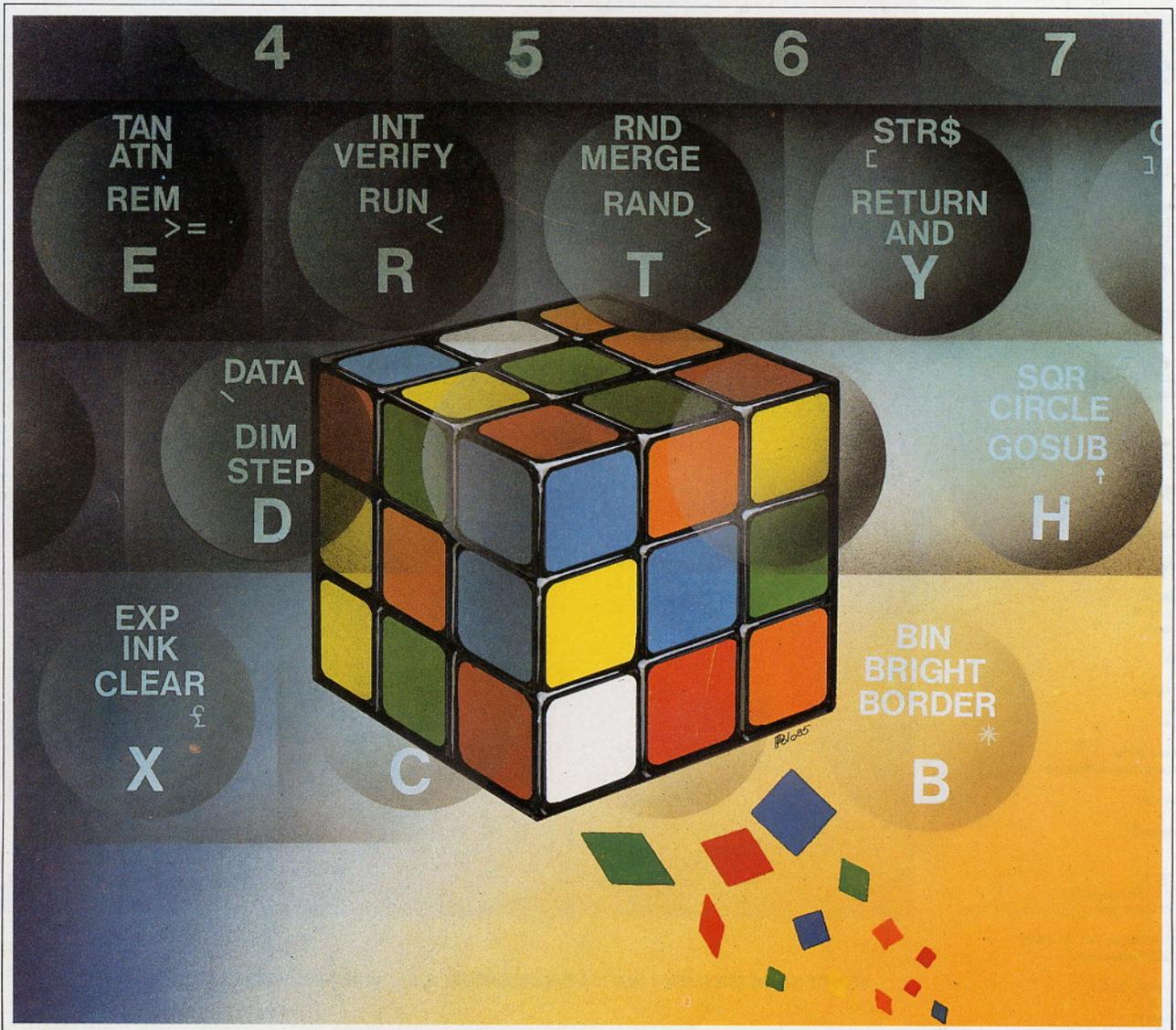


17
150pts.

ALUN

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek





PRACTICAS CON MATRICES

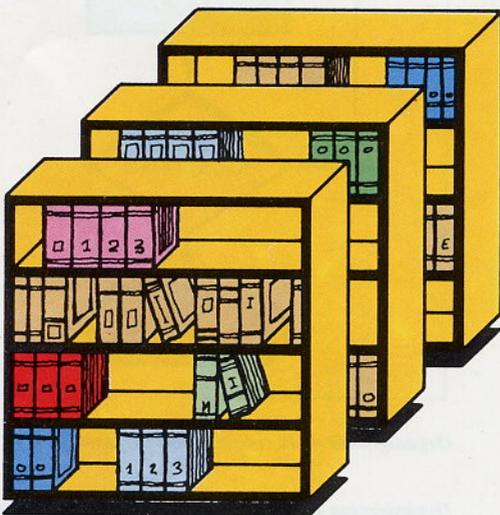


SEGURAMENTE lo más complicado del manejo de matrices sea su aplicación práctica. La verdad es que, con los conocimientos adquiridos

hasta ahora, podemos emplear de forma eficaz los vectores y tablas en nuestros programas. Sin embargo, para tener una mayor orientación sobre cómo manejar esta potente herramienta de programación, propondremos algunos ejemplos prácticos, tratando de abarcar, al menos, las aplicaciones más inmediatas de la teoría

VECTORES

Los vectores o matrices unidimensionales, son la expresión más simple de una estructura de conjunto. Nos encontramos pues, ante una serie ordenada de un determinado número de elementos, calificados por un índice. Decimos que la serie es ordenada porque, con independencia del contenido y la estructura misma de los elementos que la componen, podemos conocer la posición relativa de cualquiera de ellos dentro de la serie, así como cual le precede y cual le sigue.



J\$ (8,16)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1																
2	J	U	A	N		M	A	N	U	E	L					
3	J	O	S	E		F	E	R	N	A	N	D	O			
4	V	I	C	E	N	T	E									
5	C	A	R	L	O	S										
6	A	N	A													
7	J	O	S	E												
8	M	I	L	A	G	R	O	S								
	A	N	T	O	N	I	O									

La representación de un vector de cadena es una tabla.

Supongamos, por ejemplo, un juego en el que intervengan N jugadores. Si almacenamos sus nombres en una matriz, siempre sabremos a qué jugador le corresponde el siguiente turno de juego, sólo con sumar 1 al índice del vector, y controlando el hecho de hacer éste igual a 1 cuando, por efecto de la operación de suma, su valor se haga igual a N+1, es decir, exceda al último jugador:

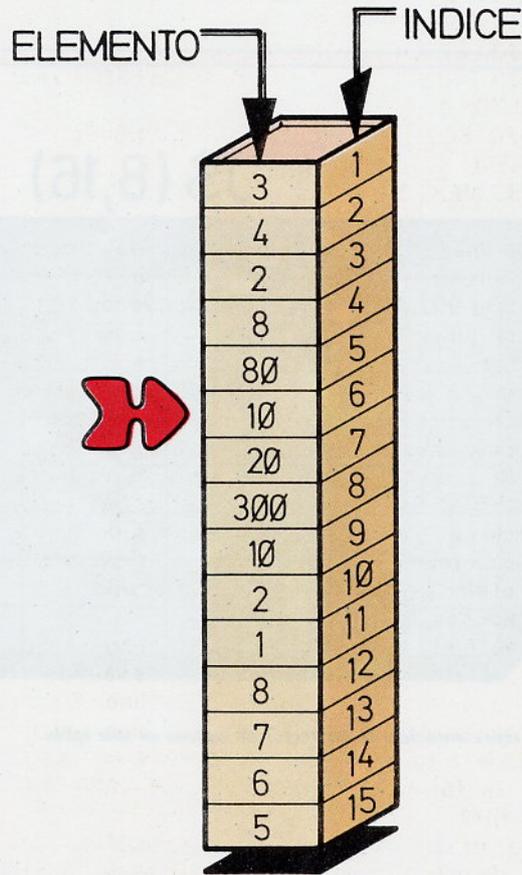
```

10 REM - TURNO DE JUGADORES
20 INPUT "Numero de jugadores:";N
30 DIM J$(N,16)
40 FOR I=1 TO N
50 INPUT "Jugador "+STR$ I+":";LINE
J$(I)
60 NEXT I
70 PRINT "TURNOS","JUGADORES"
80 FOR I=0 TO 31: PRINT "=";: NEXT I
90 PRINT
100 FOR I=1 TO N: PRINT I,J$(I): NEXT I

```

Las matrices aportan al BASIC un sistema altamente eficaz para el almacenamiento y clasificación de datos.

Una matriz es una serie ordenada, en la cual, independientemente de su contenido, podemos conocer mediante el índice, la posición relativa que ocupan sus elementos.



i!

La hipotética cuarta dimensión, tan traída y llevada por la ciencia ficción, puede ser claramente imaginada informáticamente mediante una estructura tetradimensional. Su representación geométrica es evidentemente bastante compleja, aunque generalmente se le asigna el sentido de tiempo a la cuarta dimensión, de forma que estas matrices pueden representarse mediante estructuras espaciales (tridimensionales), analizadas en un determinado instante (cuarta dimensión).

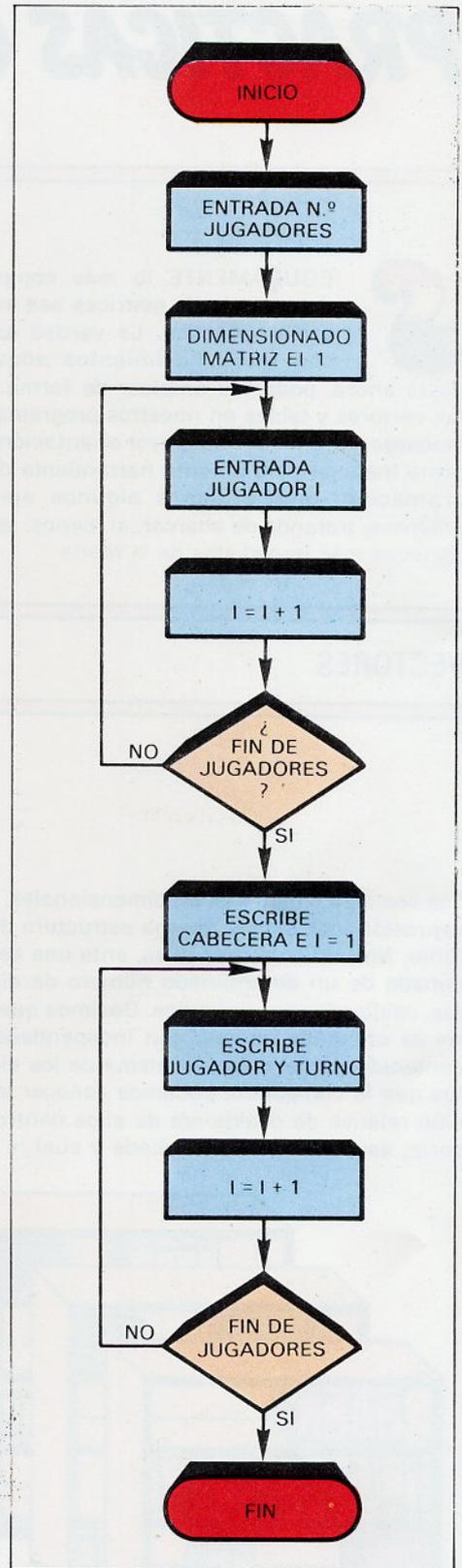
*

El vector es una serie ordenada porque, con independencia del contenido de los elementos que lo componen, sabemos la posición relativa de uno determinado de ellos dentro de la serie, facilitándonos así mismo conocer cual es el que le precede y el que le sigue.

MATRICES NUMERICAS DE UNA DIMENSION

Las matrices pueden definirse como multidimensionales. En cualquier caso, la estructura más simple de matriz numérica es el vector, es decir, la matriz unidimensional.

En base a los conocimientos adquiridos en el capítulo anterior, proponemos ahora el desarrollo de un programa que, a partir de una serie de un número variable de elementos, gestione la extracción al azar de la totalidad de ellos, sin repetición. En el programa de ejemplo, hemos definido la variable **N** como 90, para simular las extracciones de bolas en el juego del bingo:



Organigrama de la rutina turno de jugadores.

Organigrama de la rutina de extracción aleatoria.

```

10 REM EXTRACCION ALEATORIA
J.M. LOPEZ MARTINEZ
20 LET N=90: DIM S(N)
30 FOR I=1 TO N: LET S(I)=I: NEXT I
40 FOR I=1 TO N
50 LET X=INT (RND*(N-I))+1
60 PRINT (" " +STR$(S(X)))(LEN STR$(S(X)) TO )
70 FOR J=X TO N-I: LET S(J)=S(J+1):
NEXT J
80 NEXT I
    
```

En la línea 20, se define el número de elementos de la serie de aleatorios a producir, en nuestro caso 90. A continuación, dimensionamos un vector numérico **S()**, por medio de la sentencia **DIM**, con un número **N** de elementos. Esta lista contendrá la serie ordenada desde el 1 hasta el valor que asignemos de comienzo a la variable **N**. En la instrucción 30, se efectúa la carga de la matriz **S()**, siendo el contenido de cada elemento idéntico al del índice que la califica, es decir, coincide el valor del elemento en cuestión con su posición relativa dentro del vector. Así por ejemplo, el elemento uno contiene el valor uno, el dos el valor dos, etc...

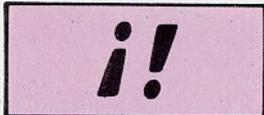
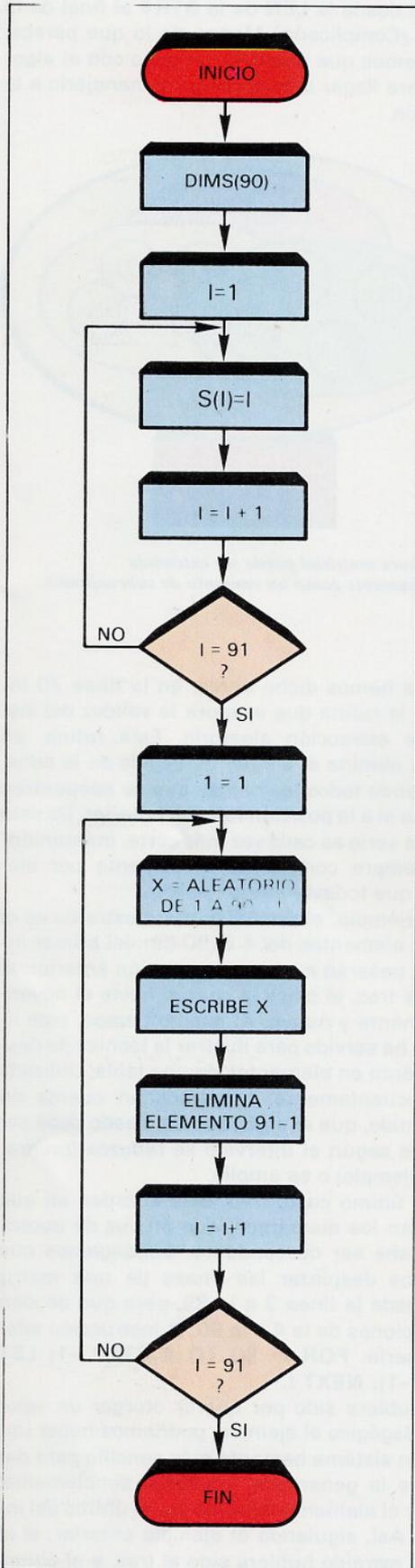
En las líneas 10 a 80 se encuentra el bucle central del programa, establecido para los valores de **I** comprendidos entre 1 y **N**, y en la línea 50, se efectúa la extracción aleatoria de un elemento de la serie, a través de la función **RND**, calculado con la fórmula siguiente: **LET X=INT(RND*(N-I))+1**.

Antes de comentar en detalle el por qué del empleo de esta fórmula, debemos hablar de las posibilidades que se nos ofrecen a la hora de extraer, de una serie de elementos, algunos de ellos, sin que éstos puedan repetirse.

La primera solución es la propuesta en el programa de ejemplo del capítulo anterior, donde a cada nuevo elemento seleccionado se le hace una marca en una posición determinada para, posteriormente, ejecutar una generación aleatoria controlada. Este control consiste en repetir una y otra vez la generación aleatoria, hasta que arroje como resultado un elemento de la matriz sin marca, es decir, inédito hasta el momento.

Lógicamente, a ninguno se nos escapa que este método es únicamente eficaz cuando, sobre una serie grande, se han de extraer unos pocos elementos, como es el caso de las diez preguntas sobre el total de los países europeos; esto es debido a que cada vez se genera un número aleatorio, comprendido en el intervalo que va desde uno al total de países europeos. Si en nuestro ejemplo actual empleamos el mismo procedimiento, según se van extrayendo los números, se hace más y más difícil, obtener uno que no haya salido ya.

Con el empleo de la fórmula de la línea 50, el margen se reduce progresivamente desde **N** has-



Una matriz de dos dimensiones, es un conjunto de conjuntos, es decir, una estructura formada por varios conjuntos de un nivel superior, representados por el primer índice, y varios de nivel inferior, representados por el segundo.

*

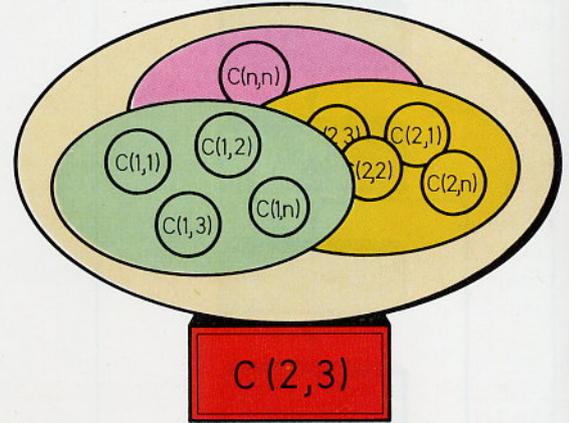
Al hablar de matrices de tres dimensiones, nos movemos en un conjunto que abarca a un subconjunto, y éste a su vez a otro subconjunto de él, de forma anidada y perfectamente estructurada.

ta uno, por cada nueva pasada del bucle principal de programa **FOR NEXT**, obteniéndose mejores resultados. Sin embargo, supongamos que el primer número seleccionado no sea el último de la serie (el noventa en nuestro caso), cosa por otra parte harto probable; al reducir en uno el margen de operación de **RND** (de 1 a 89), descartaría la elección del 90, con lo cual este sistema es auténticamente catastrófico, puesto que sólo funcionará cuando el número elegido en cada ocasión sea el último del intervalo (90, 89, 88, 87, etc...). La explicación de por qué elegir este método, aparentemente absurdo, se encuentra en la línea 70, estudiada un poco más adelante, la cual sirve de complemento a la línea de extracción aleatoria, y hace que el nuevo sistema sea completamente válido.

En la línea 60 se imprime el elemento de la serie obtenido en la línea anterior, justificado por la derecha en cuatro posiciones, cosa que hacemos para que los números queden columnados y aparezcan 8 en cada línea. El algoritmo empleado para conseguirlo es: **PRINT (" " +STR\$ S(X))(LEN STR\$ S(X) TO)**.

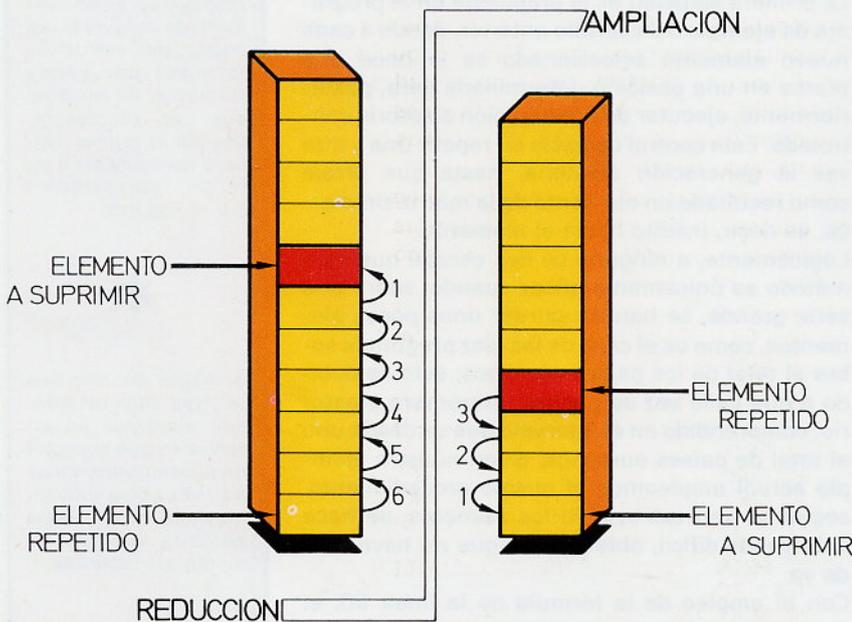
Generalizando, para todas las ocasiones en que nos interese obtener columnas de números ajustados a la derecha, podemos decir que el sistema consiste en montar una cadena, suma de tantos espacios como deseemos sea el ancho de la columna menos 1, más la **STR\$** de la variable a imprimir, para escribir de toda ella la parte com-

prendida desde la **LEN** de la **STR\$** al final de la cadena. ¿Complicado? Menos de lo que parece, sólo tenemos que practicar un poco con el algoritmo, para llegar a comprender y manejarlo a la perfección.



La estructura matricial puede ser entendida matemáticamente como un conjunto de subconjuntos.

Según el desplazamiento de elementos de una matriz tienda a reducirla o a ampliarla, el STEP del FOR empleado en el algoritmo, deberá ser negativo o positivo, respectivamente.



Como ya hemos dicho antes, en la línea 70 incluimos la rutina que asegura la validez del sistema de extracción aleatoria. Esta rutina, en esencia, elimina el elemento elegido de la serie, desplazando todos los demás que se encuentran detrás de él a la posición relativa anterior. De esta forma, la serie es cada vez más corta, manteniéndose siempre compuesta únicamente por elementos que todavía no han salido.

Si, por ejemplo, el primer número extraído es el tres, los elementos del 4 al 90 (fin del primer intervalo), pasarán a ocupar la posición anterior: el cuatro la tres, el cinco la cuatro, hasta el noventa la ochenta y nueve. Al mismo tiempo, esta línea nos ha servido para ilustrar la técnica de desplazamiento en elementos de una tabla, utilizada muy frecuentemente; tengamos en cuenta en este sentido, que el algoritmo empleado debe ser diferente según el intervalo se reduzca (programa de ejemplo) o se amplíe.

En este último caso, sólo varía el orden en que se alteran los elementos, que en vez de ascendente debe ser descendente. Supongamos que queremos desplazar las líneas de una matriz **S()**, desde la línea 3 a la 89, para que ocupen las posiciones de la 4 a la 90; la instrucción adecuada sería: **FOR I= 90 TO 4 STEP -1: LET S(I)=S(I-1): NEXT I**.

Si no hubiera sido por querer otorgar un valor más pedagógico al ejemplo, podríamos haber utilizado un sistema bastante más sencillo para dar validez a la generación aleatoria: simplemente, sustituir el elemento extraído por el último del intervalo. Así, siguiendo el ejemplo anterior, si el número extraído hubiera sido el tres, y el último



del intervalo el noventa, el problema se hubiera limitado a efectuar $LET S(3)=S(90)$, y generalizando, la línea 70 del programa, pasaría a ser $70 LET S(X)=S(N-I+1)$.

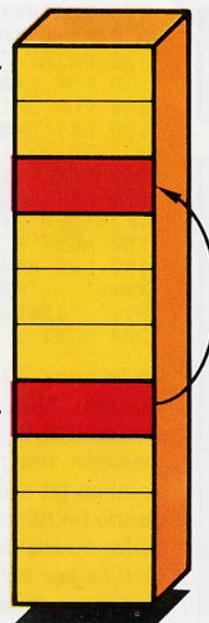
Por último, en la línea 80, se cierra el bucle principal del programa con la instrucción **NEXT I**. De esta forma, el bucle de la línea 70 queda anidado dentro del bucle exterior.

Como podemos ver, el hecho de haber definido el número de elementos de la serie mediante una variable (**N**), nos permite alterarla fácilmente, flexibilizando así el programa, puesto que funcionará tanto con 90 elementos (ejemplo), como sería el caso de la extracción de bolas de bingo, como con 40, para la extracción de cartas de la baraja española, o cualquier otra aplicación similar.

POSICION ESCOGIDA

INTERVALO

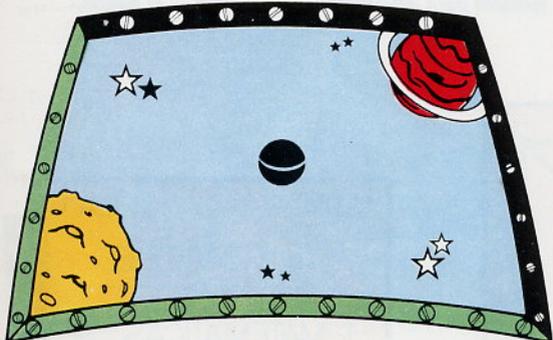
ULTIMO ELEMENTO



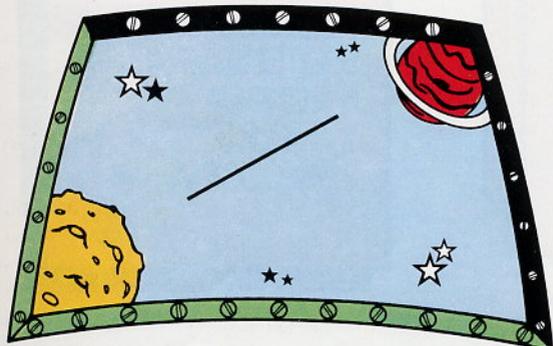
El sistema más fácil para dar validez al algoritmo de extracción aleatoria, habría sido hacer una copia en la posición escogida del último elemento del intervalo.

MATRICES NUMERICAS DE DOS DIMENSIONES

Para tener una idea clara de la estructura de una matriz de dos dimensiones, debemos pensar en



Las variables, o la matrices unidimensionales de un solo elemento, pueden representarse geométricamente por un punto.



La representación geométrica de listas (vectores) es una línea.

que nos movemos dentro de un conjunto de conjuntos, es decir, tenemos varios conjuntos de un nivel superior, representados por el primer índice, y varios de nivel inferior, representados por el segundo.

Dado que para acceder a un elemento determinado debemos especificar sus dos índices, vemos claramente una analogía con la representación cartesiana en el plano, a través de la abscisa y la ordenada; debido a ello, este tipo de variables suscritas se conocen como **TABLAS**, puesto que gráficamente pueden ser representadas como una tabla formada por filas y columnas, o si los preferimos, como una serie de vectores situados uno al lado de otro.

A modo de ejemplo, proponemos un programa por el que se obtiene un informe sobre las calificaciones y notas medias de los 9 alumnos de una clase.

```

10 REM ESTADISTICA DE CALIFICACIONES
J.M LOPEZ MARTINEZ
20 DIM C(9,8)
30 PRINT TAB 7; "- CARGA DE DATOS -"
40 FOR L=0 TO 31: PRINT "=";: NEXT L
50 PRINT ""
60 FOR J=1 TO 9: FOR K=1 TO 8
70 INPUT "Alumno "+STR$ J+" NOTA "+STR$
K+" :";C(J,K): IF C(J,K)<0 OR C(J,K)>10
THEN GO TO 70
80 PRINT (" "+STR$ C(J,K))(LEN STR$
C(J,K) TO );
90 NEXT K: PRINT "": NEXT J
100 CLS : PRINT "NUM";

```

!

Los vectores o matrices unidimensionales, son la expresión más simple de una estructura de conjunto. Consisten en una serie ordenada de un número determinado de elementos, calificados por un índice.

La única limitación dimensional en las matrices la impone la propia memoria; por tanto, podemos crear estructuras de datos de extraordinaria complejidad.

i!

En general, cualquier estructura matricial, se puede entender como un árbol invertido, en el que cada índice genera las ramas correspondientes a otros índices.

*

Las estructuras matriciales de dos dimensiones, son representadas usualmente como tablas, en las cuales uno de los índices son las filas y el otro las columnas, por tanto, forman geométricamente un plano.

*

También es usual representar las estructuras matriciales como conjuntos de subconjuntos.

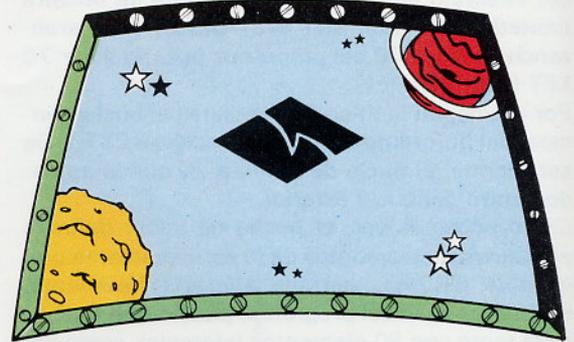
```

110 FOR K=1 TO 8: PRINT ( ' N'+STR$ K)(L
EN STR$ K TO );: NEXT K
120 PRINT " N.M"
130 FOR L=0 TO 31: PRINT "=";: NEXT L
140 FOR J=1 TO 9
150 LET S=0: PRINT ( " "+STR$ J)(LEN ST
R$ J TO );
160 FOR K=1 TO 8
170 LET S=S+C(J,K): PRINT ( " "+STR$ C(
J,K))(LEN STR$ C(J,K) TO );
180 NEXT K
190 LET X#=STR$ INT ((S/8+.05)*10): LET
X#=( " "+X$)(LEN X$ TO ): PRINT X$( TO
3);. ".X$(4)'"
200 NEXT J
    
```

En la línea 20 se define, por medio de una instrucción **DIM**, la matriz numérica de dos dimensiones **C(9,8)**. Al corresponder un índice por dimensión, nos referiremos en adelante al primero de ellos (el número de alumno) como **J**, y al segundo (el número de nota de cada alumno) como **K**. La matriz tiene, pues, una estructura de nueve filas por ocho columnas, accediéndose a cada elemento de la misma, indicando los correspondientes valores de **J** y **K** (alumno y nota).

En las líneas 30 a 50 se gestiona la cabecera de la carga de datos. Primero se imprime un título centrado y, seguidamente, una línea de guiones tras la cual se produce una de separación.

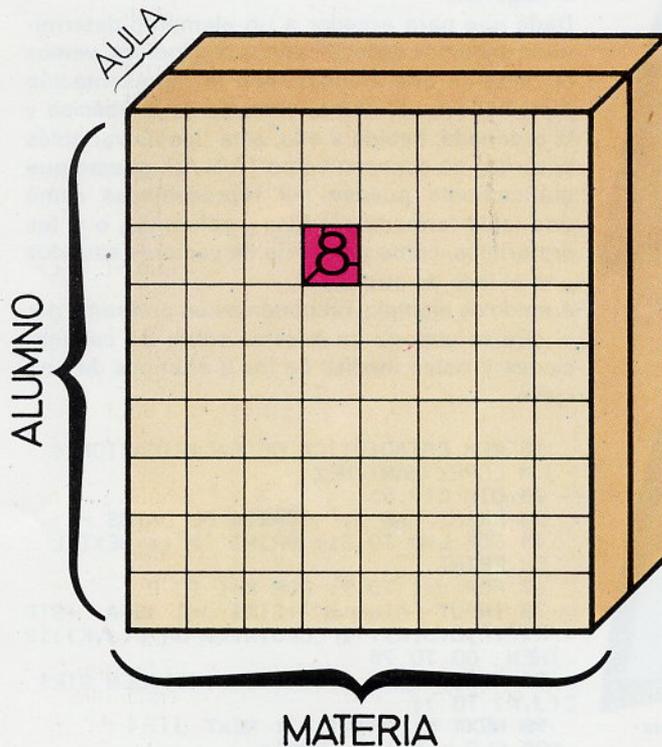
En la líneas 60 a 90 se gestiona la entrada de datos propiamente dicha. La 60 abre los bucles **FOR NEXT** anidados de las variables **J** y **K**, los cuales realizan la carga ordenada de los datos en la tabla, gracias a la siguiente instrucción, que contiene el **INPUT**.



La representación geométrica escogida para una matriz bidimensional, es un plano.

La instrucción 70 se divide en dos partes bien diferentes. En la primera de ellas, se aceptan los datos con un mensaje de petición, construido a base de las palabras descriptivas de los índices y los respectivos valores de éstos mismos, conociéndose de este modo, en todo momento, el dato exacto a teclear. En la segunda parte, se efectúa la depuración mínima necesaria para obtener una cierta garantía sobre la validez de los datos introducidos. En este caso, como el **INPUT** es de tipo numérico, no es necesario verificar que los datos introducidos lo sean. De esta forma, con-

El aula es el tercer índice utilizando en el programa de ejemplo para determinar un elemento de la matriz.



ALUMNO 3	AULA 1
CALIFICACIONES	
1	MATEMATICAS 9
2	FISICA 2
3	QUIMICA 4
4	BIOLOGIA 8
5	LITERATURA 5
6	LENGUA 7
7	INGLES 9
8	FILOSOFIA 3

centramos nuestros esfuerzos en controlar que éstos se encuentren dentro de los márgenes razonables (0 a 10).

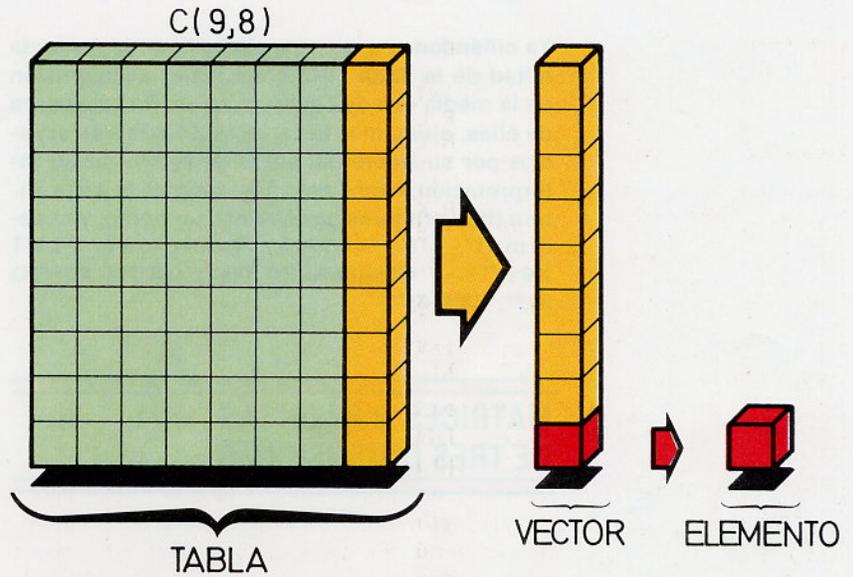
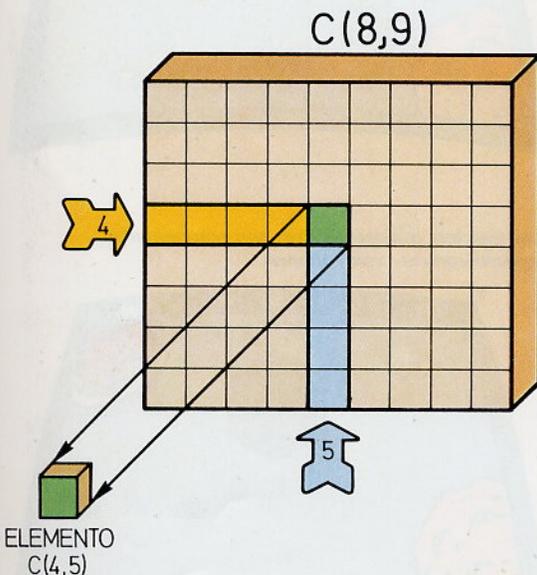
En la línea 80 se van imprimiendo los datos introducidos justificados a la derecha, dentro de columnas de 3 posiciones cada una, completando así una línea por cada 8 calificaciones introducidas.

La línea 90 cierra los bucles anidados de notas y alumnos, controlando un espacio doble entre líneas por medio de la instrucción **PRINT''**, concluyendo en esta línea la carga de datos en la matriz.

Las líneas de la 100 a la 200 gestionan la impresión de la estadística de resultados obtenidos. Dentro de ellas, las líneas 100 a 130 imprimen una cabecera para los resultados. Esta cabecera se construye, partiendo del borrado de la pantalla y la impresión de un título, para la columna destinada a albergar los números de los alumnos. A continuación, se imprimen una serie de ocho cabeceras formadas por la inicial **N** seguida del número de la columna. Por último, recurrimos a la misma rutina de subrayado con el símbolo igual (=), ya empleada en ocasiones anteriores.

En las líneas 140 a 200 se desarrolla verdaderamente la actividad de impresión de calificaciones y cálculo e impresión de la nota media: la 140 comienza con el bucle exterior de alumnos; seguidamente, la instrucción 150 imprime el comienzo de la línea de detalle de calificaciones de cada alumno, consistente en su número centrado fren-

Para designar cualquier elemento de una tabla, son precisos dos índices.



La matriz numérica puede utilizarse como conjunto de varios vectores (tabla). Tal es el caso de la matriz C(9,8) en el programa de ejemplo.

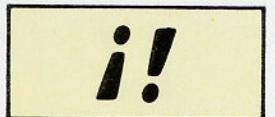
te a la cabecera, haciendo cero la variable **S**, que contiene la suma de las notas de cada alumno para el cálculo de la media.

En las líneas 160 a 180 codificamos el bucle de impresión de calificaciones columnadas por la derecha, con acumulación en la variable **S** de cada nota.

Por último, en la instrucción 190 resolvemos la impresión de la nota media, después de efectuar su cálculo. La operación es simple, bastando con dividir **S** por ocho. Sin embargo, nosotros la hemos complicado un poco, al asegurarnos de que las notas medias se impriman redondeadas y con un decimal. Para poder hacerlo, utilizaremos el algoritmo: $INT((S/8+.05)*10)$.

Este algoritmo puede utilizarse de forma general para imprimir los valores numéricos redondeados a un número fijo de decimales. Para ello, debemos tener en cuenta que a la variable en cuestión es preciso sumarle un 5 en la posición decimal siguiente por la derecha a la que pretendemos redondear el resultado. En nuestro ejemplo, sumamos .05 porque queremos redondear a un decimal. Si quisiéramos redondear a enteros escribiríamos .5 y para redondear el resultado a dos decimales .005.

Una vez hecho esto, basta con multiplicar el resultado obtenido por 10, 100,... para obtener uno, dos... decimales, y por último, calcular la parte entera de lo obtenido hasta el momento. Con esto concluye la fase de cálculo, pero aún falta imprimir los resultados, es decir, interpretar el contenido de la variable alfanumérica en la cual hemos colocado el resultado de las operaciones, en nuestro caso **X\$**, cuyo valor es la **STR\$** del **INT** del valor numérico obtenido.



Para acceder a un elemento determinado debemos especificar los dos índices; una clara analogía con la representación cartesiana en el plano, a través de la abscisa y la ordenada.

*

Las matrices tridimensionales precisan de tres índices para calificar a cada elemento, lo que implica cierta relación con la representación espacial pura, en tres dimensiones.

Ya ciñéndonos a nuestro ejemplo, en la segunda mitad de la línea 190, efectuamos la impresión de la media con dos instrucciones. En la primera de ellas, ajustamos la variable **X\$** con tres espacios por su izquierda, antes de comenzar su interpretación; hecho esto, imprimimos la parte entera (tres primeras posiciones), un punto, y el decimal (la cuarta posición): **LET X\$=(" "+X\$(LEN X\$ TO 4):PRINT X\$(TO 3);".";X\$(4))**.

MATRICES NUMERICAS DE TRES DIMENSIONES

Las matrices tridimensionales precisan de tres índices para calificar a cada elemento. Si hemos dicho que las bidimensionales, tienen cierta similitud con la representación cartesiana en el plano, podemos decir ahora que éstas la tienen con la representación espacial pura, en tres dimensiones. Nos movemos, en este caso, en un conjunto que abarca a un subconjunto, y éste a su vez a otro subconjunto de él, de forma anidada y perfectamente estructurada.

Volviendo al ejemplo anterior de las calificaciones, podemos proponer ahora una ampliación al problema planteado anteriormente, donde el análisis se extiende a las dos aulas que componen el curso. Definimos de esta forma una matriz tridimensional **C(2,9,8)**, e índices **I, J y K**.

Por lo demás, el desarrollo del programa es bastante similar al anterior, con la inclusión de una rutina de borrado parcial de la pantalla, para admitir separadamente las calificaciones de las dos aulas, tal como podemos observar en el correspondiente listado.

```

10 REM ESTADISTICA DE CALIFICACIONES - J.M. LOPEZ MARTINEZ
20 DIM C(2,9,8): LET S$="": FOR L=0 TO 31: LET S$=S$+" ": NEXT L
30 PRINT TAB 7; "- CARGA DE DATOS -"
40 FOR L=0 TO 31: PRINT "=";: NEXT L
50 PRINT ""
60 FOR I=1 TO 2: PRINT AT 2,0: FOR L=0 TO 16: PRINT S$: NEXT L: PRINT AT 2,0: FOR J=1 TO 9: FOR K=1 TO 8
70 INPUT "Curso "+STR$ I+" alumno "+STR$ J+" NOTA "+STR$ K+"": C(I,J,K): IF C(I,J,K)<0 OR C(I,J,K)>10 THEN GO TO 70
80 PRINT (" "+STR$ C(I,J,K))(LEN STR$ C(I,J,K) TO 3);".";X$(4)
90 NEXT K: PRINT "": NEXT J: PAUSE 0: NEXT I
    
```

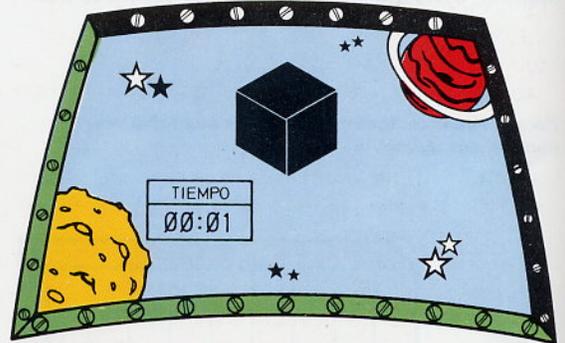
```

100 CLS : PRINT "NUM";
110 FOR K=1 TO 8: PRINT (" N"+STR$ K)(LEN STR$ K TO 3);: NEXT K
120 PRINT " N.M"
130 FOR L=0 TO 31: PRINT "=";: NEXT L
140 FOR I=1 TO 2: PRINT AT 2,0: FOR L=0 TO 16: PRINT S$: NEXT L: PRINT AT 2,0: FOR J=1 TO 9
150 LET S=0: PRINT (" "+STR$ J)(LEN STR$ J TO 3);: NEXT J
160 FOR K=1 TO 8
170 LET S=S+C(I,J,K): PRINT (" "+STR$ C(I,J,K))(LEN STR$ C(I,J,K) TO 3);: NEXT K
180 NEXT K
190 LET X$=STR$ INT ((S/8+.05)*10): LET X$=(" "+X$(LEN X$ TO 4):PRINT X$(TO 3);".";X$(4)
200 NEXT J: PAUSE 0: NEXT I
    
```

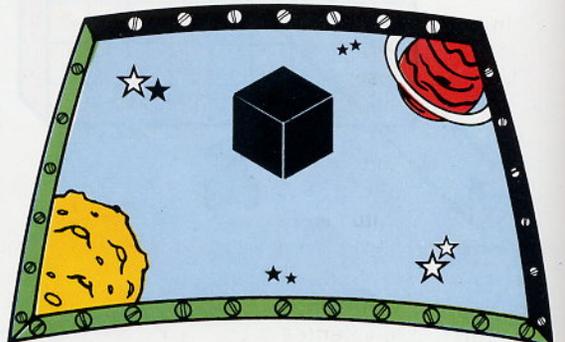
Para finalizar añadiremos que las matrices no presentan una restricción a tres dimensiones, sino que también podemos construir estructuras tetradimensionales, etc... Así por ejemplo, podríamos conservar en una misma matriz las notas obtenidas por todos los alumnos en todas las asignaturas para las diferentes clases, y además, durante todo el curso; en este caso, la cuarta dimensión escogida sería el tiempo (los meses del curso).



Las representaciones tetradimensionales son extraordinariamente más complejas que las vistas hasta ahora; un caso habitual es considerar el tiempo como la cuarta dimensión.



Las matrices tridimensionales son representadas geoméricamente con volumen.



Las estructuras matriciales unidimensionales (vectores o listas), se pueden representar como una línea, o como un punto en el caso particular de un vector con un solo elemento.



Las estructuras matriciales tridimensionales, se pueden representar gráficamente como un volumen, en el cual uno de los índices corresponde a la altura, otro a la anchura y el restante a la profundidad. Así pues, representan una figura en el espacio y ya no en el plano.

LAS IMPRESORAS DEL SPECTRUM

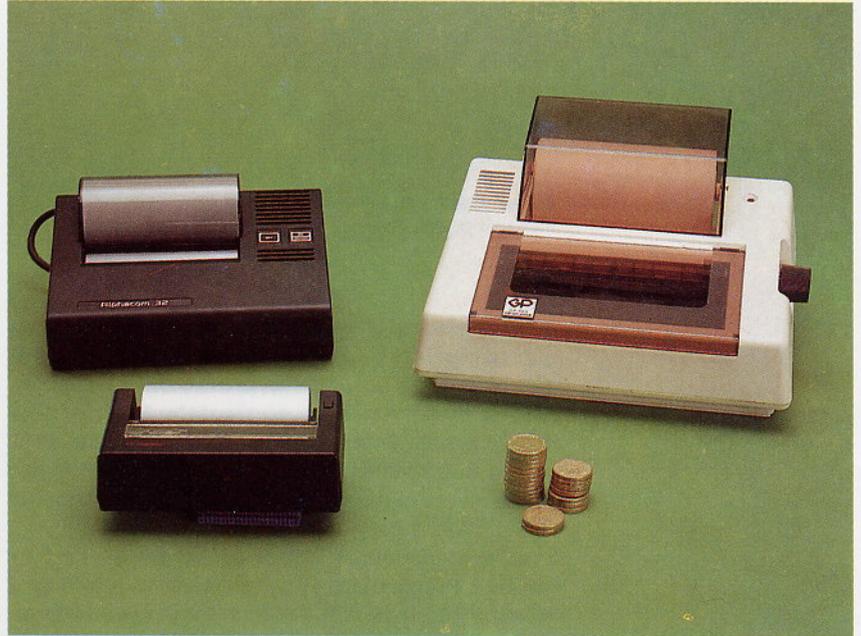


ON el paso del tiempo, entre los usuarios del Spectrum circula una idea generalizada: «A mi ordenador le falta algo». Debido a esto, tanto Sinclair, fabricante de nuestro micro, como otras firmas dedicadas a la producción de hardware, lanzan constantemente al mercado nuevos periféricos, con los que satisfacer la creciente demanda de los usuarios; eso sí, a costa de aumentar la confusión por parte del comprador, dado el elevado número de complementos disponibles. En las páginas de esta sección, tratamos de orientar a todos aquellos que en determinado momento deban tomar una decisión sobre este o aquel equipo accesorio a nuestro Spectrum, así como a los que puedan ya poseerlo, con el fin de esclarecer sus posibilidades de manejo, yendo más allá de las escuetas explicaciones que adjuntan los manuales de instrucciones que, por si además fuera poco, en muchos casos se encuentran en inglés.

Cuando tratamos el tema de los *joysticks*, vimos como, inevitablemente, fuera cual fuera el modelo elegido, debíamos disponer de un interface. Lo mismo ocurría con los discos; incluso en el capítulo dedicado como introducción a las impresoras, hacíamos hincapié en los interfaces más utilizados por estas.

Sin duda es la impresora el periférico más ampliamente difundido, entre los ordenadores a los que se les pretende encomendar una misión más allá del genocidio extraterrestre. Debido a ello, tres firmas comerciales han decidido lanzar al mercado unas impresoras que, sin necesidad de interface especial, puedan ajustarse a las necesidades básicas del usuario del Spectrum. Todas ellas con una característica común: su bajo coste. No quiere decir esto que no dispongan de interface. Este va acoplado dentro del propio equipo, y además, está diseñado para funcionar exclusivamente con nuestro Spectrum, gracias a lo cual, el precio del periférico disminuye sensiblemente. Las tres impresoras mencionadas, disponen del juego de caracteres completo de nuestro ordenador, incluyendo mayúsculas, minúsculas, caracteres gráficos, en vídeo inverso, etc. Por tanto, admiten la posibilidad de conseguir caracteres subrayados o acentuados, de la misma manera que lo haríamos para presentarlos en la pantalla del televisor o monitor.

Para su control, se utilizan los comandos LLIST y LPRINT, equivalentes a LIST y PRINT, aunque



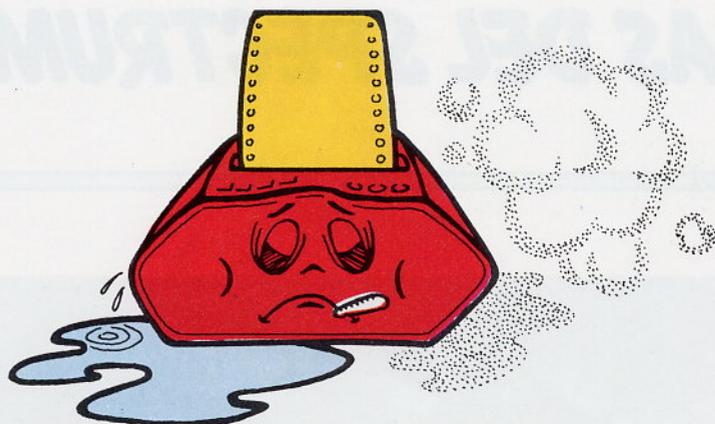
Las tres impresoras que tratamos en este capítulo, tienen en común su bajo precio.

referidos a la impresora y no a la pantalla. En este aspecto, es conveniente hacer una pequeña puntualización. La salida de información en el Spectrum, se efectúa a través de distintas corrientes; así, las corrientes 0 y 1 están asociadas a la parte inferior de la pantalla, la 2 a su parte superior, y la 3 a la impresora. Las restantes, no se encuentran asignadas a ningún canal (periférico) específico.

Según el periférico que deseemos emplear, el Spectrum se encarga de seleccionar la corriente adecuada. De esta manera, PRINT habitualmente se remite a la zona superior de la pantalla; po-

Los periféricos (CANALES) se asocian al ordenador a través de CORRIENTES.





La humedad y el polvo son los grandes enemigos de nuestros periféricos.

demos modificar este estado de cosas, simplemente indicando la corriente concreta por la cual deseamos enviar la información, mediante **PRINT # <número de corriente>**. Por tanto, la desviación de datos a la impresora, puede conseguirse con **PRINT #3**, o lo que es lo mismo: **LPRINT** es una forma abreviada de escribir **PRINT #3**.

Si queremos compatibilizar un programa, de forma que escriba sus resultados en la impresora en vez de en la pantalla, bastará con ejecutar previamente la orden **OPEN #2, "p"**, gracias a la cual, el flujo que habitualmente se envía a la pantalla (corriente 2 asociada al canal "s"), se bifurca a la impresora, tal como indica el parámetro "p" ("p" como abreviatura de *printer*) a continuación de **OPEN**. Volvamos ahora al tema fundamental del capítulo.

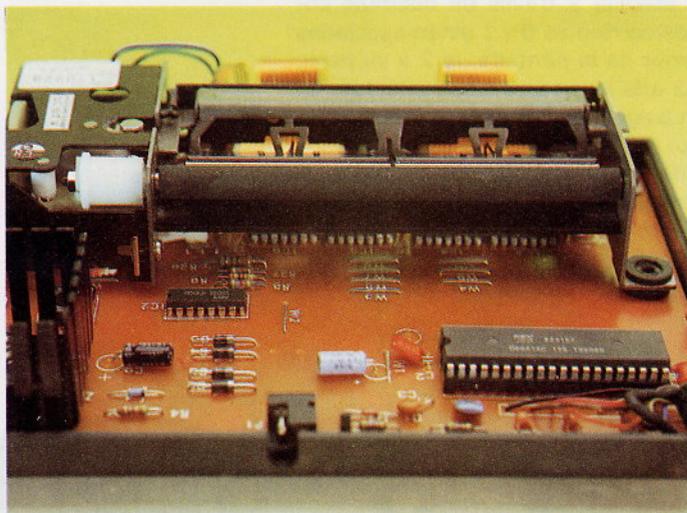
Todas las impresoras que estamos tratando implementan el comando **COPY**, el cual efectúa

una copia en papel del contenido de la pantalla, sin tener en cuenta, lógicamente, el color de la misma, sino sólo el estado del archivo de imagen, no incluyendo las dos últimas filas, reservadas por lo general al Sistema. Por otra parte, al ser periféricos especialmente diseñados para el Spectrum, todos ellos siguen el formato de impresión en treinta y dos columnas por fila.

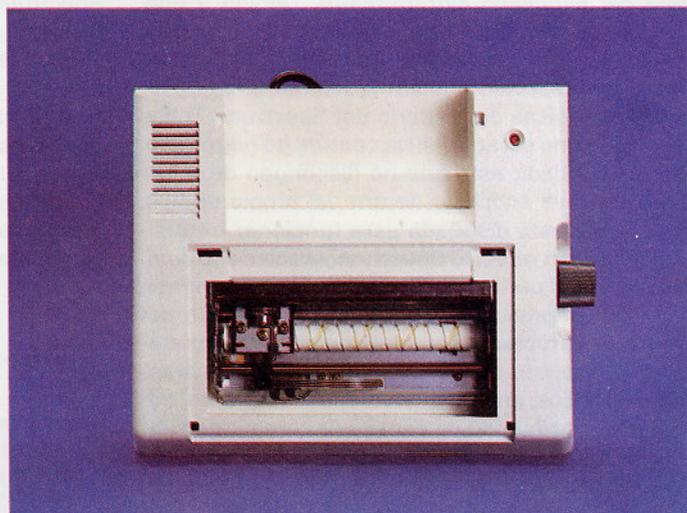
UNA CHISPA IMPRESIONANTE

Bueno, lo cierto es que no se trata de una enorme descarga eléctrica, que pueda hacer peligrar nuestra integridad física en el momento de poner en funcionamiento la impresora. Sencillamente, es el sistema empleado por la ZX PRINTER (impresora ZX) en la composición de los caracteres. Se trata de una impresora electrostática, es decir, cada punto de los que conforman un carácter, se genera al incidir sobre el papel una diminuta chispa eléctrica, producida entre una placa fija, cargada, y una pequeña aguja que se desliza a lo ancho del papel.

El papel utilizado es especial, recubierto en una de sus caras por una fina capa de compuestos metálicos, de forma que al recibir la descarga, «revela» un punto, cambiando a una tonalidad más oscura. Su anchura máxima es de 100 mm. En el caso de la ZX PRINTER, el elemento impresor no es en realidad una sola aguja, sino dos, que realizan idéntico cometido. Estas se sitúan en oposición una de la otra, sobre una pequeña



Debido a su sistema de impresión, la AlphaCom 32 pertenece al tipo denominado «de líneas».



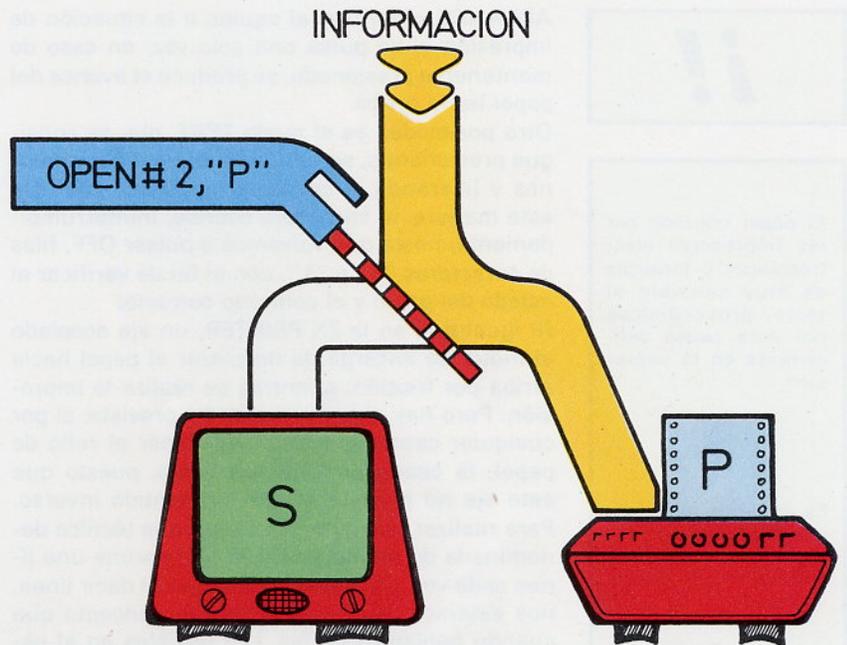
Como se puede observar en la foto, la impresora Seikosha GP-50 es del tipo matricial.

correa dentada de material plástico, apoyada sobre dos engranajes, uno de los cuales va unido al motor que suministra a la correa una velocidad de giro constante, desplazando de esta manera las agujas horizontalmente sobre el papel. Al mismo tiempo, cada vez que una de las agujas completa una pasada, un eje, igualmente engranado al motor, se encarga de desplazar hacia arriba el papel, con el fin de posicionar la siguiente línea de impresión. El sistema resulta, sin duda, ingenioso, y dada la simplicidad del mismo, los resultados, mejores de lo que cabría esperar, siempre y cuando nuestro objetivo no sea otro que el de obtener cortos listados, o sacar algún que otro comentario por impresora.

Moviéndonos en el campo de los pequeños detalles, diremos que el nivel del ruido es bajo, y en ningún caso se hace molesto. La calidad de la impresión la hace cuando menos legible, aunque se observan algunas deficiencias a la hora de trazar líneas verticales. Hoy en día, esta impresora que fue una gran ayuda, en su momento, para los pioneros del Spectrum, está prácticamente en desuso, dejando paso a otros equipos que la superan en velocidad y calidad.

CARACTERES CALIENTES

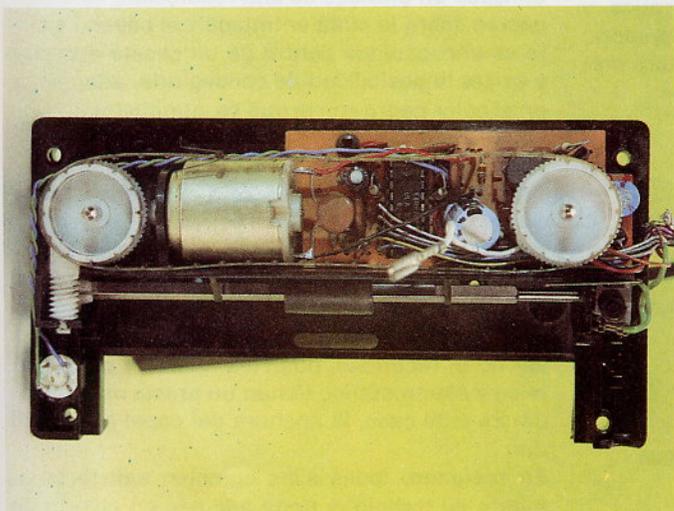
Otra posible solución económica a la hora de adquirir una impresora, la proporciona la ALPHA-COM 32. De nuevo, se trata de un aparato compacto, de fácil manejo, totalmente compatible con



El flujo de información que habitualmente discurre por PRINT hacia la pantalla (canal "s"), puede ser fácilmente desviado a la impresora con OPEN #2,"p" (canal "p").

nuestro Spectrum. Todo el equipo va encerrado dentro de una carcasa de plástico inyectado, especialmente resistente a los golpes. De él parte el conector acoplable a la tarjeta de expansión de nuestro micro. Para su funcionamiento, es necesaria una fuente de alimentación externa, suministrada junto con la impresora.

Posee un pulsador OFF, cuya misión es desactivar el paso de información hacia la impresora, aunque esto no, quiere decir que quede desconectada de la alimentación. El botón ON/



El mecanismo de impresión de la ZX Printer, se basa en dos agujas montadas en posición opuesta sobre una polea.



Cada una de las impresoras estudiadas emplea un tipo de papel diferente: normal (Seikosha GP-50), térmico (Alphacom 32) y electrostático (ZX Printer).

i!

El papel utilizado por las impresoras electrostáticas y térmicas es muy sensible al tacto, provocándose por esta causa deficiencias en la impresión.

*

Es importante seleccionar un papel de calidad adecuada para conseguir una buena impresión.

*

En las impresoras Seikosha GP-50S y Alphacom 32, debido a la distinta separación entre puntos horizontales y verticales, las imágenes reproducidas mediante COPY, se distorsionan ligeramente, alargándose en sentido vertical.

*

Las impresoras, como cualquier periférico, no deben ser conectadas al bus de expansión del Spectrum sin haber cortado previamente la alimentación a ambos aparatos.

ADVANCE, retornará el equipo a la situación de impresión si se pulsa una sola vez; en caso de mantenerse presionado, se produce el avance del papel hacia arriba.

Otra posibilidad es el modo TEST, que se consigue presionando, simultáneamente, ambos botones y liberando, a continuación, el de OFF. De este manera, la impresora escribe, ininterrumpidamente, hasta que volvamos a pulsar OFF, filas de caracteres "8" y "1", con el fin de verificar el estado del papel y el centrado correcto.

Al igual que en la ZX PRINTER, un eje acoplado al motor se encarga de desplazar el papel hacia arriba por fricción, mientras se realiza la impresión. Pero hay algo que no se ha previsto: si por cualquier causa queremos rebobinar el rollo de papel, la operación será imposible, puesto que este eje no permite el giro en sentido inverso. Para realizar la impresión, utiliza una técnica denominada de líneas, es decir, se imprime una línea cada vez. Conviene aclarar que al decir línea, nos estamos refiriendo al mismo concepto que cuando hablamos de las 175 posibles en el esquema de alta resolución del archivo de imagen. De esta manera, la velocidad de impresión aumenta considerablemente, y la ALPHACOM 32 es capaz de escribir 16 líneas enteras (dos filas de caracteres completas) cada segundo, equivalentes aproximadamente a 64 caracteres por segundo. Frente a esta rapidez considerable, el inconveniente estriba en que empleará el mismo tiempo en imprimir, por ejemplo, una sola "A" que toda una fila de éstas. No obstante, la velocidad de impresión la sitúa como la más rápida dentro de su categoría.

El papel utilizado, nuevamente, es especial. En este caso se trata de papel térmico, es decir, recubierto por una de sus caras de sustancias termosensibles, que cambian de color cuando reciben una determinada cantidad de calor. Los resultados, en lo concerniente a calidad gráfica, y teniendo en cuenta el tamaño de la impresora, pueden considerarse excelentes. La anchura má-

xima del papel utilizado, no debe superar los 109 mm.

La tecnología térmica supone otra de las virtudes del equipo, puesto que, a la vez de tener menos elementos mecánicos móviles, ventaja contra las averías, el nivel sonoro del proceso impresión, disminuye a límites bajísimos.

ECONOMICA DE IMPACTO

La SEIKOSHA GP-50S puede considerarse como la hermana pequeña, dentro de la amplia gama de impresoras matriciales, que esta firma japonesa ha comercializado en los últimos años.

Se trata de un equipo compacto, especialmente diseñado para el ZX 81 y el ZX Spectrum. También precisa una fuente de alimentación exterior, que suministra la energía necesaria para hacer funcionar el aparato. Un interruptor ON/OFF y un indicador luminoso, sirven para verificar que la impresora se ha conectado correctamente a la alimentación. En el caso de producirse algún error, la lámpara parpadeará mientras éste no sea subsanado.

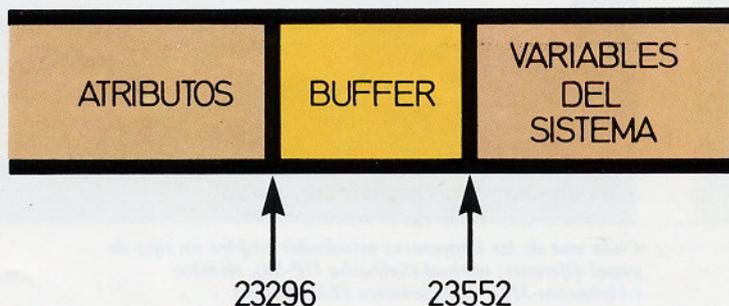
Cuando se pulsa ON, la impresora realiza una secuencia de inicialización, desplazando la cabeza de impresión dos veces a lo ancho del papel, quedando de esta manera, preparada para la escritura.

El cabezal de impresión está formado por una matriz de 7x7 pequeñas agujas, más otra adicional utilizada en gráficos de alta resolución; éstas impactan sobre la cinta entintada y el papel. La cinta va encapsulada dentro de un casete especial, y existe la posibilidad de conseguirla, además de en el color negro con el que se suministra el equipo, en distintos colores tales como el rojo, naranja, azul, verde, etc.

Por tratarse de una impresora matricial, el nivel de ruido aumenta considerablemente respecto a las anteriores. La velocidad de impresión es de 35 caracteres por segundo, y la calidad gráfica, aceptable. Tiene la ventaja de admitir papel normal, cuestión a tener en cuenta a la hora de pensar en el recambio, pues los rollos de papel térmico y electrostático tienen un precio más elevado. En este caso, la anchura del papel es de 126 mm.

En resumen, todas ellas cumplen satisfactoriamente su trabajo, y tiene además en común un bajo coste, aunque lógicamente, no debemos esperar resultados excesivamente espectaculares.

La zona de memoria que el Spectrum emplea como intermediario entre la impresora y el ordenador, se denomina buffer de la impresora.



EL CUBO MAGICO

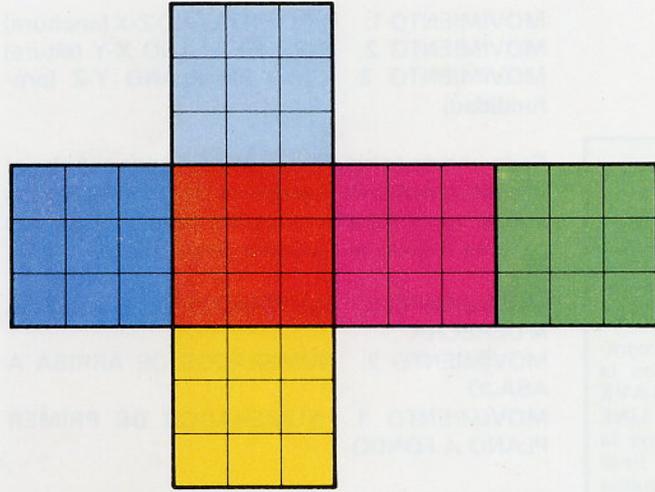
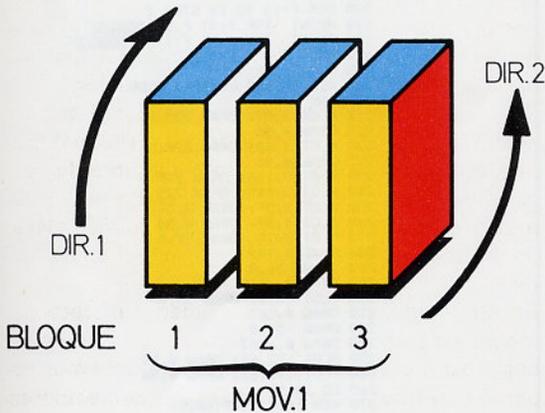


SEGURAMENTE, en alguna ocasión hemos tenido en nuestras manos el famoso CUBO DE RUBIK. Hemos dado vueltas y más vueltas a cada uno de los bloques en que está dividido, y solamente en contadas ocasiones hemos logrado reordenar satisfactoriamente los nueve cuadritos que componen cada una de las seis caras. Con este nuevo programa, tendremos plena garantía de alejar el aburrimiento.

Aunque este juego ha hecho auténtico furor entre la juventud europea, no daremos por supuesto que las nuevas generaciones conocen completamente su funcionamiento, y brindaremos unas breves explicaciones sobre el objetivo de este pasatiempo. El protagonista del juego es un cubo móvil (seis caras), compuesto por nueve cuadritos de color por cada cara; todas las caras pueden girar sobre sí mismas, en el sentido horario o antihorario.

Cada una de las caras está compuesta por tanto por nueve bloquecitos de un mismo color (por supuesto diferente en cada una de ellas), que se pueden desplazar mediante un giro. El cubo inicia el juego con una zona de cada color, posteriormente se desordena de una manera aleatoria, con varios movimientos, y acto seguido, se puede pasar a ordenarlo nuevamente. Aquí llega el problema: desordenarlo es bien fácil, pero girar las caras adecuadas para que todos los colores vuelvan a quedar en su lugar, no es precisamente sencillo.

El primer movimiento se considera en anchura.



En el desarrollo del cubo que se presenta en la pantalla, la cara roja coincide con la superior.

EL CUBO DEL SPECTRUM

La representación en la pantalla de una figura con volumen es evidentemente muy compleja, de ahí que hayamos escogido como solución, representar el cubo en desarrollo, es decir, en forma de cruz, tal como se muestra en la figura. Esto implica, que la cara roja representa la parte superior, la azul clara la posterior, la amarilla la anterior, la verde la inferior, la magenta la derecha y la azul oscura la izquierda.

Una vez conocida la situación del cubo, podemos comenzar el programa. Como primera medida se nos solicita la elección del nivel de dificultad, el cual consiste en el número de movimientos aleatorios a que se someterá el cubo en su fase de desorden. Una vez escogido el nivel, el cubo se presentará en la pantalla, y se desordenará, para a continuación pasar a la fase de resolución del problema por parte del jugador.

La codificación de nuestra jugada, es decir, la ma-

nera en que vamos a indicar al Spectrum el movimiento que queremos realizar, es bastante fácil de aprender, y pronto nos acostumbraremos a ella. Lo más práctico será que sigamos las indicaciones que se brindan a continuación, auxiliados por los gráficos explicativos.

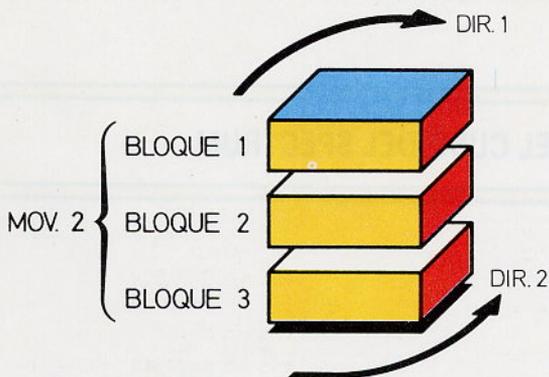
En primer lugar hemos de indicar el tipo de movimiento a efectuar. Este dependerá del plano en que deseamos realizar la jugada:

- MOVIMIENTO 1. GIRO EN PLANO Z-X (anchura)
- MOVIMIENTO 2. GIRO EN PLANO X-Y (altura)
- MOVIMIENTO 3. GIRO EN PLANO Y-Z (profundidad)

Cada uno de estos movimientos, podría señalar el desplazamiento de cualquiera de los tres bloques que se van a ver afectados. Así pues, el siguiente paso es indicar el bloque a desplazar (de 1 a 3):

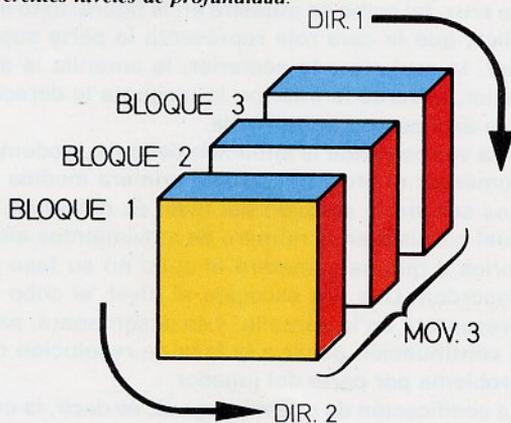
- MOVIMIENTO 1. NUMERADOS DE IZQUIERDA A DERECHA
- MOVIMIENTO 2. NUMERADOS DE ARRIBA A ABAJO
- MOVIMIENTO 3. NUMERADOS DE PRIMER PLANO A FONDO

Finalmente, sólo queda especificar el sentido de giro del bloque escogido. Y en este caso, se cum-



El movimiento dos se efectúa en las diferentes alturas.

El tercer tipo de movimiento se lleva a cabo en los diferentes niveles de profundidad.



ple indudablemente que una imagen vale más que cien palabras, de manera que lo más sencillo será que observemos los gráficos adjuntos, donde se detalla mediante una flecha, qué sentido es el 1 y cuál el 2, según cada movimiento. Por cada jugada que ejecutamos, el ordenador efectúa una considerable cantidad de operaciones: calcula el efecto del movimiento, actualiza las seis tablas que tiene definidas, controla si el cubo ha sido solucionado, incrementa el contador de jugadas y, finalmente, muestra el movimiento especificado.

Esta serie de tareas, añadido a la estructura del programa, en forma de subrutinas, hace que el desplazamiento del cubo no sea inmediato, pero sin duda lo suficientemente ágil para lo que se pretende, sobre todo teniendo en cuenta que en este juego se confirma un conocido refrán: «Más vale maña que fuerza». Efectivamente, en el juego original siempre nos cabía la posibilidad de desmontar el cubo y volver a montarlo ordenado, sin embargo ahora...

La introducción del programa, sólo tiene el problema de las letras subrayadas que aparecen en el listado; como ya sabemos, éstas corresponden a los caracteres gráficos de las teclas afectadas. A modo orientativo, señalaremos que dichos caracteres se encuentran en las siguientes líneas: 80, 90, 130, 140, 170, 180, 6875, 6880, 6885, 6890, 6895, 6900, 6905, 6910, 6945, 6950, 6955 y 6960. Para la grabación del programa, utilizaremos el comando **SAVE "CUBO MAG."** (sin autoejecución), o bien **SAVE "CUBO MAG." LINE 1** (con autoejecución a partir de la primera línea).



```

10 REM *****
20 REM * J.M.MAYORAL SERRANO *
30 REM *****
40 GO SUB 5000
42 BORDER 0: PAPER 0: INK 9: POKE 23658,
8
45 CLS : GO SUB 7100
46 GO SUB 5125: GO TO 200
50 REM PRESENTACION PANTALLA
60 FOR N=1 TO 4
70 FOR F=8 TO 12 STEP 2
80 PRINT INK N;AT F,7*(N-1)+1;"ABABAB"
90 PRINT INK N;AT F+1,7*(N-1)+1;"COCDCO"

100 NEXT F
110 NEXT N
120 FOR F=1 TO 5 STEP 2
130 PRINT INK 5;AT F,8;"ABABAB"
140 PRINT INK 5;AT F+1,8;"COCDCO"
150 NEXT F
160 FOR F=15 TO 19 STEP 2
170 PRINT INK 6;AT F,8;"ABABAB"
180 PRINT INK 6;AT F+1,8;"COCDCO"
190 NEXT F
195 GO TO 370
200 REM DIBUJO LIMITES CUBO
205 INK 9
210 PLOT 3,59: DRAW 225,0
215 DRAW 0,56
220 PLOT 3,60: DRAW 224,0
225 DRAW 0,56
230 PLOT 3,116: DRAW 225,0
240 PLOT 3,115: DRAW 225,0
250 PLOT 3,59: DRAW 0,56
260 PLOT 4,59: DRAW 0,56
270 PLOT 59,3: DRAW 57,0
280 DRAW 0,169
290 DRAW -56,0
300 DRAW 0,-168
310 PLOT 60,4: DRAW 55,0
320 DRAW 0,167
330 DRAW -54,0
340 DRAW 0,-167
350 PLOT 172,61: DRAW 0,54
360 PLOT 171,61: DRAW 0,54
365 GO TO 50
370 REM INIC. VARIABLES
    
```

i!

Para grabar el programa, utilizaremos la instrucción **SAVE "CUBO MAG." LINE 1**. Si suprimimos la especificación final **LINE 1**, el programa no se autoejecutará al finalizar la carga, aunque podrá ser lanzado mediante **RUN**, sin ninguna dificultad.

*

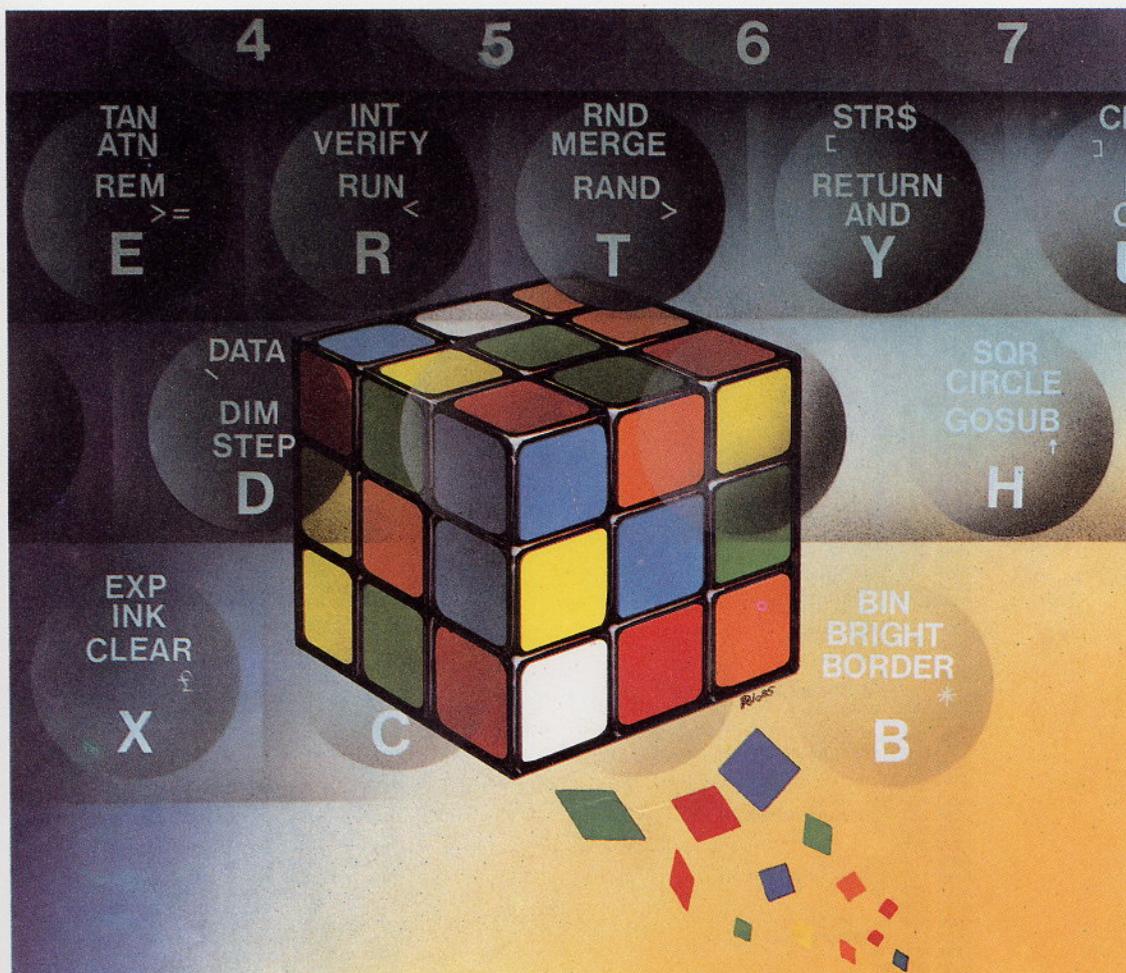
Al introducir el programa, no olvidemos sustituir los caracteres subrayados, por los gráficos definidos correspondientes a las teclas indicadas.

*

A la hora de efectuar una jugada, hemos de introducir el número correspondiente a: tipo de movimiento, número de bloque y dirección en la cual dicho bloque va a desplazarse.



PROGRAMA



```

375 DIM Z(3,3): DIM R(3,3)
380 DIM M(3,3): DIM V(3,3)
385 DIM C(3,3): DIM A(3,3)
390 DIM P(3): LET CONTA=0
395 GO SUB 6770: GO SUB 7137
699 REM INTROD. DATOS
700 PRINT AT 17,16: FLASH 1: PAPER 2: 'MOV
IMIENTO?'
705 FOR N=1 TO 3
710 PRINT PAPER 6:AT 19,17+N*2;N
715 NEXT N
720 LET K$=INKEY$
725 IF CODE K$<49 OR CODE K$>51 THEN GO
TO 720
730 BEEP .2,40: BEEP .2,38
735 LET MOV=VAL K$
740 GO SUB 5195
745 PRINT AT 17,17: FLASH 1: PAPER 4: 'BL
OQUE ?'
750 FOR N=1 TO 3
755 PRINT PAPER 7:AT 19,17+N*2;N
760 NEXT N
765 LET K$=INKEY$
770 IF CODE K$<49 OR CODE K$>51 THEN GO
TO 765
775 BEEP .2,40: BEEP .2,38
780 LET BLOCK=VAL K$
785 GO SUB 5195
790 PRINT AT 17,17: FLASH 1: PAPER 5: 'DIR
ECCION ?'
795 FOR N=1 TO 2
800 PRINT PAPER 2:AT 19,19+N*2;N
805 NEXT N
810 LET K$=INKEY$
815 IF CODE K$<49 OR CODE K$>50 THEN GO
TO 810
820 BEEP .2,40: BEEP .2,38
825 LET DIR=VAL K$
830 GO SUB 5195
835 LET FORM=4-BLOCK
840 REM PROGRAMA PRINCIPAL
845 LET CONTA=CONTA+1
850 GO SUB 6980
900 GO SUB 6860
910 GO SUB 6830
920 GO TO 700
5000 REM GRAFICOS USUARIO

```

```

5010 DATA 0,127,127,127,127,127,127,127
5020 DATA 0,254,254,254,254,254,254,254
5030 DATA 127,127,127,127,127,127,127,0
5040 DATA 254,254,254,254,254,254,254,0
5042 DATA 74,32,77,32,77,65,89,79,82,65,76
5043 DATA 32,83,69,82,82,65,78,79
5060 LET A$='ABCD'
5070 FOR N=1 TO LEN A$
5080 FOR K=0 TO 7
5090 READ A
5100 POKE USR A$(N)+K,A
5110 NEXT K
5120 NEXT N
5122 RETURN
5125 CLS : PRINT AT 0,0: PRINT
5130 FOR N=0 TO 18
5140 READ B
5150 PRINT INK 5; BRIGHT 1:TAB 30:CHR# B
5160 NEXT N
5170 PRINT INK 5; BRIGHT 0:AT 2,31:'. '
5180 PRINT INK 5:AT 4,31:'. '
5190 RETURN
5195 REM SBR. BORRADO PARCIAL
5200 FOR F=17 TO 19 STEP 2
5205 FOR C=16 TO 28
5210 PRINT AT F,C:'. '
5215 NEXT C
5220 NEXT F
5225 RETURN
5226 REM
5230 REM SBR - 1
5231 REM
5235 REM PASO A PUENTE
5240 FOR M=1 TO 3
5245 LET P(M)=A(M,BLOCK)
5250 NEXT M
5255 REM PASO V a A (3 a 4)
5260 LET X=3
5265 FOR M=1 TO 3
5270 LET A(M,BLOCK)=V(X,FORM)
5275 LET X=X-1
5280 NEXT M
5285 REM PASO C a V (2 a 3)
5290 LET X=3
5295 FOR M=1 TO 3
5300 LET V(M,FORM)=C(X,BLOCK)
5305 LET X=X-1

```

```

5310 NEXT M
5315 REM PASO R a C (1 a 2)
5320 FOR M=1 TO 3
5325 LET C(M,BLOCK)=R(M,BLOCK)
5330 NEXT M
5331 REM PASO PUENTE a R
5332 FOR M=1 TO 3
5333 LET R(M,BLOCK)=P(M)
5334 NEXT M
5335 RETURN
5340 REM SBR - 2
5345 REM MUEVE VERTICES
5350 LET P(1)=Z(1,3)
5355 LET Z(1,3)=Z(3,3)
5360 LET Z(3,3)=Z(3,1)
5365 LET Z(3,1)=Z(1,1)
5370 LET Z(1,1)=P(1)
5375 REM MUEVE ADYACENTES CENTR
5380 LET P(1)=Z(2,3)
5385 LET Z(2,3)=Z(3,2)
5390 LET Z(3,2)=Z(2,1)
5395 LET Z(2,1)=Z(1,2)
5400 LET Z(1,2)=P(1)
5405 RETURN
5410 REM SBR - 3
5415 REM MUEVE VERTICES
5420 LET P(1)=M(1,3)
5425 LET M(1,3)=M(1,1)
5430 LET M(1,1)=M(3,1)
5435 LET M(3,1)=M(3,3)
5440 LET M(3,3)=P(1)
5445 REM MUEVE ADYACENTES CENTR
5450 LET P(1)=M(2,3)
5455 LET M(2,3)=M(1,2)
5460 LET M(1,2)=M(2,1)
5465 LET M(2,1)=M(3,2)
5470 LET M(3,2)=P(1)
5475 RETURN
5480 REM SBR - 4
5485 REM
5490 FOR M=1 TO 3
5495 LET P(M)=C(M,BLOCK)
5500 NEXT M
5505 REM M
5507 LET X=3
5510 FOR M=1 TO 3
5515 LET C(M,BLOCK)=V(X,FORM)

```

```

5520 LET X=X-1
5525 NEXT M
5530 REM
5535 LET X=3
5540 FOR M=1 TO 3
5545 LET V(M,FORM)=A(X,BLOCK)
5550 LET X=X-1
5555 NEXT M
5560 REM
5565 FOR M=1 TO 3
5570 LET A(M,BLOCK)=R(M,BLOCK)
5575 NEXT M
5580 REM
5585 FOR M=1 TO 3
5590 LET R(M,BLOCK)=P(M)
5595 NEXT M
5600 RETURN
5605 REM SBR - 5
5610 REM
5615 LET P(1)=Z(1,3)
5620 LET Z(1,3)=Z(1,1)
5625 LET Z(1,1)=Z(3,1)
5630 LET Z(3,1)=Z(3,3)
5635 LET Z(3,3)=P(1)
5640 REM
5645 LET P(1)=Z(2,3)
5650 LET Z(2,3)=Z(1,2)
5655 LET Z(1,2)=Z(2,1)
5660 LET Z(2,1)=Z(3,2)
5665 LET Z(3,2)=P(1)
5670 RETURN
5675 REM SBR - 6
5680 REM
5685 LET P(1)=M(1,3)
5690 LET M(1,3)=M(3,3)
5695 LET M(3,3)=M(3,1)
5700 LET M(3,1)=M(1,1)
5705 LET M(1,1)=P(1)
5710 REM
5715 LET P(1)=M(2,3)
5720 LET M(2,3)=M(3,2)
5725 LET M(3,2)=M(2,1)
5730 LET M(2,1)=M(1,2)
5735 LET M(1,2)=P(1)
5740 RETURN
5745 REM SBR - 7
5750 REM

```

PROGRAMA

```

5755 FOR M=1 TO 3
5760 LET P(M)=Z(M,FORM)
5765 NEXT M
5770 REM
5775 LET X=1
5780 FOR M=1 TO 3
5785 LET Z(M,FORM)=A(BLOCK,X)
5790 LET X=X+1
5795 NEXT M
5800 REM
5805 LET X=3
5810 FOR M=1 TO 3
5815 LET A(BLOCK,M)=M(X,BLOCK)
5820 LET X=X-1
5825 NEXT M
5830 REM
5835 LET X=1
5840 FOR M=1 TO 3
5845 LET M(M,BLOCK)=C(FORM,X)
5850 LET X=X+1
5855 NEXT M
5860 REM
5865 LET X=3
5870 FOR M=1 TO 3
5875 LET C(FORM,M)=P(X)
5880 LET X=X-1
5885 NEXT M
5890 RETURN
5895 REM SBR - 8
5900 LET P(1)=R(1,1)
5905 LET R(1,1)=R(3,1)
5910 LET R(3,1)=R(3,3)
5915 LET R(3,3)=R(1,3)
5920 LET R(1,3)=P(1)
5925 REM
5930 LET P(1)=R(1,2)
5935 LET R(1,2)=R(2,1)
5940 LET R(2,1)=R(3,2)
5945 LET R(3,2)=R(2,3)
5950 LET R(2,3)=P(1)
5955 RETURN
5960 REM SBR - 9
5965 LET P(1)=V(1,1)
5970 LET V(1,1)=V(1,3)
5975 LET V(1,3)=V(3,3)
5980 LET V(3,3)=V(3,1)
5985 LET V(3,1)=P(1)
5990 REM
5995 LET P(1)=V(1,2)
6000 LET V(1,2)=V(2,3)
6005 LET V(2,3)=V(3,2)
6010 LET V(3,2)=V(2,1)
6015 LET V(2,1)=P(1)
6020 RETURN
6025 REM SBR - 10
6030 REM
6035 FOR M=1 TO 3
6040 LET P(M)=M(M,BLOCK)
6045 NEXT M
6050 REM
6055 LET X=3
6060 FOR M=1 TO 3
6065 LET M(M,BLOCK)=A(BLOCK,X)
6070 LET X=X-1
6075 NEXT M
6080 REM
6085 LET X=1
6090 FOR M=1 TO 3
6095 LET A(BLOCK,M)=Z(X,FORM)
6100 LET X=X+1
6105 NEXT M
6110 REM
6115 LET X=3
6120 FOR M=1 TO 3
6125 LET Z(M,FORM)=C(FORM,X)
6130 LET X=X-1
6135 NEXT M
6140 REM
6145 FOR M=1 TO 3
6150 LET C(FORM,M)=P(M)
6155 NEXT M
6160 RETURN
6165 REM SBR - 11
6170 LET P(1)=R(1,1)
6175 LET R(1,1)=R(1,3)
6180 LET R(1,3)=R(3,3)
6185 LET R(3,3)=R(3,1)
6190 LET R(3,1)=P(1)
6195 REM
6200 LET P(1)=R(1,2)
6205 LET R(1,2)=R(2,3)
6210 LET R(2,3)=R(3,2)
6215 LET R(3,2)=R(2,1)
6220 LET R(2,1)=P(1)
6225 RETURN
6230 REM SBR - 12
6235 LET P(1)=V(1,1)
6240 LET V(1,1)=V(3,1)
6245 LET V(3,1)=V(3,3)
6250 LET V(3,3)=V(1,3)
6255 LET V(1,3)=P(1)
6260 REM
6265 LET P(1)=V(1,2)
6270 LET V(1,2)=V(2,1)
6275 LET V(2,1)=V(3,2)
6280 LET V(3,2)=V(2,3)
6285 LET V(2,3)=P(1)
6290 RETURN
6295 REM SBR - 13
6300 FOR M=1 TO 3
6305 LET P(M)=V(FORM,M)
6310 NEXT M
6315 REM

```

```

6320 FOR M=1 TO 3
6325 LET V(FORM,M)=M(FORM,M)
6330 NEXT M
6335 REM
6340 FOR M=1 TO 3
6345 LET M(FORM,M)=R(FORM,M)
6350 NEXT M
6355 REM
6360 FOR M=1 TO 3
6365 LET R(FORM,M)=Z(FORM,M)
6370 NEXT M
6375 REM
6380 FOR M=1 TO 3
6385 LET Z(FORM,M)=P(M)
6390 NEXT M
6395 RETURN
6400 REM SBR - 14
6405 LET P(1)=A(1,1)
6410 LET A(1,1)=A(3,1)
6415 LET A(3,1)=A(3,3)
6420 LET A(3,3)=A(1,3)
6430 LET A(1,3)=P(1)
6435 REM
6440 LET P(1)=A(1,2)
6445 LET A(1,2)=A(2,1)
6450 LET A(2,1)=A(3,2)
6455 LET A(3,2)=A(2,3)
6460 LET A(2,3)=P(1)
6465 RETURN
6470 REM SBR - 15
6475 LET P(1)=C(1,1)
6480 LET C(1,1)=C(1,3)
6485 LET C(1,3)=C(3,3)
6490 LET C(3,3)=C(3,1)
6495 LET C(3,1)=P(1)
6500 REM
6505 LET P(1)=C(1,2)
6510 LET C(1,2)=C(2,3)
6515 LET C(2,3)=C(3,2)
6520 LET C(3,2)=C(2,1)
6525 LET C(2,1)=P(1)
6530 RETURN
6535 REM SBR - 16
6540 FOR M=1 TO 3
6545 LET P(M)=Z(FORM,M)
6550 NEXT M
6555 REM
6560 FOR M=1 TO 3
6565 LET Z(FORM,M)=R(FORM,M)
6570 NEXT M
6575 REM
6580 FOR M=1 TO 3
6585 LET R(FORM,M)=M(FORM,M)
6590 NEXT M
6595 REM
6600 FOR M=1 TO 3
6605 LET M(FORM,M)=V(FORM,M)
6610 NEXT M
6615 REM
6620 FOR M=1 TO 3
6625 LET V(FORM,M)=P(M)
6630 NEXT M
6635 RETURN
6640 REM SBR - 17
6645 LET P(1)=A(1,1)
6650 LET A(1,1)=A(1,3)
6655 LET A(1,3)=A(3,3)
6660 LET A(3,3)=A(3,1)
6665 LET A(3,1)=P(1)
6670 REM
6675 LET P(1)=A(1,2)
6680 LET A(1,2)=A(2,3)
6685 LET A(2,3)=A(3,2)
6690 LET A(3,2)=A(2,1)
6695 LET A(2,1)=P(1)
6700 RETURN
6705 REM SBR - 18
6710 LET P(1)=C(1,1)
6715 LET C(1,1)=C(3,1)
6720 LET C(3,1)=C(3,3)
6725 LET C(3,3)=C(1,3)
6730 LET C(1,3)=P(1)
6735 REM
6740 LET P(1)=C(1,2)
6745 LET C(1,2)=C(2,1)
6750 LET C(2,1)=C(3,2)
6755 LET C(3,2)=C(2,3)
6760 LET C(2,3)=P(1)
6765 RETURN
6770 REM INIC. TABLAS
6775 FOR P=1 TO 3
6780 FOR Q=1 TO 3
6785 LET Z(P,Q)=1
6790 LET R(P,Q)=2
6795 LET M(P,Q)=3
6800 LET V(P,Q)=4
6805 LET C(P,Q)=5
6810 LET A(P,Q)=6
6815 NEXT Q
6820 NEXT P
6825 RETURN
6830 REM CUBO COORECTO ?
6835 GO SUB 7900
6840 GO TO 700
6845 REM DISPLAY CARAS/CUBO
6850 FOR F=1 TO 3
6855 FOR C=1 TO 3
6860 PRINT INK Z(F,C);AT 2*F+6,2*C-1; "AB"
6865 PRINT INK Z(F,C);AT 2*F+7,2*C-1; "CD"
6870 PRINT INK R(F,C);AT 2*F+6,2*C+6; "AB"
6875 PRINT INK R(F,C);AT 2*F+7,2*C+6; "CD"
6880 PRINT INK M(F,C);AT 2*F+6,2*C+13; "AB"
6885 PRINT INK M(F,C);AT 2*F+7,2*C+13; "CD"
6890 PRINT INK V(F,C);AT 2*F+6,2*C+6; "AB"
6895 PRINT INK V(F,C);AT 2*F+7,2*C+6; "CD"
6900 PRINT INK C(F,C);AT 2*F+7,2*C+20; "AB"
6905 PRINT INK C(F,C);AT 2*F+6,2*C+20; "AB"
6910 PRINT INK V(F,C);AT 2*F+7,2*C+20; "CD"
6915 NEXT C
6920 NEXT F
6940 FOR C=1 TO 3
6945 PRINT INK C(F,C);AT 2*F-1,2*C+6; "AB"
6950 PRINT INK C(F,C);AT 2*F,2*C+6; "CD"
6955 PRINT INK A(F,C);AT 2*F+13,2*C+6; "AB"
6960 PRINT INK A(F,C);AT 2*F+14,2*C+6; "CD"
6965 NEXT C
6970 NEXT F
6975 RETURN
6980 REM SBR CONTROLA MOVIMIENTOS.
6985 IF MOV=1 AND BLOCK=1 AND DIR=1 THEN
GO SUB 5230: GO SUB 5340: RETURN
6990 IF MOV=1 AND BLOCK=1 AND DIR=2 THEN
GO SUB 5480: GO SUB 5605: RETURN
7000 IF MOV=1 AND BLOCK=2 AND DIR=1 THEN
GO SUB 5230: RETURN
7005 IF MOV=1 AND BLOCK=2 AND DIR=2 THEN
GO SUB 5480: RETURN
7010 IF MOV=1 AND BLOCK=3 AND DIR=1 THEN
GO SUB 5230: GO SUB 5410: RETURN
7015 IF MOV=1 AND BLOCK=3 AND DIR=2 THEN
GO SUB 5480: GO SUB 5675: RETURN
7020 REM
7025 REM
7030 IF MOV=2 AND BLOCK=1 AND DIR=1 THEN
GO SUB 5750: GO SUB 5895: RETURN
7035 IF MOV=2 AND BLOCK=1 AND DIR=2 THEN
GO SUB 6025: GO SUB 6165: RETURN
7040 IF MOV=2 AND BLOCK=2 AND DIR=1 THEN
GO SUB 5750: RETURN
7045 IF MOV=2 AND BLOCK=2 AND DIR=2 THEN
GO SUB 6025: RETURN
7050 IF MOV=2 AND BLOCK=3 AND DIR=1 THEN
GO SUB 5750: GO SUB 5960: RETURN
7055 IF MOV=2 AND BLOCK=3 AND DIR=2 THEN
GO SUB 6025: GO SUB 6230: RETURN
7060 REM
7065 REM
7070 IF MOV=3 AND BLOCK=1 AND DIR=1 THEN
GO SUB 6295: GO SUB 6400: RETURN
7075 IF MOV=3 AND BLOCK=1 AND DIR=2 THEN
GO SUB 6535: GO SUB 6640: RETURN
7080 IF MOV=3 AND BLOCK=2 AND DIR=1 THEN
GO SUB 6295: RETURN
7085 IF MOV=3 AND BLOCK=2 AND DIR=2 THEN
GO SUB 6535: RETURN
7090 IF MOV=3 AND BLOCK=3 AND DIR=1 THEN
GO SUB 6295: GO SUB 6470: RETURN
7095 IF MOV=3 AND BLOCK=3 AND DIR=2 THEN
GO SUB 6535: GO SUB 6705: RETURN
7100 REM NIVEL DE DIFICULTAD
7105 CLS
7110 PRINT AT 10,2: PAPER 6; INK 1; "INTRODUCE NIVEL DE DIFICULTAD"
7115 PRINT AT 12,11; "1 -a- 9"
7120 LET K$=INKEY$
7125 IF CODE K$<49 OR CODE K$>57 THEN GO TO 7120
7130 BEEP .2,40
7135 LET NI=2*VAL K$: RETURN
7137 REM DESORDENACION
7138 PRINT FLASH 1;AT 2,15: PAPER 2; " ;AT 4,15;
7139 PRINT AT 3,15: PAPER 2; FLASH 1; " DESORDENANDO
7140 FOR G=1 TO NI
7145 LET MOV=INT (RND*3)+1
7150 LET BLOCK=INT (RND*3)+1
7155 LET DIR=INT (RND*2)+1
7160 LET FORM=4-BLOCK
7165 BEEP .1,40: GO SUB 6980
7170 NEXT G
7175 BEEP .1,40: BEEP .1,40
7180 FOR F=2 TO 4
7185 PRINT AT F,15; "
7190 NEXT F
7195 GO SUB 6860
7195 RETURN
7200 REM COMPROBACION
7210 LET C1=Z(1,1)
7215 LET C2=R(1,1)
7220 LET C3=M(1,1)
7225 LET C4=V(1,1)
7230 LET C5=C(1,1)
7235 LET C6=A(1,1)
7240 FOR F=1 TO 3
7245 IF Z(F,C)=C1 AND R(F,C)=C2 AND M(F,C)=C3 AND V(F,C)=C4 AND C(F,C)=C5 AND A(F,C)=C6 THEN GO TO 7990
7250 RETURN
7255 NEXT F
7260 REM CUBO ACERTADO
7265 PRINT FLASH 1; PAPER 7; INK 9; AT 1,1 5; "LO CONSEGUISTE"
7270 PRINT PAPER 6; INK 2; AT 3,16: CONTA: PAPER 0; INK 9; " MOVIMIENTOS"
7275 PRINT AT 16,15; "OTRA PARTIDA ?"
7280 PRINT AT 18,19; "(S/N)"
7285 IF INKEY$="S" THEN RUN
7290 IF INKEY$="N" THEN GO TO 10000
7295 GO TO 8050

```