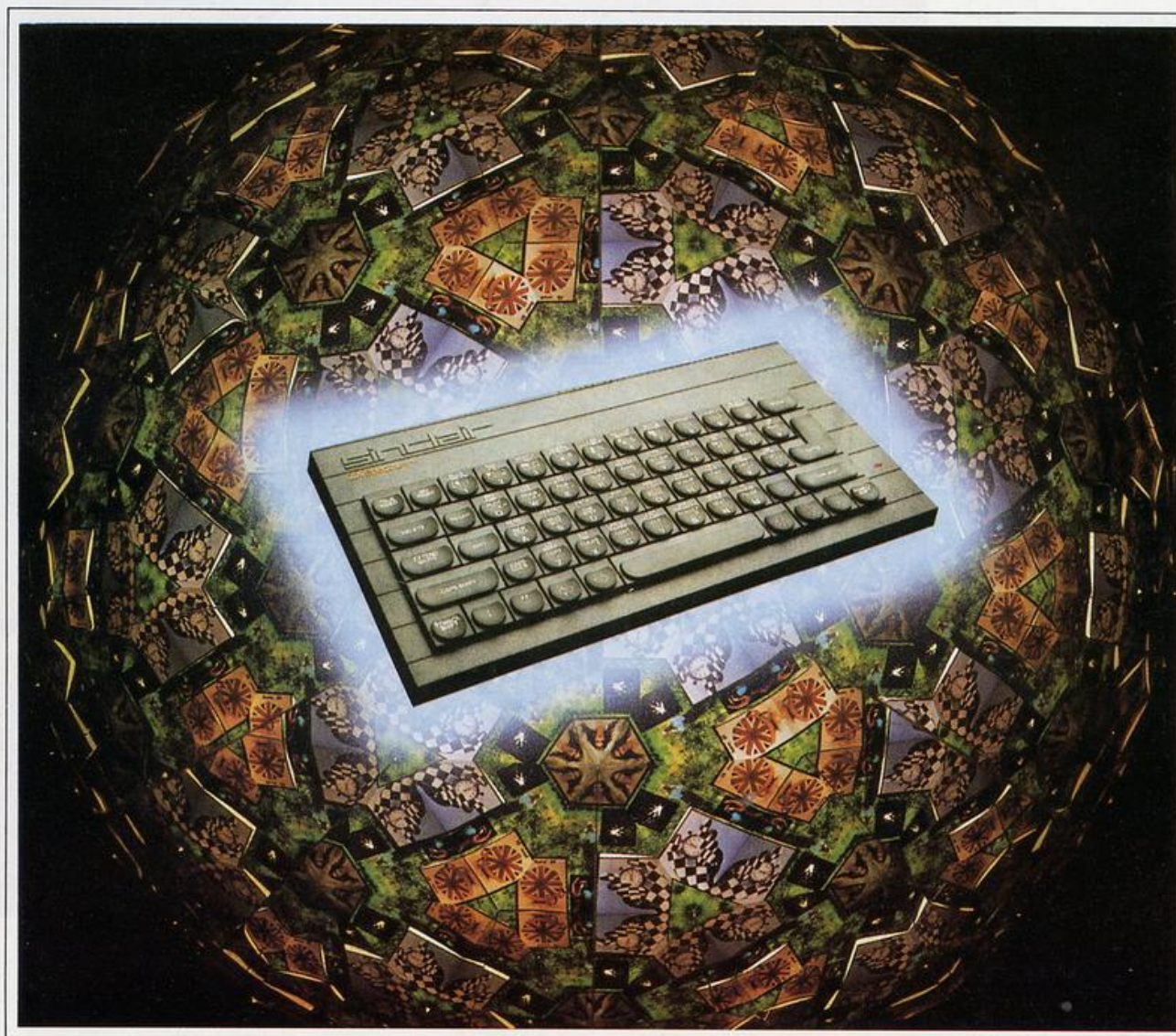


41  
185 pts.  
IVA INCLUIDO

# PU

## Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek

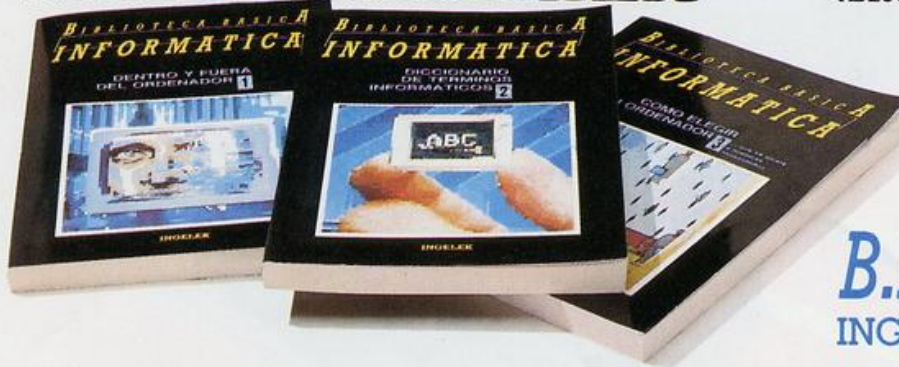




# UNA GRAN OBRA A SU ALCANCE



UNA OBRA COMPLETISIMA EN 30 VOLUMENES QUE TRATA TODOS LOS TEMAS, DESDE QUE ES UN ORDENADOR HASTA EL ESTUDIO DE LOS DIVERSOS LENGUAJES, PASANDO POR LOS LENGUAJES, METODOS DE PROGRAMACION, ELECCION DEL ORDENADOR ADECUADO, DICCIONARIO, ETC.



**B.B.I.**  
INGELEK

## 30 EXTRAORDINARIOS VOLUMENES DE APARICION SEMANAL CON TODOS LOS CONCEPTOS DE LA INFORMATICA

GRAN OFERTA DE SUSCRIPCION  
9.995 PTAS

AHORRE MAS DE 1.000 PTAS Y LLEVESE UNA MAGNIFICA CALCULADORA SOLAR  
VALORADA EN 2.500 PTAS.



OFERTA VALIDA UNICAMENTE  
PARA ESPAÑA

## SUSCRIBASE POR TELEFONO

Todos los días, excepto sábados y festivos,  
de 8 a 6,30 atenderemos sus consultas en el



# 2505820



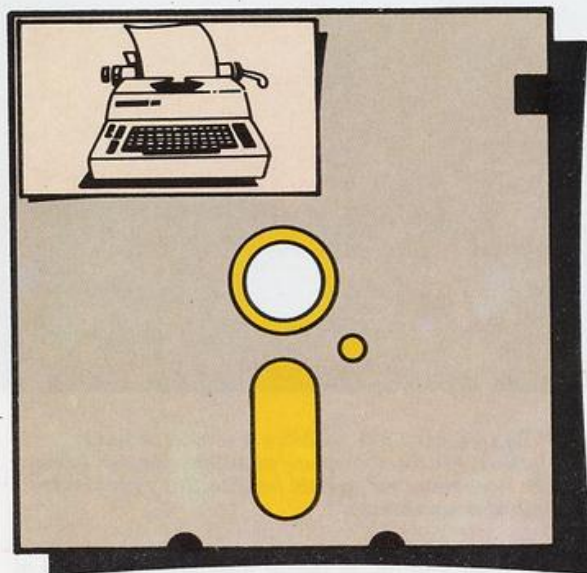
# PROCESADORES DE TEXTO



NICIAMOS con este capítulo un ciclo dedicado al tratamiento de las aplicaciones comerciales más usuales en cualquier ordenador. Nada más propio en una obra de estas características que comenzar esta serie de artículos tratando un programa destinado precisamente a la transcripción escrita de documentos: los procesadores de textos.

El Spectrum es sin duda uno de los ordenadores que dispone en la actualidad de una mayor variedad de programas, no sólo versando sobre temáticas muy diferentes, sino también ofreciendo una gran variedad en aplicaciones con objetivos muy similares; sin embargo, dentro de los denominados procesadores de textos, este fenómeno no se produce.

Un magnífico programa: TASWORD TWO ha venido a satisfacer completamente las necesidades del usuario del Spectrum en el ámbito de los procesadores de textos, no dejando opción al nacimiento de ningún otro programa en este sentido, por ser prácticamente imposible aportar mayores prestaciones.

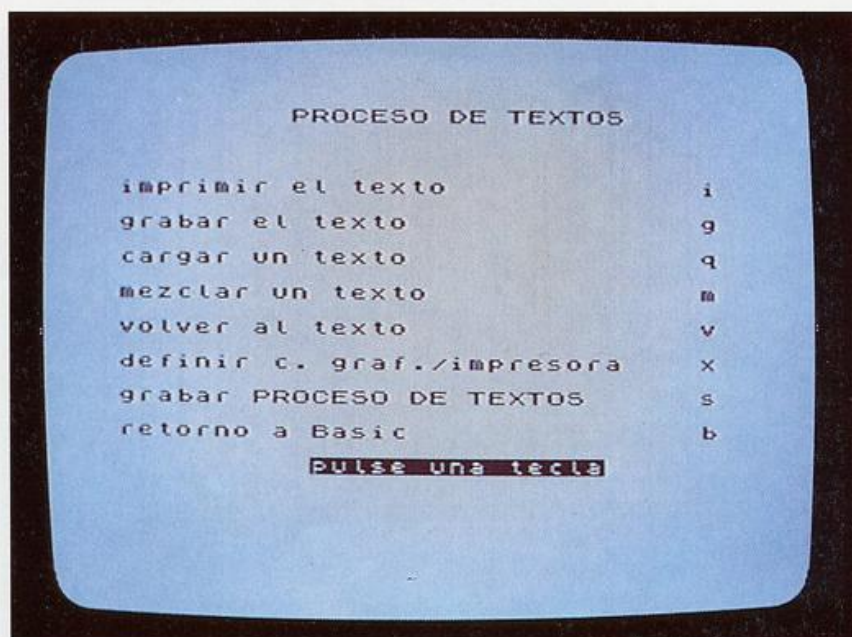


*Un procesador de textos es cualquier paquete software que tenga por objetivo la entrada de textos para su posterior elaboración.*

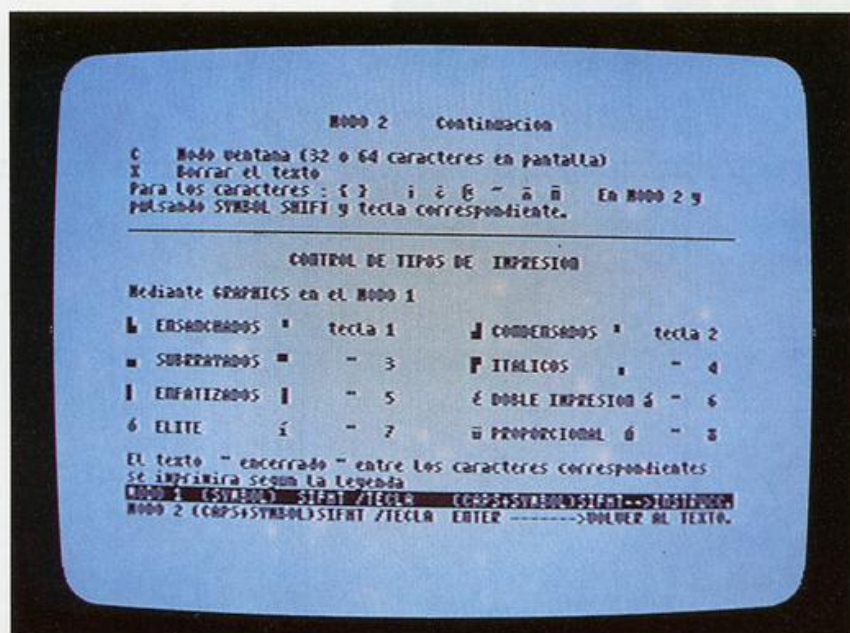
## ¿QUE ES UN PROCESADOR DE TEXTOS?

Seguramente que todos nosotros tenemos una idea, aunque sea aproximada, de las características, funciones y objetivos de un procesador de textos. Para asentar estos conocimientos básicos realizaremos una descripción de los aspectos más relevantes de una aplicación de esta índole. En general, un procesador de textos es cualquier paquete de *software* que tenga por objetivo la entrada de textos, ya sea directamente desde el teclado, o partiendo de un soporte magnético o de otro tipo, para su posterior elaboración siguiendo unas normas muy variadas, y salida del mencionado texto una vez procesado, usualmente en forma de escrito, aunque también a través de otros medios como puede ser el propio soporte magnético.

*Desde el modo de edición podemos acceder a dos pantallas resumen de instrucciones.*







*El TASWORD TWO posibilita la edición de textos continuos (sin word wrapping ni justificación), en bandera (con word wrapping pero sin justificación) y justificados (ambas características).*

Sin duda alguna, el punto más árduo de todo el proceso seguido es la elaboración de los datos introducidos para su posterior salida. Es en esta faceta en la que se diferencian por sus prestaciones los diversos programas de este tipo. Veamos entonces cuáles son las características más ha-

*Algunas versiones para el mercado español del TASWORD TWO incorporan no sólo caracteres propios de nuestro idioma, sino también la posibilidad de manejo del MICRODRIVE e impresoras semi-profesionales.*

bituales de dichos programas en esta fase del proceso.

Esencialmente la pantalla de nuestro ordenador se convierte en una ventana de una inmensa e imaginaria hoja de papel sobre la cual podemos escribir, borrar, y modificar los textos escritos a nuestro antojo. Para ello se establecen determinadas secuencias de teclas que, en forma de comandos, ejecutan las distintas funciones de las cuales esté dotado el procesador en cuestión.

Así por ejemplo, podremos desplazar la ventana a lo largo del texto, delimitar el número de caracteres que se nos muestran a un mismo tiempo (desplazar la ventana no sólo a lo largo sino a lo ancho de la hoja), efectuar la justificación mediante la inclusión automática de espacios en las líneas de texto introducidas, fijación de tabuladores y márgenes de escritura, y algunas otras funciones un tanto más especializadas.

En este sentido, disponemos de la posibilidad de borrar zonas extensas de texto, insertar líneas en blanco entre zonas ya redactadas copiar bloques de líneas, desplazar párrafos de texto desde un punto a otro del mismo, buscar palabras, reemplazar de forma automática unas por otras, y un sin fin de aplicaciones entre las que se cuentan las propias de las características de salida del texto.

En este último sentido, podemos marcar el tipo de letra a emplear para la salida impresa del documento, el número de copias que de éste se requieren, el espaciado entre líneas a utilizar... O bien, en lo referente al almacenamiento sobre soporte magnético la grabación de copias de seguridad, el encadenamiento o fusión de diferentes textos, la reedición de los mismos o la destrucción de las copias desechadas, así como la combinación del proceso de textos con otros programas tales como bases de datos o ficheros particulares que nos permitan efectuar la edición de cartas personalizadas, circulares, etc...

Una vez terminada esta revisión sobre las características presentes habitualmente en las aplicaciones de proceso de textos, pasemos a estudiar concretamente las prestaciones que nos ofrece el destinado a nuestro ordenador.

## TASWORD TWO



En el año 1983 la firma TASMAN SOFTWARE lanza al mercado su programa TASWORD TWO con el apabullante subtítulo de *the word proces-*





sor (el procesador de textos). Este determinante calificativo no resulta en absoluto un abuso de confianza en su producto, y la reacción del mercado ante el mismo es buena prueba de ello.

De hecho, como se ha mencionado anteriormente, en el mercado nacional no existen más que versiones sobre este mismo programa que lo modifican en aspectos muy concretos y bastante superficiales, aunque no en su esencia; tales como adaptaciones de la Aplicación al empleo de MIDRODRIVE, etc... que más adelante repasaremos con mayor detenimiento.

En su forma original, TASWORD TWO es un programa compuesto por un soporte BASIC de reducido tamaño que lleva a cabo funciones auxiliares y un bloque de rutinas en código máquina conformando el núcleo principal del programa.

La aplicación está controlada desde el BASIC por un menú general a través del cual se tiene acceso a las siguientes opciones:

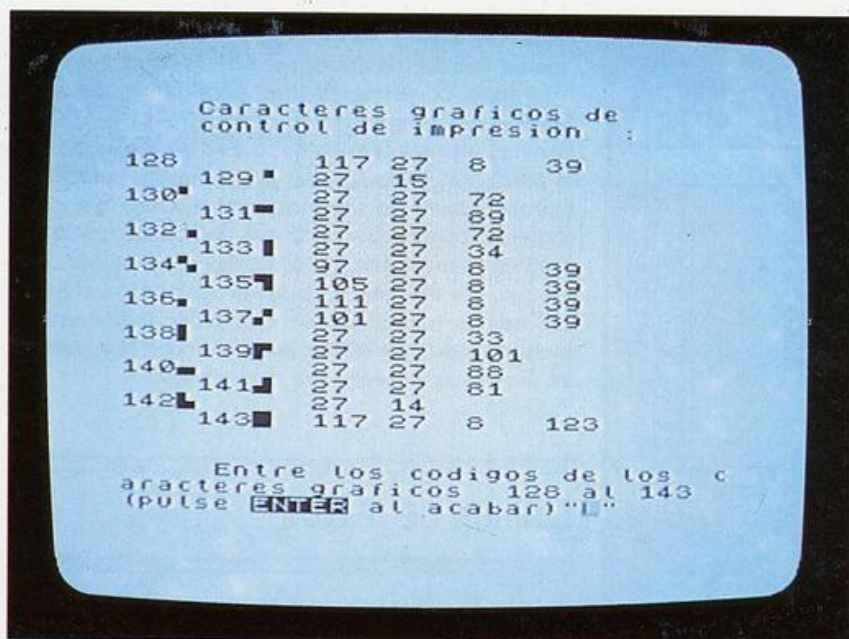
- IMPRESION.
- GRABACION.
- LECTURA.
- ENCADENAMIENTO.
- EDICION.
- DEFINICION DE CARACTERES Y CODIGOS DE IMPRESORA.
- COPIA DEL PROGRAMA.
- ACCESO AL BASIC.

El primer punto realiza la impresión del documento actualmente en memoria en cualquier impresora adaptada al programa. Inicialmente, éste puede trabajar con la ZX PRINTER, aunque mediante la opción de definición de códigos de impresora es posible controlar periféricos de tipo profesional a través de los correspondientes interfaces, como más adelante veremos.

Las tres siguientes opciones del menú son bien simples puesto que efectúan la grabación, carga o fusión en memoria de textos almacenados sobre soporte magnético (casete en la versión original). Como datos adicionales al realizar una grabación el programa indica la longitud de éste en líneas y el número de bytes de memoria empleados (este siempre coincide con el producto de 64×el número de líneas).

Asimismo, y como factor de seguridad, una vez completada la grabación se da opción a la verificación de dicho proceso. Por su parte, la lectura de un texto borra automáticamente cualquiera que existiera en memoria con anterioridad pasando a cargar el nuevo a partir de la primera línea, mientras que la fusión añade el leído inmediatamente a continuación del presente en memoria (al principio si no hubiera ninguno).

La definición de caracteres y códigos de impresora nos permite asignar a determinados caracteres gráficos accesibles por el teclado, coincidentes con los serigrafiados sobre las teclas numéricas, códigos para impresora que no constituyen caracteres representables sino la activación de

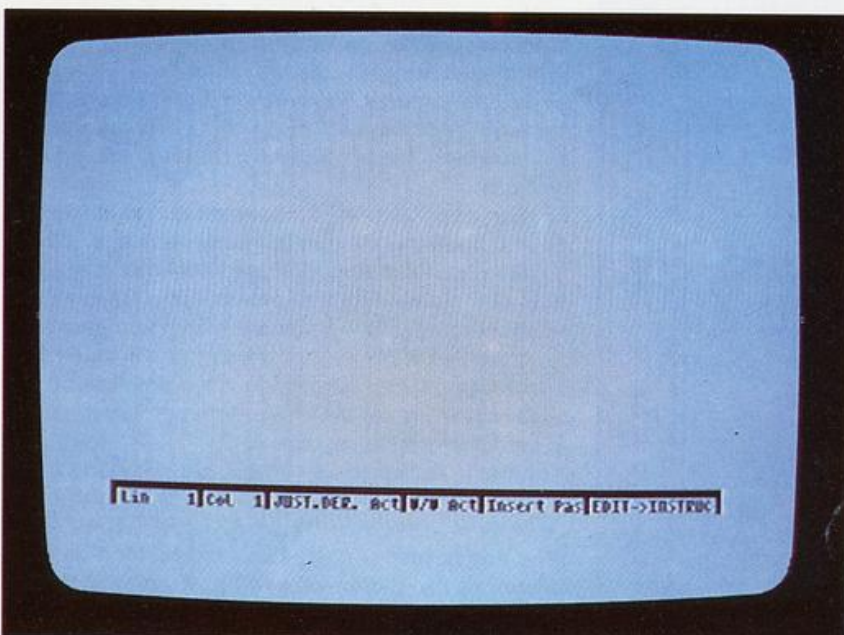


*Durante el proceso de edición, se muestra constantemente en la zona inferior de la pantalla información sobre la posición del cursor y las características de la edición.*

determinadas funciones de la misma como son el salto automático de página, impresión en enfático, itálica, subrayado, etc...

Lógicamente, estas características sólo son posibles en impresoras de una cierta calidad, pero no en la ZX PRINTER. Precisamente pensando en

*La ventana mostrada usualmente por el TASWORD es de 22 líneas de 64 caracteres sobre un máximo de 300 líneas disponibles.*





i!

La aplicación está controlada desde el BASIC por un menú general a través del cual se tiene acceso a las siguientes opciones: IMPRESION, GRABACION, LECTURA, ENCADENAMIENTO, EDICION, DEFINICION DE CARACTERES Y CODIGOS DE IMPRESORA, COPIA DEL PROGRAMA, ACCESO AL BASIC.

\*

El texto completo está compuesto por un máximo de 300 líneas de 64 caracteres de ancho, a lo largo de los cuales nos podremos desplazar mediante el uso de los cursores produciendo los correspondientes *scrolling* para visualizar permanentemente en pantalla el punto y entorno donde nos encontramos.

\*

La definición de caracteres y códigos de impresora nos permite asignar a determinados caracteres gráficos códigos para impresora que no constituyen caracteres representables sino la activación de determinadas funciones de la misma como son el salto automático de página, impresión en enfatizado, itálica, subrayado, etc...

este tipo de impresoras semi-profesionales esta misma opción del menú permite el control de los interfaces comerciales más habituales: Kempston e Indescomp.

Los dos últimos puntos del menú se explican por sí solos. La grabación del programa nos permite extraer copias de seguridad del mismo, completamente operativas pero que jamás deberemos emplear para vulnerar los derechos de autor del mismo. Por otra parte, la opción de retorno al BASIC nos permite el acceso a éste donde podemos incluir nuevas partes del programa con funciones de particular interés para nuestro uso.

## EDICION DE TEXTOS

Intencionadamente, hemos pospuesto el comentario de esta opción por ser la más importante del programa al constituir propiamente dicho su fase de proceso de textos. Los anteriores puntos del menú eran gestionados desde el BASIC, lo cual nos puede permitir, dotados de cierto nivel de conocimientos, modificar las características de las mismas. Por el contrario, las rutinas de edición se encuentran íntegramente programadas en código máquina, siendo por tanto de más difícil acceso.

Al entrar en el modo de edición se presenta ante nosotros una ventana sobre el texto completo conformada por 22 líneas de 64 caracteres, estando reservadas la 23 y 24 de la pantalla, para información del Sistema tal como: línea y columna en que se encuentra el cursor, estado activo o pasivo de los modos de justificación automática, inserción, fijación de mayúsculas o edición de gráficos.

El texto completo está compuesto por un máximo de 300 líneas de 64 caracteres de ancho, a lo largo de los cuales nos podremos desplazar mediante el uso de los cursores produciendo los correspondientes *scrolling* para visualizar permanentemente en pantalla el punto y entorno donde nos encontramos. Dicha posición viene siempre marcada por un cursor intermitente representado por el carácter gráfico de la tecla 3.

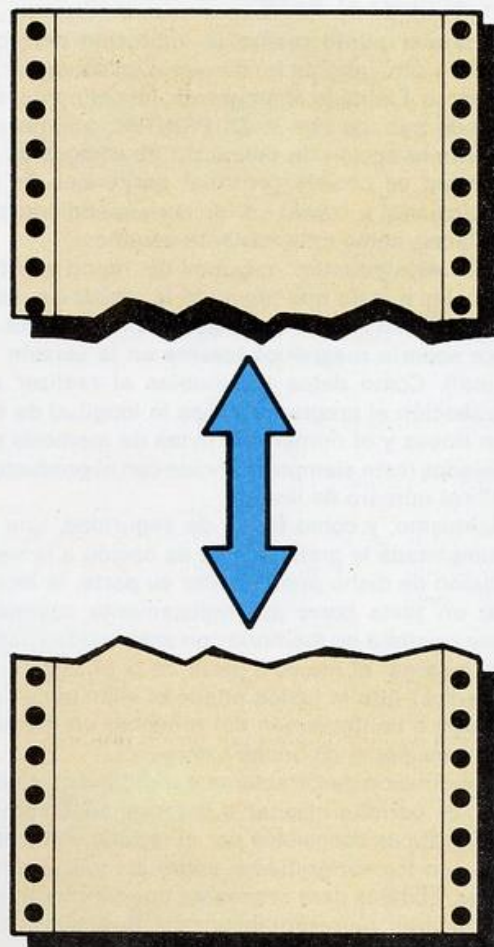
Aunque la Aplicación se acompaña de un manual de considerable extensión, mediante la pulsación de **EDIT** en combinación con **EXTENDED MODE** tenemos acceso a dos pantallas resumen de las funciones de edición, que como veremos a continuación no son pocas.

Como ya hemos mencionado, las flechas nos per-

miten el desplazamiento por el texto carácter a carácter, **TRUE VIDEO** el retroceso palabra por palabra, e **INVERSE VIDEO** el avance del mismo tipo. **DELETE** efectúa el borrado del carácter anterior, con desplazamiento del resto de la línea una posición a la izquierda, **CAPS LOCK** la conexión/desconexión del modo mayúsculas y **GRAPHICS** función similar para el modo gráfico, mientras que **ENTER** efectúa un retorno de carro (paso al comienzo de la línea siguiente) y la pulsación combinada de las teclas **CAPS SHIFT** y **SYMBOL SHIFT** al modo **EXTENDED**. Para desplazamientos a zonas más lejanas de la ventana de texto se habilitan los siguientes controles: **TO** (*scroll adelante una línea*) **THEN** (*scroll atrás una línea*), **EXTENDED F** (*scroll adelante una página, 22 líneas*), **EXTENDED G** (*scroll atrás una página*), **AT** desplazamiento a la primera línea del texto y **OR** a la última.

Cada vez que se alcanza el final de una línea ésta es automáticamente justificada a la derecha con inserción de espacios adicionales entre palabras

*Entre las funciones que desarrolla el TASWORD TWO nos encontramos con la inserción de líneas en blanco entre zonas ya redactadas.*





de derecha a izquierda, hasta completar la longitud máxima de la misma, antes de posicionar el cursor al comienzo de la siguiente línea. Esta posibilidad puede ser deshabilitada por medio de **EXTENDED J**, quedando entonces los textos dispuestos en bandera por efecto del *word wrapping* (salto automático a la línea siguiente de la última palabra que no cabe completa en la anterior). Esta última función también puede ser desactivada (**EXTENDED W**), con lo cual los textos serán escritos en modo continuo sin tenerse en cuenta para nada la longitud de las líneas.

Para la inserción de caracteres o palabras entre líneas ya justificadas utilizamos **AND**, así como para insertar párrafos podemos hacer uso del modo de inserción automático pulsando **EXTENDED I**. La función inversa, es decir, la eliminación de líneas completas, se consigue mediante el empleo de **NOT**.

En cuanto al tratamiento por línea, éstas pueden justificarse (**EXTENDED J**) y desjustificarse (**EXTENDED H**) individualmente, así como centrarse (**<>**), desplazarse hacia la izquierda carácter a carácter (**<=**) o hacia la derecha (**=>**). También es posible justificar todo un párrafo (desde la línea en que se encuentra el cursor hasta el próximo punto y aparte) mediante **STEP**.

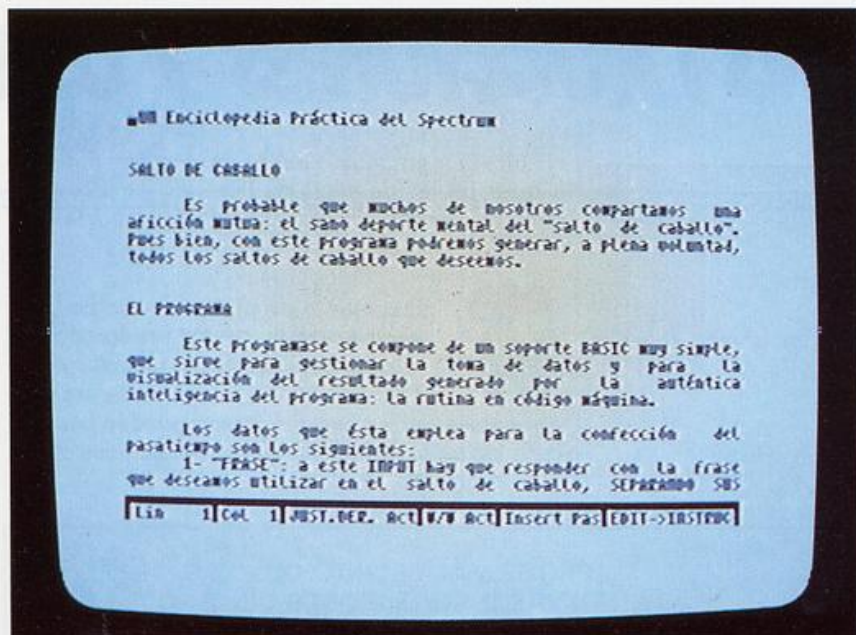
En cuanto al formato de presentación la fijación del margen izquierdo se consigue a través de **EXTENDED A**, la del derecho con **EXTENDED D** y la restauración de ambos con **EXTENDED S**. Asimismo, **EXTENDED C** visualizará una ventana de texto de 22 líneas por 32 caracteres de tamaño normal, con lo cual la escritura de un texto irá produciendo automáticamente el desplazamiento de la misma para proporcionar una visión de mayor entorno posible.

Por último, antes de entrar en las características más especiales del procesador, cabe mencionar que el borrado del texto en memoria se consigue por medio de **EXTENDED X** (con un mensaje permiso de validación) y la salida al menú de opciones mediante **STOP**.

Las versiones nacionales implementan, generalmente a través de los caracteres gráficos, signos propios del alfabeto español como son la apertura de interrogación y admiración, letra eñe mayúscula y minúscula, vocales acentuadas y u con diéresis.

Seguramente las características más espectaculares del procesador dentro del modo edición son la posibilidad de desplazar y copiar bloques de texto y la de la búsqueda y búsqueda con reemplazo de una palabra por otra o una frase.

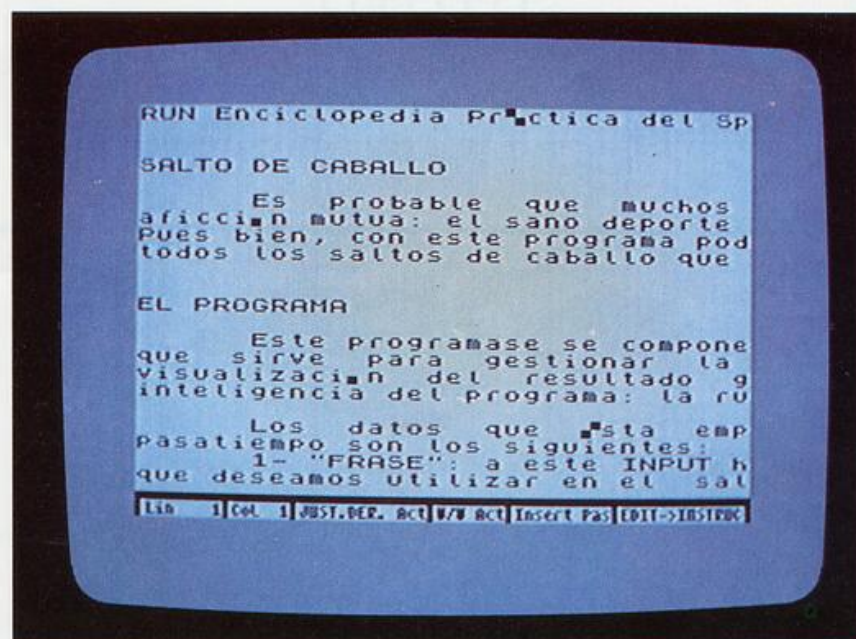
Para el tratamiento de bloques se ha de fijar previamente el límite superior (**EXTENDED B**) e inferior del mismo (**EXTENDED V**) y situar el cursor en el punto al cual se desee transferir el texto, pulsando **EXTENDED M** para el desplazamiento del bloque y **EXTENDED N** para la reproducción del mismo.



*La pulsación de **EXTENDED C** nos permite la entrada o salida del modo «ventana de 32 caracteres».*

Para finalizar, la opción de búsqueda o reemplazo de palabras se consigue a través de **EXTENDED R**, con enlace al BASIC para efectuar la entrada de las palabras a buscar o sustituir y, en este último caso la del texto o palabra que va a servir para la sustitución.

*La posibilidad de asignación de códigos de control a los caracteres gráficos, da entrada a características como la generación de distintos tipos de letra en impresoras semi-profesionales, etc...*





# INSTRUCCIONES Y OPERACIONES



L conjunto de instrucciones del Z80 es una ampliación del presente en el microprocesador 8080 de la firma INTEL. Por ello, la mayoría de los programas escritos para este último pueden ser ejecutados en el desarrollado por los técnicos de ZILOG en el año 1976.

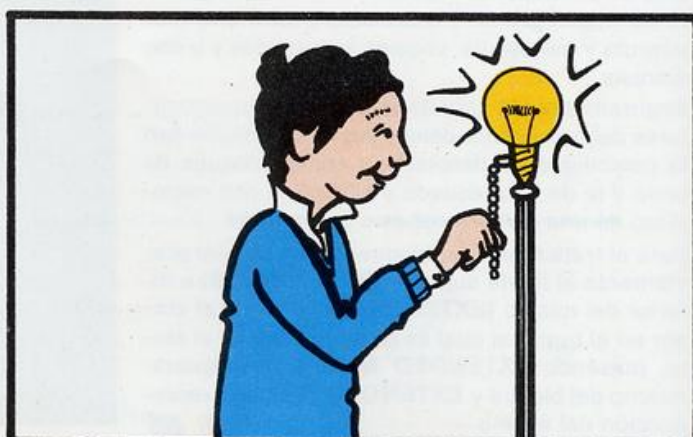
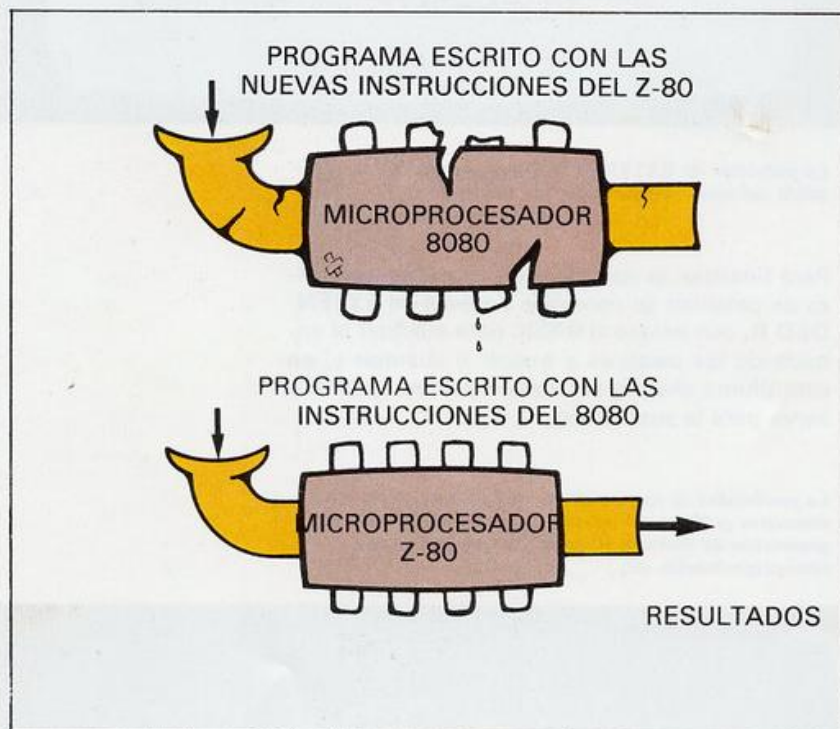
Es más, la compatibilidad podría hacerse efectiva a la inversa, siempre y cuando no incluyéramos en nuestros programas las nuevas instrucciones implementadas. No es este, en principio, nuestro objetivo, pues son precisamente los nuevos grupos los que confieren una potencia fuera de común a este singular microprocesador de 8 bits. En el capítulo anterior tratamos muchos nuevos conceptos y ya es hora de formalizar claramente su significado. Continuemos, por tanto, con algunas definiciones útiles.

A nadie se le oculta que cuando hablamos de programación, cualquier instrucción está formada por un grupo de caracteres que definen una operación. Un carácter es cualquier símbolo de los que nosotros consideramos elementales, como una letra del alfabeto, un número del 0 al 9, o los especiales: dólar (\$), asterisco (\*), entre otros.

Una operación es la acción específica que el ordenador ejecuta cuando una instrucción se lo exige. Por tanto, cualquier instrucción en C/M estará compuesta, entre otras cosas que en seguida

*Los programas escritos para el 8080 pueden ejecutarse en el Z80, pero el proceso inverso no siempre es posible.*

*Instrucción: definición de la operación. Operación: acción específica a ejecutar.*







comentaremos, por uno o varios códigos de operación, los cuales especifican al Z80 lo que debe hacer en todo momento.

## LOS FORMATOS

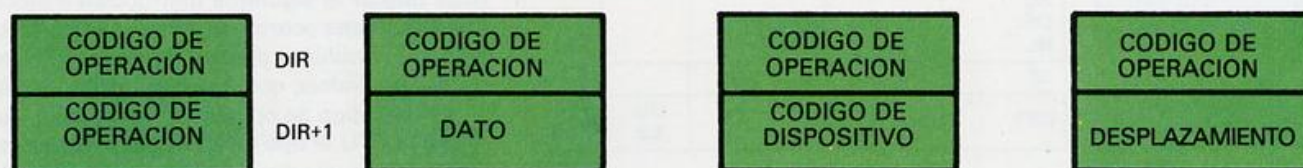
*Formato de las instrucciones del Z80.*

Las tablas que nos acompañan en estas páginas contienen el conjunto completo de instrucciones

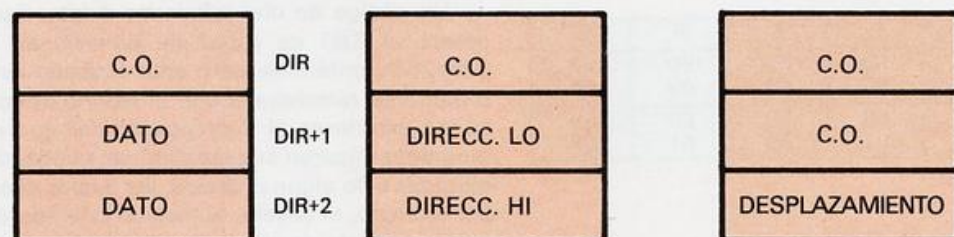
### 1. INSTRUCCIONES DE UN BYTE:



### 2. INSTRUCCIONES DE DOS BYTES:



### 3. INSTRUCCIONES DE TRES BYTES:



### 4. INSTRUCCIONES DE CUATRO BYTES:







que es posible implementar manejando el Z80, ordenadas según su función. Son 158 tipos diferentes (696 de diferente código de operación) y no debemos preocuparnos si encontramos algunos símbolos o datos que todavía no hayamos estudiado. Todo llegará en su momento. En la figura se muestran los formatos de las ins-

ENTRADA/SALIDA		TRANSFERENCIA		BUSQUEDA	
		ORIGEN		DIRECCION DE BUSQUEDA	
INA,(n)	DB n	DESTINO (DE)	(HL)	INSTRUCCION	(HL)
OUT(n),A	D3 n		ED A0 LDI		ED A1 CPI
INI	ED A2		ED B0 LDIR		ED B1 CPIR
INIR	ED B2		ED A8 LDD		ED A9 CPD
IND	ED AA		ED B8 LDDR		ED B9 CPDR
INDR	ED BA	INTERCAMBIO (EX y EXX)			
OUTI	ED A3		AF' BC' DE' HL' IX		
OTIR	ED B3		AF 08		
OUTD	ED AB		BC DE HL D9		
OTDR	ED BB		DE EB		
			(SP) E3 DD E3 FD E3		

## ENTRADA/SALIDA

	A	B	C	D	E	H	L
IN	ED 78	ED 40	ED 48	ED 50	ED 58	ED 60	ED 68
OUT	ED 79	ED 41	ED 49	ED 51	ED 59	ED 61	ED 69

## ARITMETICO DE 16 BITS

		BC	DE	HL	SP	IX	IY
ADD	HL	09	19	29	39		
	IX	DD 09	DD 19		DD 39	DD 29	
	IY	FD 09	FD 19		FD 39		FD 29
ADC	HL	ED 4A	ED 5A	ED 6A	ED 7A		
SBC	HL	ED 42	ED 52	ED 62	ED 72		
INC		03	13	23	33	DD 23	FD 23
DEC		0B	1B	2B	3B	DD 2B	FD 2B

trucciones, atendiendo al número de octetos necesarios para su correcta construcción. Como observamos pueden estar compuestas por uno, dos, tres o cuatro bytes, extremo que podemos confirmar ojeando las tablas principales. Pero, ¿cómo es capaz de reconocer el microprocesador su longitud?

Si nos fijamos con detalle, el primer byte de todas las instrucciones es siempre un código de operación, sea cual sea el tamaño de ésta. El proceso seguido por la CPU de nuestro Spectrum es el siguiente:

— El registro PC le indica la dirección de memoria donde encontrar la siguiente instrucción. Allí lee un código de operación, y en función de éste, reconoce el número de bytes de la instrucción, cantidad que es añadida al contador de programa PC. Finalmente, la instrucción es procesada y el Z80 busca un nuevo código de operación en la dirección señalada por el registro PC.

Algunas instrucciones, las cuales provocan saltos o bifurcaciones en el proceso secuencial de un programa (algo parecido a lo que ocurre con las instrucciones **GO TO** o **GOSUB** del BASIC), pueden modificar bruscamente el contenido del registro PC, pero siempre quedará almacenado en éste la dirección de memoria en donde la CPU debe buscar la siguiente instrucción a ejecutar. Los siguientes octetos al primer código de operación si los hubiere pueden contener información de diversa índole, que a continuación detallamos:

— Otro código de operación de 8 bits, el cual indica a la CPU el tipo de instrucción de que se trata (transferencia de datos, entrada/salida, lógica, manejo del STACK, etc.).

— Un dato de 8 ó 16 bits, codificado en binario, ASCII, binario codificado en decimal (BCD), etc.

— Un código de dispositivo de 8 bits. Teóricamente el Z80 es capaz de direccionar hasta 218=256 periféricos, pero esta cuestión está íntimamente relacionada con el diseño de los circuitos exteriores al microprocesador que en su momento trazaron sus fabricantes, quedando habilitadas sólo algunas líneas, las cuales soportan por ejemplo, el casete, el teclado o la impresora.

— Un byte de desplazamiento, es decir, un número en complemento a dos con signo, empleado en las instrucciones que manejan direccionamiento indexado (pronto aprenderemos a calcular el complemento a dos de un número).

— Una dirección determinada por dos bytes, donde el microprocesador siempre interpretará como byte de menor peso, LO (abreviatura de *Low*, bajo), el presente en primer lugar, siendo el segundo el más significativo, HI (abreviatura de *High*, alto).

Para todos ha de estar claro que el número en cuestión es interpretado por el Z80 de la siguiente manera:

— Una dirección determinada por dos bytes, donde el microprocesador siempre interpretará como byte de menor peso, LO (abreviatura de *Low*, bajo), el presente en primer lugar, siendo el segundo el más significativo, HI (abreviatura de *High*, alto).

Para todos ha de estar claro que el número en cuestión es interpretado por el Z80 de la siguiente manera:

VALOR TOTAL = VALOR DEL BYTE LO + 256 × VALOR DEL BYTE HI

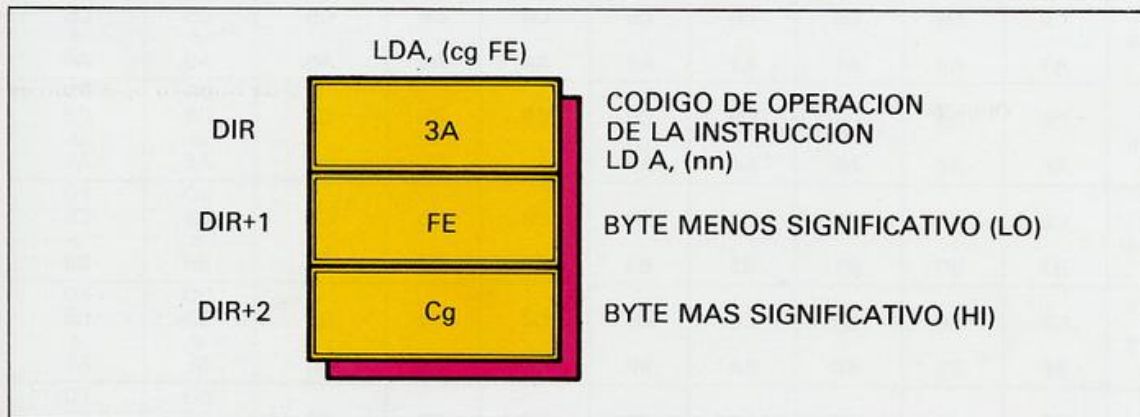




Por ejemplo, consideremos la instrucción **LD A,(C9FE)**, la cual «carga» el acumulador con el valor almacenado en la dirección C9FE H. En la figura se muestra el formato en que el ordenador debe encontrarla para su correcta interpretación. En el próximo capítulo encontraremos las indicaciones necesarias para obtener el máximo rendi-

miento de las tablas y realizaremos un recorrido a través de los diferentes grupos que las componen.

*Cuando la CPU interpreta una dirección supone que ha encontrado los bytes que la definen en el orden.*



## MANIPULACION DE BITS

BIT		A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
		CB	CB	CB	CB	CB	CB	CB	CB	DD CB d 46	FD CB d 46
0		47	40	41	42	43	44	45	46		
1		4F	48	49	4A	4B	4C	4D	4E	DD CB d 4E	FD CB d 4E
2		57	50	51	52	53	54	55	56	DD CB d 56	FD CB d 56
3		5F	58	59	5A	5B	5C	5D	5E	DD CB d 5E	FD CB d 5E
4		67	60	61	62	63	64	65	66	DD CB d 66	FD CB d 66
5		6F	68	69	6A	6B	6C	6D	6E	DD CB d 6E	FD CB d 6E
6		77	70	71	72	73	74	75	76	DD CB d 76	FD CB d 76
7		7F	78	79	7A	7B	7C	7D	7E	DD CB d 7E	FD CB d 7E
0		87	80	81	82	83	84	85	86	DD CB d 86	FD CB d 86
1		8F	88	89	8A	8B	8C	8D	8E	DD CB d 8E	FD CB d 8E

# i!

El conjunto de instrucciones del Z80 es una ampliación del presente en el microprocesador 8080 de la firma INTEL.

# \*

El primer byte de todas las instrucciones es siempre un código de operación.





RES	2	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
	3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E
	4	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
	5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE	FD CB d AE
	6	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
	7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE
SET	0	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6
	1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
	2	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6	FD CB d D6
	3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
	4	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
	5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
	6	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
	7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

## RETORNO

CONDICION										
		INCOND.	ARRASTRE	NO ARRASTRE	CERO	NO CERO	PARIDAD PAR	PARIDAD IMPAR	SIGNO NEG.	SIGNO POS.
RET	(SP) (SP+1)	C9	D8	D0	C8	C0	E8	E0	F8	F0
RETI	(SP) (SP+1)	ED 4D								
RETN	(SP) (SP+1)	ED 45								





## SALTOS Y LLAMADAS

CONDICION											
		INCOND.	ARRASTRE	NO ARRASTRE	CERO	NO CERO	PARIDAD PAR	PARIDAD IMPAR	SIGNO NEG.	SIGNO POS.	B≠0
JP	nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	IA n n	I2 n n	
JR	PC+e	18 e-2	38 e-2	30 e-2	28 e-2	20 e-2					
JP	(HL)	E9									
JP	(IX)	DD E9									
JP	(IY)	FD E9									
CALL	nn	CD n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n	
DJNZ	PC+e										10 e-2
DIRECCION DE LLAMADA											
				00 h	08 h	10 h	18 h	20 h	28 h	30 h	38 h
INSTRUCCION				RST 0	RST 8	RST 10	RST 18	RST 20	RST 28	RST 30	RST 38
CODIGO DE OPERACION				C7	CF	D7	DF	E7	EF	F7	FF

## CARGA DE 16 BITS (LD)

ORIGEN											POP
	AF	BC	DE	HL	SP	IX	IY	nn	(nn)	(SP)	
DESTINO	AF									F1	
	BC							01 n n	ED 4B n n	C1	
	DE							11 n n	ED 5B n n	D1	
	HL							21 n n	2A n n	E1	
	SP			F9		DD F9	FD F9	31 n n	ED 7B n n		
	IX							DD 21 n n	DD 2A n n	DD E1	
	IY							FD 21 n n	FD 2A n n	FD E1	
	(n)	ED 43 n n	ED 53 n n	22 n n	ED 73 n n	DD 22 n n	FD 22 n n				
PUSH	(SP)	F5	C5	D5	E5	DD E5	FD E5				

# i!

Los nuevos grupos de instrucciones implementadas en el Z80 son los que dotan de una potencia fuera de lo común a este microprocesador.

# \*

El Z80 es un microprocesador de 8 bits.





## LOGICO 8 BITS

ORIGEN											
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
AND	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
XOR	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
OF	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
CP	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n

## CARGA DE 8 BITS (LD)

	ORIGEN															
	I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX+d)	(IY+0)	(nn)	n
A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n n	3E n
B			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d		06 n
C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
D			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
H			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
(HL)			77	70	71	72	73	74	75							36 n
(BC)			02													
(DE)			12													
(IX+d)			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d n
(IY+d)			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d n
(nn)			32 n n													
I			ED 47													
R			ED 4F													





## ROTACION Y DESPLAZAMIENTO

ORIGEN Y DESTINO										
	A	B	C	D	E	H	L	(HL)	(IX+d)	(YI+d)
RLC	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06
RRC	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E
RL	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16
RR	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E
SLA	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26
SRA	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E
SRL	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB d	FD CB d
RLD								ED 6F		
RRD								ED 67		
RLCA	07									
RRCA	0F									
RLA	17									
RRA	1F									

## ARITMETICO 8 BITS

ORIGEN											
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
ADD	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADC	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUB	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SBC	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
INC	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DEC	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

## ARITMETICO Y DE CONTROL DE LA C.P.U.

DAA	27
CPL	2F
NEG	ED 44
CCF	3F
SCF	37
NOP	00
HALT	76
DI	F3
EI	FB
IM0	ED 46
IM1	ED 56
IM2	ED 5E







# IMPRESA

**i!**

Si somos observadores nos daremos cuenta de que las letras minúsculas empleadas en los juegos ITALICA y ROTULACION son iguales, por lo que podremos aprovechar con habilidad el esfuerzo de introducción si deseamos disponer de ambos.



En el caso de los juegos de caracteres, al estar íntegramente constituidos por datos, la reubicabilidad es completa, es decir, no importa el lugar de memoria en que los grabemos, podrán ser situados en cualquier punto de la memoria.



Como sabemos, el juego de caracteres que utiliza el Spectrum (mayúsculas, minúsculas y caracteres especiales) están almacenados en ROM en grupos de 8 bytes, en donde cada uno de estos octetos equivale a una línea del *pattern* (forma del carácter).

Entre las muchas Variables del Sistema que nuestro Spectrum posee, existe una que le indica al microprocesador la dirección de comienzo del generador de caracteres. **CHARS** es nombre simbólico que Sir Clive Sinclair le ha dado a esta variable, y que no tiene nada que ver con las variables que nosotros definimos en BASIC.

Inicialmente, el puntero de la variable **CHARS** señala a una dirección de la ROM, pero en contra de lo que pudiéramos pensar, dicha dirección no indica el comienzo del generador de caracteres, sino 256 bytes antes. Esto es debido a que los primeros 32 caracteres no son representables (8 bytes por 32 caracteres=256 bytes).

Al encontrarse en zona RAM el área de Variables del Sistema, podemos «engañar» a nuestro Spectrum modificando la información que contiene la variable **CHARS**, e introduciendo en la dirección adecuada nuestro propio juego de caracteres.

Sin duda, el problema más grave es disponer de un juego de caracteres definido por el cual poder sustituir el propio de Sinclair. Recrearse la forma de tantos caracteres es sin duda un árduo trabajo, aunque para animarnos incluimos en este artículo nada más y nada menos que cuatro juegos diferentes, de los cuales tenemos una muestra por impresora, para que podamos realizar nuestra elección con conocimiento de causa.

En cuanto a la introducción de los datos correspondientes a cada juego, podemos utilizar cualquier cargador de código máquina, como por ejemplo el que reproducimos en estas páginas y cuyo funcionamiento es de extremada sencillez. Una vez introducido (podemos convertirlo para alguna otra ocasión en que necesitemos una rutina similar) y ejecutado, aparecerá en la pantalla la petición de DIRECCION DE INICIO. Esta supone la dirección de memoria a partir de la cual se comenzarán a introducir los datos.

En nuestro caso, como se puede apreciar en los listados de los juegos de caracteres, hemos seleccionado la dirección 30720, por ser múltiplo de 256 y válida para los modelos de 16K y 48K, aunque al tratarse de datos esta dirección carece de importancia decisiva, siempre y cuando no la escojamos tan baja que altere el propio programa BASIC, o alguna otra zona de memoria de importancia.

Acto seguido se efectúa la toma de datos byte a byte, que deberán ser introducidos, tal como aparecen en el listado, de izquierda a derecha y de arriba a abajo, siendo el número de la columna izquierda un mero indicativo de la dirección a partir de la cual se efectúa la carga, para servirnos de guía entre el maremagnum de cifras (algo así como las piedras de Pulgarcito).

Como característica adicional de la rutina, si entramos el código E, se tratará como una detección de error, y el programa retrocederá un paso en la introducción de los datos. Asimismo, el código F se ha habilitado para la notificación de fin de trabajo, lo cual dará acceso a una pregunta sobre el nombre con el cual deseamos realizar la grabación.

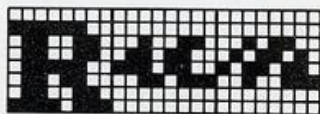
De esta sencilla manera podremos hacernos con los juegos de caracteres que acompañan este artículo.

A continuación exponemos tres pequeños programas con los cuales, una vez definido el nuevo juego de caracteres, podremos:

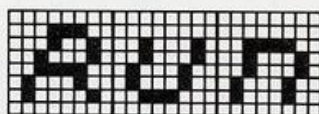
1. Calcular el nuevo contenido de la variable **CHARS** en función de la dirección a partir de la cual queremos almacenar el nuevo juego de caracteres. (CALCULADOR.)

2. Cargar los valores que definen los *patterns*. (GENERADOR.)

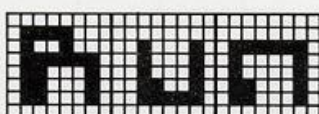
3. Reubicar el juego de caracteres. (REUBICADOR.)



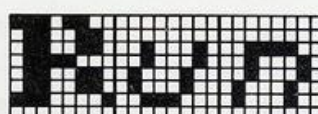
MARGARITA



ITALICA



DATA



ROTULACION



## MARGARITA

```

! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~ @

```

## DATA

```

! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~ @

```

## ITALICA

```

! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~ @

```

## ROTULACION

```

! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~ @

```

## SINCLAIR

```

! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~ @

```

## PROGRAMA CALCULADOR

```

1 REM ***** RUN *****
10 PRINT AT 20,0;"INTRO. DIR. DE ALMACE
NAMIENTO"
20 INPUT "DIRECCION=" ; DIR
30 LET CHARS=DIR-256
40 LET B=INT (CHARS/256)
50 LET A=CHARS-256*B
60 PRINT "DIR. ALMACEN.~";DIR
70 PRINT "DIREC. CHAR.~";CHAR
80 PRINT,"POKE ";23606;" ";A
90 PRINT,"POKE ";23607;" ";B
100 PRINT
110 POKE 23692,255
120 GOTO 10

```

## PROGRAMA REUBICADOR

```

1 REM ***** RUN *****
10 INPUT "DIREC. ORIGEN ";D1
20 INPUT "DIREC. NUEVA ";D2
30 FOR N=0 TO 1023
40 POKE (D2+N), PEEK (D1+N)
50 NEXT N
60 STOP

```

## PROGRAMA GENERADOR

```

1 REM ***** RUN *****
10 INPUT "DIREC. COMIENZO ";D
20 FOR N=0 TO 1023
30 PRINT AT 10,3;"DIRECCION ";N:D
40 INPUT "CONTENIDO ";C
50 POKE (D+N),C
60 NEXT N
70 STOP

```

## PROGRAMA CARGADOR

```

10 REM FERNANDO LOPEZ MARTINEZ $ CARGADOR DE CODIGO
MAQUINA $ CARACTERES
20 LET B=CHARS B+CHARS B+CHARS B+CHARS B+CHARS B+
CHARS B+CHARS B: DEF FN P$(I)=( " " +AS)(LEN AS TO )
30 POKE 23658,0: BORDER 1: PAPER 1: INK 7: CLS
40 INPUT "DIRECCION DE INICIO ?";A: LET P=A
50 PRINT A:TAB 7;
60 FOR I=1 TO 6
70 INPUT "BYTE ?"; LINE AS
80 IF AS="F" THEN GO TO 160
90 IF AS="E" THEN GO TO 230
100 IF VAL AS<8 OR VAL AS>255 THEN GO TO 40
110 POKE A,VAL AS
120 LET A=A+1
130 PRINT FN P$(I); " ";
140 NEXT I: PRINT
150 GO TO 50
160 INPUT "NOMBRE DEL FICHERO ?";AS
170 IF AS="" THEN GO TO 30
180 IF LEN AS>10 THEN GO TO 160
190 SAVE AS:CODE P,A-P+1
200 PRINT #0: " PREPARE LA VERIFICACION"
210 VERIFY AS:CODE P,A-P+1
220 RUN
230 IF I=1 THEN PRINT AT 23-PEEK 23689,31: LET I=7
240 PRINT B$;
250 LET A=A-1: LET I=I-1
260 GO TO 70

```

## MARGARITA

```

30720 0 0 0 0 0 0
30726 0 0 24 24 24 24
30732 0 0 0 0 0 36
30738 36 0 0 0 0 0
30744 0 36 126 36 36 126
30750 36 0 0 126 208 126
30756 17 254 16 0 0 98
30762 100 0 16 38 70 0
30768 16 48 68 48 90 132
30774 122 0 0 16 0 0
30780 0 0 0 4 0 0
30786 0 0 0 4 0 0
30792 32 16 16 16 16 16
30798 32 0 0 0 20 0
30804 62 0 20 0 0 0
30810 0 0 62 0 0 0
30816 0 0 0 0 0 0
30822 0 16 0 0 0 0

```

```

30828 62 0 0 0 0 0
30834 0 0 24 24 0 0
30840 0 2 4 0 16 32
30846 64 0 0 60 70 203
30852 211 98 60 0 0 24
30858 40 0 0 0 62 0
30864 56 68 6 28 112 193
30870 254 0 124 131 3 12
30876 131 131 124 0 0 24
30882 56 72 255 0 0 0
30888 126 192 252 2 3 130
30894 124 0 62 64 252 194
30900 131 66 60 0 34 127
30906 129 18 12 18 16 0
30912 68 66 195 126 129 129
30918 126 0 60 66 193 67
30924 63 2 124 0 0 0
30930 16 0 0 16 0 0
30936 0 0 16 0 0 16
30942 16 32 0 0 4 0
30948 16 0 4 0 0 0
30954 0 62 0 62 0 0
30960 0 0 16 4 0 0
30966 16 0 60 102 67 6
30972 12 0 0 0 0 60
30978 74 86 94 64 68 0
30984 24 24 44 44 126 70
30990 239 0 252 102 102 124
30996 102 102 252 0 60 98
31002 96 96 96 98 60 0
31008 252 102 102 102 102 102
31014 252 0 254 98 104 120
31020 186 98 254 0 254 98
31026 184 120 184 96 248 0
31032 68 102 96 110 102 102
31038 68 0 255 102 102 126
31044 102 102 255 0 60 24
31050 24 24 24 24 68 0
31056 126 12 12 12 148 148
31062 120 0 242 100 104 128
31068 108 102 242 0 240 96
31074 96 96 98 98 254 0
31080 199 110 118 86 86 70
31086 239 0 199 98 114 98
31092 78 70 226 0 60 102
31098 102 102 102 102 60 0
31104 252 102 102 102 96 96
31110 240 0 124 198 198 198
31116 198 214 124 0 252 102
31122 102 124 108 102 247 0
31128 68 98 112 60 14 102
31134 68 0 255 153 153 24
31140 24 24 60 24 24 24
31146 70 70 70 70 60 0
31152 111 70 70 44 44 44
31158 16 0 115 49 49 53
31164 53 63 18 0 119 38
31170 52 24 20 38 119 0
31176 115 98 52 24 24 24
31182 60 0 127 3 6 12
31188 24 48 127 0 24 16
31194 16 16 16 16 24 0
31200 128 64 32 16 8 4
31206 2 0 48 16 16 16
31212 16 16 48 0 16 16
31218 56 84 16 16 16 0
31224 0 0 0 0 0 0
31230 0 255 0 28 50 120
31236 48 50 124 0 0 0
31242 29 34 100 191 0 0
31248 14 10 36 50 98 189
31254 0 0 0 36 28 0
31260 96 191 0 0 1 2
31266 60 72 200 119 0 0
31272 0 0 28 36 120 159
31278 0 0 15 25 18 60
31284 248 87 144 224 0 0
31290 31 58 212 63 72 112
31296 7 9 18 60 228 71
31302 0 0 4 0 8 16
31308 48 223 0 0 2 0
31314 12 56 216 127 144 240
31320 14 18 28 52 124 167
31326 0 0 6 9 18 28
31332 56 223 0 0 0 0
31338 84 170 170 87 0 0
31344 0 0 44 90 182 39
31350 0 0 0 0 28 38
31356 101 184 0 0 0 0
31362 20 58 246 183 64 192
31368 0 0 30 38 253 0
31374 24 48 0 32 56 72
31380 152 31 0 0 0 4
31386 12 22 58 221 0 0
31392 6 12 31 56 88 143
31398 0 0 0 0 38 108
31404 236 55 0 0 0 0
31410 34 87 188 56 0 0
31416 0 0 107 218 98 108
31422 0 0 0 0 18 45
31428 76 179 0 0 0 0
31434 9 50 220 63 72 112
31440 0 60 40 92 159 36
31446 48 48 14 0 0 0
31452 0 0 14 0 0 0
31458 0 0 8 0 0 0
31464 112 16 16 20 16 16
31470 112 0 0 20 48 0
31476 0 0 0 0 60 66
31482 153 161 161 153 66 60

```



# PROGRAMA

## ITALICA

30720	0	0	0	0	0	0
30726	0	0	24	24	24	24
30732	0	0	0	0	0	36
30738	36	0	0	0	0	0
30744	0	36	126	36	36	126
30750	36	0	0	126	200	126
30756	17	254	16	0	0	98
30762	100	0	16	38	70	0
30768	16	48	68	48	98	132
30774	122	0	0	16	0	0
30780	0	0	0	0	4	0
30786	0	0	0	0	4	0
30792	32	16	16	16	16	16
30798	32	0	0	0	20	0
30804	62	0	20	0	0	0
30810	0	0	62	0	0	0
30816	0	0	0	0	0	0
30822	0	16	0	0	0	0
30828	62	0	0	0	0	0
30834	0	0	0	24	24	0
30840	1	4	0	16	32	0
30846	64	0	0	60	70	74
30852	148	164	120	0	0	48
30858	0	16	32	32	248	0
30864	0	28	34	4	56	64
30870	124	0	0	30	4	24
30876	4	0	0	0	0	0
30882	10	36	126	0	0	0
30888	0	38	16	60	2	68
30894	56	0	0	38	32	68
30898	66	68	56	0	0	38
30906	2	4	8	16	32	0
30912	0	28	36	56	68	68
30918	56	0	0	28	34	34
30924	28	4	56	0	0	0
30930	16	0	0	16	0	0
30936	0	0	16	0	0	16
30942	16	32	0	4	8	0
30948	16	0	4	0	0	0
30954	0	62	0	0	0	0
30960	0	0	16	0	4	8
30966	16	0	68	102	67	6
30972	12	0	0	0	0	68
30978	74	86	94	64	68	0
30984	0	12	18	34	62	66
30990	66	0	0	28	18	68
30996	34	0	66	124	8	0
31002	34	32	64	68	56	0
31008	0	24	20	34	34	68
31014	120	0	0	30	16	68
31020	32	64	120	0	0	30
31026	16	68	32	64	64	0
31032	0	28	34	32	76	68
31038	56	0	0	18	18	68
31044	36	72	72	0	0	62
31050	0	16	16	32	248	0
31056	0	2	4	68	72	0
31062	56	0	18	28	76	0
31068	48	68	66	0	8	16
31074	16	32	32	64	124	0
31080	0	34	54	42	66	68
31086	68	0	18	26	42	0
31092	44	68	68	0	28	0
31098	34	68	34	68	34	68
31104	0	28	18	34	68	64
31110	64	0	0	68	66	66
31116	164	148	120	0	0	28
31122	18	34	68	66	66	0
31128	0	28	34	24	4	68
31134	56	0	62	0	0	17
31140	16	16	32	0	0	0
31146	34	34	68	68	56	0
31152	0	34	34	36	36	40
31158	16	0	0	33	33	66
31164	66	98	36	0	0	34
31170	20	56	24	68	138	0
31176	0	34	20	0	16	32
31182	64	0	0	62	4	0
31188	16	32	124	0	24	16
31194	16	16	16	16	24	0
31200	128	64	32	16	8	0
31206	2	48	0	16	16	16
31212	16	16	48	0	0	16
31218	56	84	16	16	16	0
31224	0	0	0	0	0	0
31230	0	255	0	28	58	128
31236	68	0	124	0	0	0
31242	12	2	68	0	56	0
31248	0	16	16	68	34	66
31254	124	0	0	0	28	32
31260	32	64	56	0	2	0
31266	2	28	36	68	56	0
31272	0	0	28	34	124	64
31278	56	0	0	5	0	12
31284	16	16	32	0	0	0
31290	14	18	34	68	4	120
31296	0	16	16	62	34	68
31302	68	0	0	4	0	24
31308	0	16	120	0	2	0
31314	0	4	0	72	48	72
31320	0	16	20	56	48	72
31326	66	0	0	16	16	16
31332	32	32	24	0	0	0
31338	54	73	73	146	146	0
31344	0	0	68	34	34	68
31350	68	0	0	0	0	34
31356	36	68	56	0	0	34
31362	28	18	34	68	64	64
31368	0	0	38	18	36	68
31374	0	38	0	0	14	16
31380	16	32	32	0	0	0
31386	38	32	24	4	120	0
31392	0	4	38	0	16	18
31398	12	0	0	0	34	34
31404	68	68	56	0	0	0
31410	34	34	36	48	16	0
31416	0	0	65	65	146	146
31422	108	0	0	0	34	28
31428	24	48	68	0	0	0
31434	18	36	68	0	16	96
31440	0	0	128	0	16	32
31446	120	0	14	0	8	56
31452	8	8	14	0	0	0
31458	0	0	0	0	0	0
31464	112	16	16	28	16	16
31470	112	0	0	28	48	0
31476	0	0	0	0	60	66
31482	153	161	161	153	66	68

## DATA

30720	0	0	0	0	0	0
30726	0	0	0	24	24	24
30732	24	24	0	24	0	36
30738	36	0	0	0	0	0
30744	0	36	126	52	52	126
30750	36	0	0	16	126	80
30756	126	22	126	16	0	102
30762	100	8	16	38	102	0
30768	0	56	40	16	42	100
30774	42	16	0	48	48	64
30780	0	0	0	0	24	0
30786	48	32	32	32	48	24
30792	0	24	12	4	4	4
30798	12	24	0	102	60	24
30804	24	60	102	0	0	24
30810	24	126	126	24	24	0
30816	0	0	0	56	56	0
30822	56	96	0	0	0	0
30828	126	126	0	0	0	0
30834	0	0	112	112	112	0
30840	0	24	24	0	126	0
30846	0	24	0	126	70	74
30852	82	98	126	0	0	0
30858	24	24	0	0	0	0
30864	0	126	70	6	126	64
30870	126	0	0	126	6	6
30876	126	2	126	0	124	0
30882	68	100	100	126	4	124
30888	0	126	66	64	126	6
30894	126	0	0	126	98	96
30898	126	66	126	0	126	0
30906	2	2	6	6	6	0
30912	0	126	98	98	126	66
30918	126	0	0	126	78	78
30924	126	2	2	0	24	24
30930	24	0	0	24	24	0
30936	0	24	24	0	0	24
30942	24	96	0	0	8	16
30948	48	16	0	0	0	0
30954	126	126	0	126	126	0
30960	0	0	0	4	6	0
30966	0	0	0	126	70	38
30972	24	24	0	24	0	127
30978	65	79	75	111	96	127
30984	0	68	36	62	98	98
30990	98	0	124	68	126	126
30996	98	98	126	0	0	126
31002	66	64	96	98	126	0
31008	0	126	66	66	98	98
31014	126	0	0	126	64	126
31020	96	96	126	0	126	0
31026	64	126	96	96	96	0
31032	66	126	66	64	182	98
31038	126	0	0	66	66	126
31044	98	98	98	0	16	0
31050	16	16	24	24	24	0
31056	0	4	4	6	70	0
31062	126	0	0	68	68	126
31068	98	98	98	0	64	64
31074	64	64	96	96	124	0
31080	0	126	74	74	186	186
31086	186	0	0	126	66	66
31092	98	98	98	0	126	0
31098	98	98	98	0	126	0
31104	0	126	66	66	126	96
31110	96	0	0	126	66	66
31116	66	78	126	0	0	126
31122	72	126	98	98	98	0
31128	0	126	64	126	6	6
31134	126	0	0	126	16	16
31140	24	24	24	0	0	66
31146	66	66	98	98	126	0
31152	0	98	98	98	38	36
31158	68	0	0	74	74	74
31164	186	186	126	0	0	66
31170	126	24	126	98	98	0
31176	0	66	66	126	24	24
31182	24	0	0	126	12	8
31188	16	48	126	0	0	38
31194	24	24	24	24	38	0
31200	0	96	48	24	12	6
31206	3	0	0	120	24	24
31212	24	24	128	0	0	24
31218	68	126	90	24	24	0
31224	0	0	0	62	50	32
31230	0	255	0	0	0	255
31236	112	32	126	0	0	0
31242	62	2	62	50	62	0
31248	0	64	64	124	68	180
31254	124	0	0	0	62	34
31260	32	58	62	0	0	2
31266	2	62	34	38	62	0
31272	0	0	62	50	62	32
31278	62	0	0	28	16	56
31284	16	24	24	0	0	0
31290	62	34	50	62	2	62
31296	0	32	32	62	34	50
31302	58	0	0	0	0	0
31308	8	12	12	0	0	0
31314	0	8	12	12	4	28
31320	0	72	88	80	124	106
31326	100	0	0	24	0	0
31332	12	12	12	0	0	0
31338	126	74	186	186	98	0
31344	0	0	62	34	34	58
31350	50	0	0	0	62	38
31356	34	34	62	0	0	0
31362	62	34	50	62	32	32
31368	0	0	62	34	58	62
31374	2	2	0	0	68	36
31380	32	48	48	0	0	0
31386	62	32	62	6	62	0
31392	0	16	56	16	16	24
31398	28	0	0	0	34	34
31404	34	58	62	0	0	0
31410	58	0	22	20	28	0
31416	0	66	82	86	86	86
31422	126	0	0	0	36	36
31428	24	52	52	0	0	0
31434	50	38	64	62	2	62
31440	8	0	62	12	8	48
31446	62	0	0	14	8	24
31452	8	0	14	0	0	0
31458	24	24	24	24	24	0
31464	0	112	16	12	16	16
31470	12	0	6	6	12	24
31476	48	96	192	0	0	18
31482	68	126	126	68	80	80