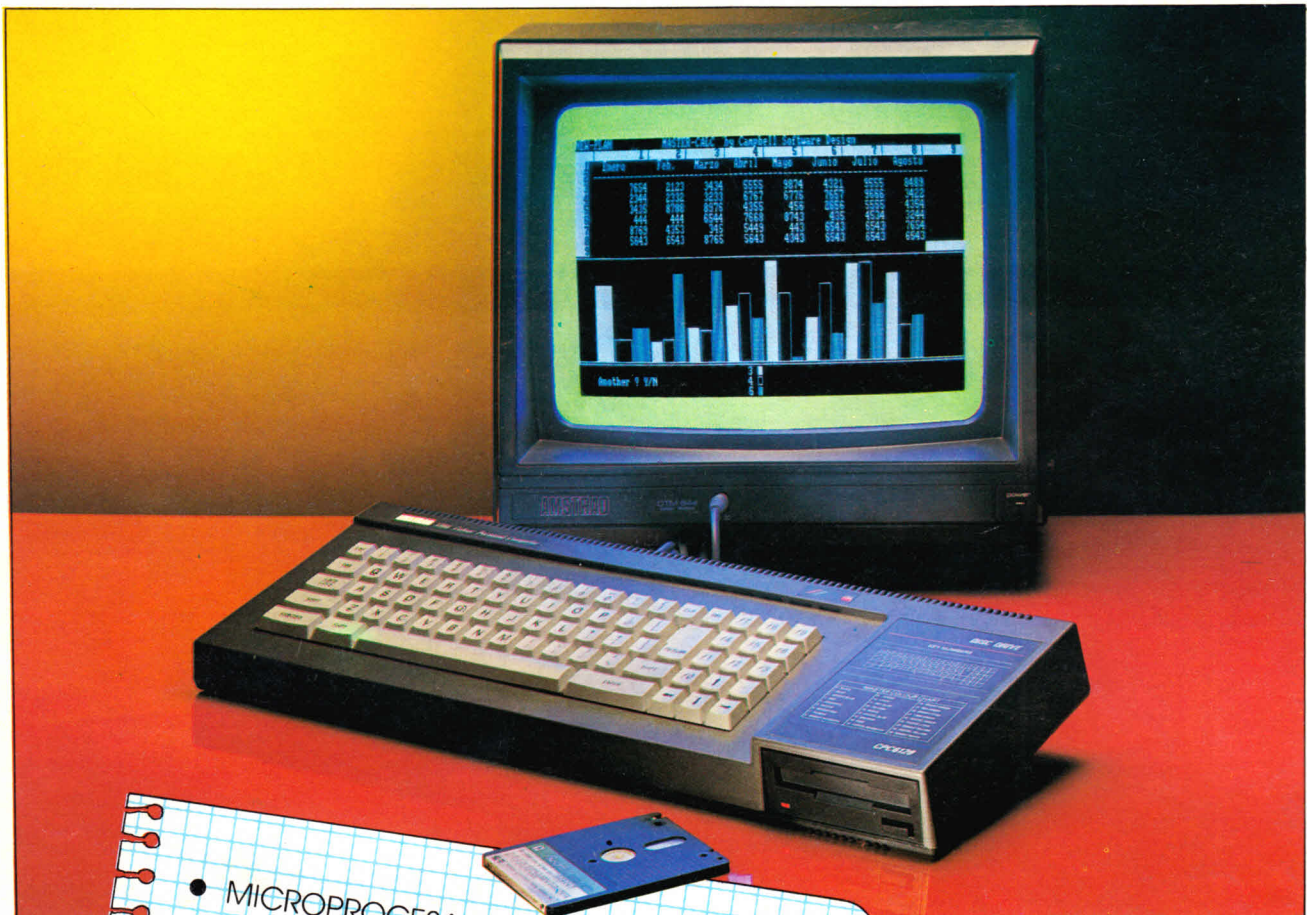
(IVA no incluido)

CONDICIONES ESPECIALES PARA DISTRIBUIDORES



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

AMSTRAD CPC-6128



- MICROPROCESADOR Z80A.
- 128 K DE MEMORIA RAM (41 K DE USUARIO EN BASIC Y 61 K EN CP/M PLUS)
- 48 K DE MEMORIA ROM QUE INCLUYEN EL LOCOMOTIVE BASIC Y EL SISTEMA OPERATIVO.
- 76 TECLAS, TECLADO NUMERICO Y DE CURSOR INDEPENDIENTE.
- TEXTO EN MONITOR DE 20, 40 U 80 COLUMNAS Y GRAFICOS CON DEFINICION DE HASTA 640 X 200 PUNTOS. 27 COLORES DISPONIBLES.
- HASTA 8 VENTANAS EN PANTALLA.
- GENERACION DE SONIDOS EN 3 VOCES Y 8 OCTAVAS.
- UNIDAD DE DISCO DE 3" (1.69 K BYTES)
- SISTEMAS OPERATIVOS AMS-DOS Y CPM/PLUS
- CONECTORES PARA IMPRESORA, JOYSTICKS, CASSETTE, SEGUNDA UNIDAD DE DISCO, ETC.

SISTEMA COMPLETO CON MONITOR EN FOSFORO VERDE, MANUAL EN CASTELLANO, GARANTIA OFICIAL AMSTRAD ESPAÑA, DISCO CON SISTEMA OPERATIVO CP/M 2.2 Y LENGUAJE DR. LOGO, DISCO CON SISTEMA OPERATIVO CP/M PLUS (CP/M 3.0) Y UTILIDADES, DISCO CON SIETE PROGRAMAS DE OBSEQUIO

84.900 Pts. + I.V.A.

SISTEMA COMPLETO IGUAL AL ANTERIOR PERO CON MONITOR EN COLOR.

119.900 Pts. + I.V.A.

AMSTRAD™
ESPAÑA

Avd. de Mediterráneo, 9, 28007 MADRID.
Tels. 433 45 48 - 433 48 76
Delegación Cataluña: C/. Tarragona, 110,
08015 BARCELONA - Tel. 325 10 58