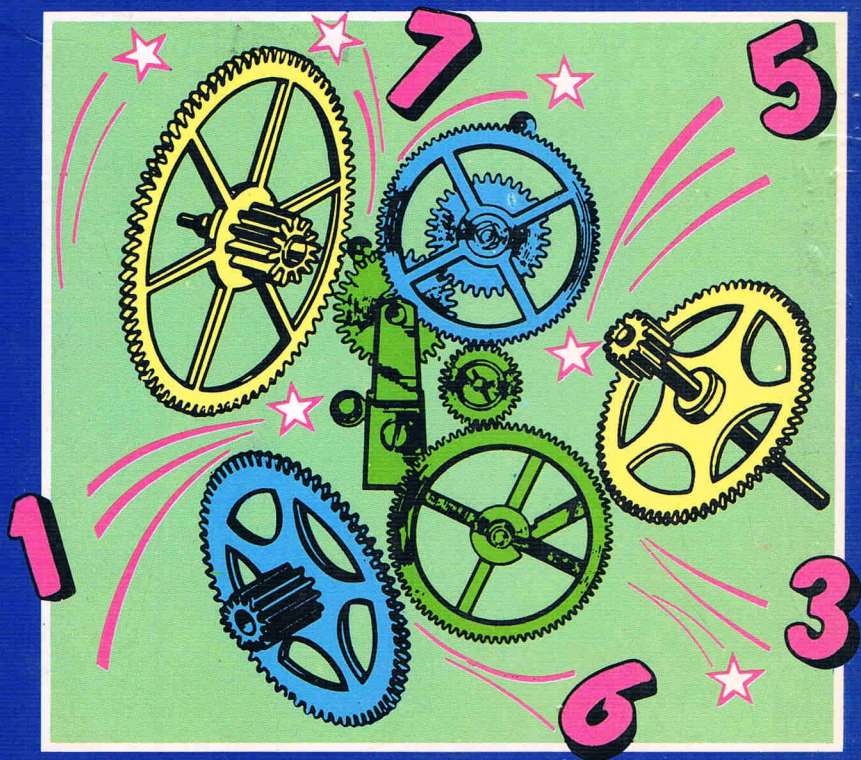


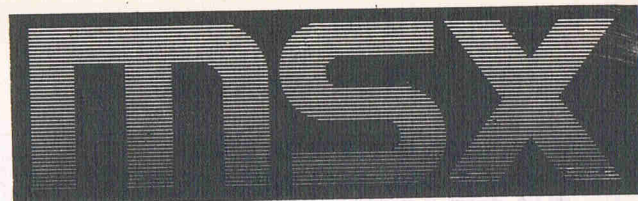
# MSX

## 30 rutinas de utilidad en Código Máquina.

Steve Webb

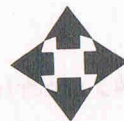


ra-ma



# *30 rutinas de utilidad en Código Máquina.*

*Steve Webb*



**TCA-MA**

Título de la obra original

USEFUL UTILITIES FOR YOUR MSX

Publicado en Gran Bretaña en 1985  
por Virgin Books Ltd, 328 Kensal Road,  
London W10 5XJ.

Copyright Steve Webb 1985

ISBN 086369 084 X

Edición en Español

MSX 30 RUTINAS DE UTILIDAD  
EN CODIGO MAQUINA

1986 RAMA

Editado por RAMA  
Ctra. de Canillas, 144  
28043 MADRID  
ESPAÑA

Depósito Legal: M. 6582 - 1986  
I. S. B. N.: 84 - 86381 - 11 - 8  
Imprime: Signo Impresores, S. A.  
Albasanz, 27. 28037 MADRID

Reservados todos los derechos en lengua española.  
No está permitida la reproducción parcial o total  
de este libro sin consentimiento por escrito del  
editor.

Consultas referentes al libro  
RAMA, Ctra. Canillas, 144 - 28043 Madrid -  
Teléfono: (91) 200.97.46/47

Traducción y Composición: CONORG, S.A.

---

## Contenidos

---

Introducción	5
La Memoria	9
Uso de las Subrutinas	23
Las Subrutinas	27
Apéndices	87



---

# Introducción

---

Este libro contiene más de 30 subrutinas en Código Máquina que puedes aprovechar para mejorar tus programas en BASIC, particularmente los programas de juegos.

Las rutinas no hacen nada que no puedas conseguir en BASIC, pero la ventaja es que son hasta 100 veces más rápidas y generalmente usan menos memoria que la subrutina equivalente en BASIC.

Con estas subrutinas serás capaz de hacer cosas en tus programas que pueden haberte estado prohibidas debido a la lentitud del BASIC. Por ejemplo, ahora serás capaz de efectuar corrimientos de imagen en cualquier dirección, con una rapidez muy elevada, simplemente llamando a la rutina apropiada.

Aseguro al lector que no necesita en absoluto ningún conocimiento de Código Máquina para hacer un uso eficaz de estas rutinas; le mostraré exactamente cómo introducir las rutinas en sus programas y cómo utilizarlas.

La eficacia de las rutinas no puede describirse por completo y necesitas verlas trabajar realmente antes de que puedas apreciar la velocidad de los programas en Código Máquina.

Así que prepara tus dedos tecladores y comencemos.





# Introducción

Este libro contiene más de 30 minutos en Código Máquina que pueden ser usados para enseñar los programas en BASIC, particularmente los programas de lectura.

Las rutinas no hacen nada que no puedan hacer los programas BASIC, pero la ventaja es que son fáciles de usar y generalmente usan menos memoria que las rutinas equivalentes en BASIC.

Con estas rutinas será capaz de hacer cosas que los programas que pueden hacer los estados existentes debido a la limitación del BASIC. Por ejemplo, podrá hacer cosas de lectura con rutinas de entrada y salida de datos, con una rutina para leer y escribir datos, con una rutina para leer y escribir datos, con una rutina para leer y escribir datos.

Asegure el lector que no necesita un programa para leer y escribir datos. El programa para leer y escribir datos es un programa que puede leer y escribir datos. El programa para leer y escribir datos es un programa que puede leer y escribir datos.

La eficacia de las rutinas no puede ser medida por la cantidad de memoria que ocupan, sino por la cantidad de memoria que ocupan. La eficacia de las rutinas no puede ser medida por la cantidad de memoria que ocupan, sino por la cantidad de memoria que ocupan.

Así que prepare sus datos y comience.



# La Memoria

La memoria es el lugar donde se almacenan los datos y los programas que se ejecutan en el computador. La memoria es el lugar donde se almacenan los datos y los programas que se ejecutan en el computador.

## La Memoria

La memoria es el lugar donde se almacenan los datos y los programas que se ejecutan en el computador. La memoria es el lugar donde se almacenan los datos y los programas que se ejecutan en el computador. La memoria es el lugar donde se almacenan los datos y los programas que se ejecutan en el computador.

- Tabla 1
- Tabla 2
- Tabla 3
- Tabla 4
- Tabla 5
- Tabla 6
- Tabla 7
- Tabla 8
- Tabla 9
- Tabla 10

---

# La Memoria

---

En consideración a los principiantes en informática, computadoras y programación, describiré brevemente la memoria de las computadoras y como los programas se depositan en ella.

Si observas el siguiente diagrama, verás que la memoria tiene sus celdillas numeradas desde la 0 hasta la 65535 (si es de 64K). Eso representa el número total de celdillas de memoria que es directamente accesible. La memoria está dividida en dos partes principales: la primera parte, desde la dirección 0 hasta la 32767, se conoce con el nombre abreviado de Memoria de Unicamente Lectura, y se conoce mucho más por la abreviatura inglesa ROM:=Read Only Memory. Esta Memoria de Unicamente Lectura puede considerarse como el director de todos los órganos que componen la computadora. Es donde reside el programa interpretador BASIC que realmente ejecuta los programas del usuario, comprueba la sintaxis de cada uno de sus comandos y realiza las funciones mandadas.

**Celdilla 0**

**ROM**

**Celdilla 32767**

**Celdilla 32768**

**RAM**

**Celdilla 65535**

La segunda parte, que ocupa las direcciones desde la 32768 hasta la 65535 es la memoria denominada de usuario o mejor Memoria de Escritura y Lectura (MEL).

Por la inercia que tiene el lenguaje, seguimos aprovechando el término abreviado inglés RAM que corresponde a la abreviatura de "Read and Alterable Memory". (Ya han sido varias las voces que se han alzado protestando por llamarla Random Access Memory, debido primordialmente a que, en cuanto al **acceso**, tanto la Memoria de Únicamente Lectura como la Memoria de Escritura y Lectura son de la misma naturaleza: acceso directo, acceso por referencia, es decir: acceso conociendo la dirección de la celdilla donde se quiere efectuar una operación.

Tus programas en BASIC quedan depositados en esa Memoria de Escritura y Lectura, en esa memoria de usuario. Si escribes un programa en BASIC que sólo tenga dos o tres líneas cortas de instrucciones, el comienzo de dicho programa estará ubicado en la celdilla de dirección 32768 y el final estará más o menos alrededor de la 32820, dependiendo de la longitud del programa. A medida que se incrementa el tamaño de tu programa, el final de él va acercándose más y más a la última celdilla, la de dirección 65535.

Puedes pensar por tanto que dispones de 32768 celdillas de memoria en las que almacenar tus programas. Desafortunadamente, siempre tienes algo menos que eso. La computadora por sí misma, o mejor el programa interpretador BASIC que ejecuta tu programa de usuario, usa para el mismo la zona de memoria a partir, aproximadamente, de la dirección 61500 hasta el final. Vienen a ser un poco menos de 4K (y recuerda que la K, el kilo informático es de 1024 y no de 1000).

Este área de memoria es la que se denomina área de trabajo del sistema. Por lo tanto, el área de memoria desde la dirección 32768 hasta aproximadamente la 61499 está disponible, está **libre**, para usarla en tus programas; esa es la razón por la que el mensaje de **28815 octetos libres**, que en inglés es 28815 "BYTES FREE" aparece en la pantalla cuando enciendes tu ordenador y el interpretador BASIC te **saluda**.

La celdilla 61500 en este contexto es la que constituiría la cima de la memoria de usuario, llamada normalmente RAMTOP y representa la máxima dirección que puede alcanzar un programa en BASIC. Si tecleas un programa de la suficiente longitud obtendrás finalmente un mensaje de error diciendo **"FUERA DE MEMORIA"**, significando que el final de tu programa ha alcanzado esa dirección y que por tanto no queda memoria disponible para depositar ninguna línea más de un programa en BASIC.

Cuando estás tecleando programas en BASIC, no tienes por qué preocuparte de dónde están ubicadas en la memoria cada una de las líneas de tu programa. Al programar en Código Máquina es importantísimo saber exactamente dónde está cada una de las subrutinas e instrucciones en la memoria; por lo tanto, reservaremos zonas específicas de la memoria para cada una de nuestras subrutinas. Por ejemplo, podemos decir que la rutina nº 1 comienza a partir de la celdilla 58000 y ocupa 60 celdillas consecutivas; por lo tanto la rutina nº 2 puede comenzar a partir de la celdilla 58060 y así sucesivamente. Si simplemente depositáramos nuestras instrucciones en Código Máquina en la memoria, no estarían muy seguras dado que a medida que el programa en BASIC se hace más largo podría llegar a escribir encima de las rutinas en Código Máquina.



De manera que ha de encontrarse un procedimiento para depositar los programas en BASIC y los programas en Código Máquina sin correr el riesgo de que el programa en BASIC llegue a ocupar las mismas celdillas que el programa en Código Máquina.

Recordarás que ya dije que el final de un programa en BASIC nunca puede sobrepasar la celdilla que actúa como cima de la memoria utilizable (RAMTOP); afortunadamente además podemos mover fácilmente ese tope superior, simplemente tecleando el comando para que **anule** los datos presentes, con la clave BASIC **CLEAR** (aclarar, cancelar, etc.) que como sabemos nos permite reservar una zona de memoria como depósito de los literales empleados en el programa, y además señalar la cima de memoria utilizable por el programa. Por lo tanto si queremos situarla en la celdilla 56000 por ejemplo, podríamos teclear:

**CLEAR 200,56000**

Una vez que hemos desplazado la cima de la memoria utilizable a la celdilla 56000, ningún programa en BASIC podrá ser capaz de sobrepasar ese tope. Podemos por tanto usar el área de memoria a partir de esa dirección 56000 para colocar nuestras rutinas en Código Máquina, con lo que estaremos seguros ya de que el programa en BASIC nunca las destruirá. Podemos, de hecho, mover esta cima de la memoria virtualmente a cualquier punto que deseemos; el comando **CLEAR 200, 45321**, por ejemplo, colocará en esa celdilla el límite superior de la memoria de usuario BASIC.

Las rutinas en Código Máquina son bastante difíciles de colocar en la memoria: no podemos simplemente teclearlas. Tenemos que aprovecharnos de un programa en BASIC que nos haga la introducción de las instrucciones en Código Máquina mucho más cómodas. Teclea el programa que te presentamos y compruébalo cuidadosamente; luego manda a la computadora que lo guarde (SAVE) en la cinta usando el siguiente comando, por ejemplo:

**SAVE"CAS:MCENT"**

```
10 CLEAR 200,56000
20 CLS
30 LOCATE 0,0
40 PRINT 'MANTENLO SIEMPRE EN MAYUSCULAS'
50 LOCATE 0,4
60 PRINT 'PULSA LA I PARA INTRODUCIR LA RUTINA (HEX-CODE)'
70 LOCATE 0,8
80 PRINT 'PULSA LA C PARA COMPROBAR LAS INSTRUCCIONES'
90 LOCATE 0,12
100 PRINT 'PULSA LA T PARA COMPROBAR LA SUMA'
110 LOCATE 0,16
120 PRINT 'PULSA LA Q PARA PARAR'
130 A$=INKEY$
140 IF A$='I' THEN GOTO 360
150 IF A$='C' THEN GOTO 650
160 IF A$='T' THEN GOTO 190
170 IF A$='Q' THEN STOP
180 GOTO 70
190 CLS
200 LOCATE 0,0
210 PRINT 'DAME LA DIRECCION DE COMIENZO'
220 INPUT SA
230 LOCATE 0,5
240 PRINT 'DAME LA DIRECCION DE FINAL'
250 INPUT EA
260 LET D=0
270 FOR C=SA TO EA
280 LET D=D+PEEK(C)
290 NEXT C
300 CLS
310 PRINT 'TOTAL SUMA=';D
320 LOCATE 0,20
330 PRINT 'PULSA LA M PARA VOLVER AL MENU'
340 IF INKEY$<>'M' THEN GOTO 340
350 GOTO 20
360 CLS
370 LOCATE 0,0
380 PRINT 'MANTENLO SIEMPRE EN MAYUSCULAS'
390 LOCATE 0,4
400 PRINT 'DAME LA DIRECCION DE COMIENZO'
```

```

410 INPUT S
420 IF S<56000 THEN GOTO 630
430 LET A$=""
440 LOCATE 0,23
450 LET ET=S
460 IF A$="" THEN INPUT A$
470 LET BAD=0
480 IF A$='M' THEN GOTO 20
490 LET C=VAL('&H'+A$)
500 IF C=0 THEN GOTO 610
510 IF BAD=1 THEN GOTO 580
520 LOCATE 0,21:PRINT S;'';A$
530 LET C=VAL('&H'+A$)
540 POKE(S),C
550 LET S=S+1
560 GOTO 430
580 LOCATE 0,22:PRINT 'INCORRECTO INTENTALO DE NUEVO'
590 LET S=ET
600 GOTO 430
610 IF A$<>'00' THEN LET BAD=1
620 GOTO 510
630 PRINT 'LA DIRECCION DE COMIENZO DEBE SER 56000 O MAYOR.'
640 GOTO 400
650 LET ND=0
660 CLS
670 LOCATE 0,0
680 PRINT 'DAME LA DIRECCION DE COMIENZO'
690 INPUT SA
700 LOCATE 0,5
710 PRINT 'DAME LA DIRECCION DE FINAL'
720 INPUT EA
730 CLS
740 IF SA+20>EA THEN GOTO 870
750 FOR C=SA TO SA+20
760 IF PEEK(C)<16 THEN GOTO 850
770 PRINT C;'';HEX$(PEEK(C))
780 NEXT C
790 IF ND=1 THEN GOTO 900
800 IF C>EA THEN GOTO 505
810 PRINT 'PULSA M PARA CONTINUAR'
820 LET SA=C
830 IF INKEY$<>'M' THEN GOTO 830
840 GOTO 740
850 PRINT C;'';HEX$(PEEK(C))
860 GOTO 780
870 FOR C=SA TO EA
880 LET ND=1
890 GOTO 760
900 PRINT 'PULSA LA M PARA VOLVER AL MENU'
910 IF INKEY$<>'M' THEN GOTO 910
920 GOTO 20

```

Posteriormente cuando queramos introducir una cualquiera de las subrutinas en Código Máquina del libro, debemos primero mandar a la computadora que traiga del cassette y cargue (LOAD) el programa anterior, usando el siguiente comando:

**LOAD "CAS:MCENT",R**

Recuerda que la "R" final de este comando, indica al ordenador que automáticamente después de traerlo del cassette ha de pasar a **ejecutarlo** (RUN), y en consecuencia aparecerá en pantalla el menú que nos ofrece las diversas tareas posibles a realizar por el programa:

**PULSA TECLA I PARA INTRODUCIR CODIGO  
PULSA TECLA C PARA COMPROBAR CODIGO  
PULSA TECLA T PARA COMPROBAR LA SUMA  
PULSA TECLA Q PARA PARAR PROGRAMA**

Veamos ahora exactamente cómo podemos imponer del ordenador las instrucciones en Código Máquina de cada subrutina, cómo podemos comprobar lo que hemos tecleado, cómo podemos mandar que lo guarde en cinta, y luego cómo podemos usar la primera subrutina que presentamos en el libro. Los principios involucrados en la introducción de esta primera subrutina, se aplican también a todas las otras subrutinas del libro, por lo que te sugiero que trabajes muy cuidadosamente siguiendo este ejemplo y asegurándote que lo comprendes completamente antes de proseguir con el resto de las subrutinas del libro.



Para ingresar en la memoria de la computadora las instrucciones que forman una subrutina en Código Máquina, debes pulsar -usando este programa cargador- la tecla "I". Se te pedirá que te asegures que el teclado está **enclavado a mayúsculas** (CAPS LOCK), y puede que en tu ordenador se detecte mediante el piloto luminoso correspondiente. Luego se te pedirá que ingreses la dirección en memoria a partir de la cual quieres colocar esa subrutina; eso se menciona al principio de cada una de las subrutinas del libro, y en el caso de la rutina 1 es la dirección 56000.

Ahora el programa ya está preparado para que ingreses una a una las instrucciones en Código Máquina que componen la rutina. Si observas la primera del libro, verás que hay varias ristas horizontales y verticales, y en cada ristra horizontal aparecen 8 pares de cifras y letras. La primera pareja de la primera subrutina es la 21. Así que debes teclear 21 y como siempre pulsar la tecla marcada **ENTER** o la marcada **RETURN** en otros modelos.

La segunda pareja de caracteres es 00, así que teclea 00 y luego **ENTER**. Continúa de esa manera tecleando todas las parejas de caracteres, cifras y números, hasta llegar a la última pareja del primer renglón que tiene las letras AF; también tecléala y pulsa **ENTER**.

Pasa ahora a la segunda ristra horizontal y haz exactamente lo mismo comenzando con la primera pareja de caracteres que es FC, teclea todas las parejas que aparecen de la misma manera. Observarás que en todas las rutinas aparece siempre al final la letra M. Esta también tienes que teclearla y pulsar **ENTER**, con lo que el programa reconocerá que estamos al final de la subrutina y volverá al menú de opciones.

El paso siguiente es comprobar que hemos introducido las instrucciones en Código Máquina de manera correcta. Esto se efectúa eligiendo en el menú de opciones la función correspondiente a la letra C. Se te pedirá entonces que ingreses la dirección de comienzo y la dirección de final del bloque de memoria donde está ubicada la subrutina que quieres comprobar; ambos parámetros están mencionados al comienzo de cada una de la subrutinas del libro. En el caso de la primera subrutina, la dirección de comienzo es la 56000 y la dirección de finalización del bloque es la 56088. El programa en BASIC efectuará ahora la tarea de ir mostrando el Código Máquina que acabas de ingresar por teclado; debes cotejarlo cuidadosamente con el listado del libro. Habiendo comprobado que lo ingresado es exactamente lo que aparece en el libro, puedes volver al menú principal pulsando la tecla M.

La última cosa que ha de hacerse ahora es lo que denominamos comprobación de la suma; y esa tarea la efectúa el programa eligiendo la opción **SIMBOLIZADA** por la tecla T. Se te pedirá también aquí que introduzcas la dirección de comienzo y la dirección donde finaliza el bloque de celdillas ocupado por la subrutina.

Después de una breve pausa, verás que aparece expuesto en pantalla un número que es el total de la suma de los valores contenidos en esas celdillas. El programa ha sumado los valores de todas las instrucciones en Código Máquina que has introducido por teclado y ese es el valor que expone. Ese valor total del bloque también se da al comienzo de cada listado de las subrutinas, y en el caso de la primera de ellas deberá aparecer el valor 10572 si todo ha sido llevado a cabo correctamente.



Todas estas comprobaciones y cotejos son muy importantes dado que -a diferencia de lo que sucede con un programa en BASIC que se detendrá cuando detecte un error- los programas en Código Máquina habitualmente acaban "corrompiendo" todo los datos que haya en la memoria, cuando no son correctos. Si durante estas comprobaciones, descubres que has cometido una equivocación al teclear el Código Máquina, todo lo que debes hacer es volver al menú principal y comenzar de nuevo con más cuidado.

Ahora que estás realmente seguro que las instrucciones en Código Máquina de la subrutina han sido correctamente tecleadas, debes mandar que lo guarde en cinta. Para hacer esto, debes primero aprovechar la opción simbolizada con la letra Q para hacer que el programa en BASIC finalice su actuación. Luego, para que la computadora guarde en la cinta del cassette la subrutina tecleada, debes dar el comando:

**BSAVE "CAS:NOM", D\_Com, D\_F\***

Y recuerda que la letra "B" de la clave BSAVE, es la que indica al ordenador que se trata en este caso de guardar un **bloque** de memoria, llevan a la cinta los datos contenidos en la celdilla de ese bloque, y que el bloque comienza en la dirección dada y termina en la dirección dada.

En el caso de la primera subrutina del libro que acabamos de ingresar en la memoria del ordenador mediante el teclado, deberemos teclear el comando:

**BSAVE "CAS:RUT1",56000,56088**

Si con ese nombre de RUT1 queremos que quede grabada en la cinta.

Te aconsejo que mandes guardar en cinta cada subrutina como mínimo dos veces y en cintas separadas; eso te evitará tener que volver a teclear una subrutina en el caso de que a una de las cintas le ocurra algún percance o se vea dañada.

Uso de las Subrutinas



---

## Uso de las Subrutinas

---

Te mostraré ahora cómo usar la rutina primera del libro en cualquiera de tus programas en BASIC. Primero apaga tu ordenador y luego enciéndelo de nuevo; con eso te aseguras que la memoria está completamente en "blanco". Antes de mandar al ordenador que traiga del cassette una de las subrutinas en Código Máquina, debes siempre teclear el comando que anula datos y prepara la cima de la memoria utilizable por el ordenador en la forma ya vista. Es decir, el comando:

**CLEAR 200,56000**

Ahora puedes mandar que traiga del cassette una sola o varias rutinas sucesivamente. Para que traiga la primera subrutina -que ya previamente habíamos guardado en la cinta cassette, simplemente tecleamos el comando:

**BLOAD "CAS:RUT1"**

Para demostrar que la rutina en Código Máquina funciona adecuadamente, y al mismo tiempo ver cómo la usamos en un programa en BASIC, necesitaremos confeccionar un pequeño programa en BASIC de prueba. En lo que se refiere a la primera rutina que efectúa la tarea de correr la imagen expuesta en pantalla, escribiremos primero un programa que llena la pantalla de determinados caracteres y luego llamaremos a la subrutina para correr cierta parte de la pantalla.



El programa en BASIC que necesitamos se muestra a continuación:

```
5 SCREEN 1
10 LET X=65
20 FOR A=6144 TO 6719
30 VPOKE(A),X
40 LET X=X+1
50 IF X=90 THEN LET X=65
60 NEXT A
70 POKE 62000,10
75 POKE 62001,5
80 POKE 62002,11
85 POKE 62003,6
90 DEF USR=56000
100 B=USR(0)
110 GOTO 100
```

Las líneas 10 a 60 del programa anterior son las encargadas de rellenar la pantalla con caracteres. Las líneas 70 a 85 dicen a la subrutina qué parte de la imagen en pantalla queremos correr. Las líneas 90 y 100 son las instrucciones en BASIC que se requieren para llamar a una rutina de **usuario** en Código Máquina, simplemente sabiendo la dirección en memoria a partir de la cual está ubicada dicha subrutina. La línea 110 simplemente manda que **vaya a 100** con lo que hacemos que la imagen en pantalla esté continuamente corriendo.

Si lo deseas, también puedes mandar al intérprete BASIC que guarde en cinta este pequeño programa en BASIC (si has escrito tu propio programa en BASIC, éste es también el momento de guardarlo en cinta). Puedes ahora mandar al intérprete que **ejecute** (RUN) el programa y verás cómo la pantalla se llena de caracteres; luego comienza a efectuar el corrimiento una determinada zona específica de la imagen. Si el corrimiento es demasiado rápido de manera que no puedas apreciar lo que está sucediendo, siempre puedes ralentizar todo el programa insertando el siguiente **bucle de retardo**:

```
105 FOR C = 1 TO 1000:NEXT C
```

Ensayá alternando los valores de las líneas 70 a 85 del programa para efectuar el corrimiento de otras partes diferentes de la imagen.

Con cada una de las subrutinas en Código Máquina del libro, incluiré detalles de cómo comprobarla y con eso adquirirás confianza en la subrutina antes de intentar aprovecharla en tus propios programas. Te aconsejo que recorras todo el libro y vayas tecleando cada una de las subrutinas y luego guardándolas en cinta. Además de comprobar cada una de las subrutinas de acuerdo con mis instrucciones, también deberás ensayar y confeccionar tus propios programas en BASIC que aprovechen la subrutina. Eso te dará confianza en ellas cuando llegue la hora de utilizarlas en un programa mayor.



**Nombre:** RUT1

**Función:** Corre el texto hacia arriba.

**Dirección de comienzo** ..... 56000

**Dirección de final** ..... 56088

**Suma total** ..... 10572

**Requisitos:** La celdilla 62000 debe contener el número de columna correspondiente a la esquina superior izquierda del área de pantalla que se va a correr. En el modo de pantalla 1, ese número deberá estar comprendido entre 1 y 32.

La celdilla 62001 debe contener el número de fila correspondiente a la esquina superior izquierda del área de pantalla que se va a correr. En el modo 1 de pantalla, ese número deberá estar comprendido entre 1 y 24.

La celdilla 62002 debe contener un número entre 1 y 24 que representa la altura del recuadro de pantalla donde va a correrse la imagen.

La celdilla 62003 debe contener un número entre 1 y 32 que representa la anchura del recuadro de pantalla donde va a correrse la información expuesta.



**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	D0	DA	01	28	00
3A	31	F2	3D	CA	DB	DA	09
C3	D3	DA	11	00	00	3A	30
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	33	F2	F5	CD
4A	00	32	40	F2	3A	32	F2
3D	F5	09	CD	4A	00	ED	42
CD	4D	00	09	F1	3D	C2	F9
DA	3A	40	F2	CD	4D	00	13
D5	E1	F1	3D	C2	EE	DA	C9
01	M						

**Nombre:** RUT2

**Función:** Corre el texto hacia abajo

**Dirección de comienzo** ..... 56100

**Dirección de final** ..... 56191

**Suma total** ..... 10442

**Requisitos:** La celdilla 62004 debe contener el número de columna correspondiente a la esquina inferior izquierda de la zona que va a correrse.

La celdilla 62005 debe contener el número de fila correspondiente a la esquina inferior izquierda de dicha área de corrimiento.

La celdilla 62006 debe contener la altura de dicha área.

La celdilla 62007 debe contener la anchura de dicha área.

**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	34	DB	01	28	00
3A	35	F2	3D	CA	3F	DB	09
C3	37	DB	11	00	00	3A	34
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	37	F2	F5	CD
4A	00	32	40	F2	3A	36	F2
3D	F5	A7	ED	42	CD	4A	00
09	CD	4D	00	A7	ED	42	F1
3D	C2	5D	DB	3A	40	F2	CD
4D	00	13	D5	E1	F1	3D	C2
52	DB	C9	02	M			

Nombre: RUT3

Función: Corre el texto hacia la izquierda.

Dirección de comienzo ..... 56200

Dirección de final ..... 56289

Suma total ..... 10551

**Requisitos:** La celdilla 62008 debe contener el número de columna correspondiente a la esquina superior izquierda del recuadro de pantalla donde va a correrse la imagen expuesta.

La celdilla 62009 debe contener el número de fila de la esquina superior izquierda de dicha área.

La celdilla 62010 debe contener la altura de dicha área.

La celdilla 62011 debe contener la anchura de dicha área.

**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	98	DB	01	28	00
3A	39	F2	3D	CA	A3	DB	09
C3	9B	DB	11	00	00	3A	38
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	3A	F2	F5	CD
4A	00	32	40	F2	3A	3B	F2
3D	F5	23	CD	4A	00	2B	CD
4D	00	23	F1	3D	C2	C1	DB
3A	40	F2	CD	4D	00	C5	E1
19	E5	D1	F1	3D	C2	B6	DB
C9	03	M					

Nombre: RUT4

Función: Corre el texto a la derecha

Dirección de comienzo ..... 56300

Dirección de final ..... 56389

Suma total ..... 10311

**Requisitos:** La celdilla 62012 debe contener el número de columna correspondiente a la esquina superior derecha del área donde va a correrse la imagen expuesta.

La celdilla 62013 es el número de fila correspondiente a la esquina superior derecha de dicha área.

La celdilla 62014 debe contener la altura de dicha área.

La celdilla 62015 debe contener la anchura de dicha área.

**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	FC	DB	01	28	00
3A	3D	F2	3D	CA	07	DC	09
C3	FF	DB	11	00	00	3A	3C
F2	3D	5F	19	ED	5B	22	F9
19	E5	D1	3A	3E	F2	F5	CD
4A	00	32	40	F2	3A	3F	F2
3D	F5	2B	CD	4A	00	23	CD
4D	00	2B	F1	3D	C2	25	DC
3A	40	F2	CD	4D	00	C5	E1
19	E5	D1	F1	3D	C2	1A	DC
C9	04	M					



El siguiente programa en BASIC puede usarse para comprobar las rutinas 1, 2, 3 y 4. El programa rellena primero la pantalla con texto; luego efectúa el corrimiento de imagen en cuatro áreas diferentes de la pantalla aprovechando estas cuatro subrutinas.

```

10 SCREEN 1
20 LET X=65
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=90 THEN X=65
70 NEXT A
80 POKE 62000,3
90 POKE 62001,2
100 POKE 62002,5
110 POKE 62003,7
115 POKE 62004,20
120 POKE 62005,10
130 POKE 62006,7
140 POKE 62007,3
150 POKE 62008,16
160 POKE 62009,15
170 POKE 62010,4
180 POKE 62011,9
190 POKE 62012,25
200 POKE 62013,22
210 POKE 62014,2
230 POKE 62015,6
240 DEF USR=56000
250 A=USR(0)
260 DEF USR=56100
270 A=USR(0)
280 DEF USR=56200
290 A=USR(0)
300 DEF USR=56300
310 A=USR(0)
330 GOTO 240

```

Nombre: RUT5

Función:

Apelando a esta subrutina, las formas de todos los caracteres serán giradas 90°. Por lo tanto, citando la rutina una vez, todos los caracteres serán girados sobre sus costados. Citándola por segunda vez, los caracteres girarán otros 90° adicionales, haciendo que estén boca abajo. Esta rutina puede usarse cuando se desee escribir texto lateralmente de abajo arriba en la pantalla.

Dirección de comienzo ..... 56400  
Dirección de final ..... 56471  
Suma total ..... 8491

Limitaciones: Se usa sólo en el modo 1 de pantalla.

21	00	00	3E	FF	F5	E5	3E
08	F5	CD	78	DC	F1	23	3D
C2	59	DC	E1	E5	13	EB	01
08	00	CD	5C	00	E1	01	08
00	09	F1	3D	C2	55	DC	C9
11	62	F2	CD	4A	00	47	3E
08	F5	CD	8F	DC	F1	3D	C2
81	DC	C9	CD	4A	00	47	1A
CB	10	CB	17	12	1B	C9	05
M							



## Nombre: RUT6

**Función:** Apelando a esta rutina, todos los caracteres expuestos sobre un determinado fondo volverán a ser expuestos con un color nuevamente especificado. Por ejemplo, si deseas que cambien los caracteres que tienen un fondo verde a un fondo amarillo, puedes hacerlo simplemente llamando a esta rutina.

Dirección de comienzo ..... 56500  
Dirección de final ..... 56535  
Suma total ..... 4708

**Requisitos:** La celdilla 62017 debe contener el número identificativo que representa el color del fondo que se desea sustituir.

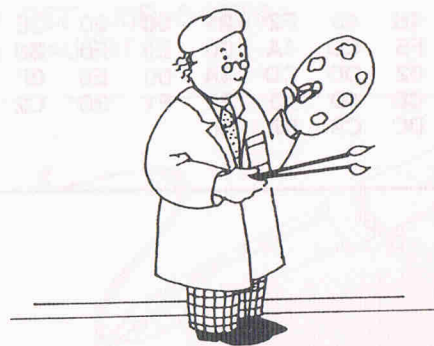
La celdilla 62018 debe contener un número que identifique el color del nuevo fondo reemplazante. (Consulta el apéndice para encontrar los números que identifican los diversos colores).

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

ED	4B	41	F2	21	00	20	3E
20	F5	CD	4A	00	E6	0F	B8
C2	D0	DC	CD	4A	00	E6	F0
B1	CD	4D	00	23	F1	3D	C2
BD	DC	C9	06	M			

El siguiente programa puede utilizarse para demostrar la rutina anterior. El programa rellena primero la pantalla con caracteres con diversos colores de fondo. Luego, después de una breve pausa, todos los caracteres con un fondo blanco se volverán a exponer con un color verde para el fondo.

```
10 SCREEN 1
20 LET X=0
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=256 THEN LET X=0
70 NEXT A
80 FOR X=17 TO 31
90 VPOKE(8190+X),X
100 NEXT X
110 FOR X=17 TO 31
120 VPOKE(8190+X),X
130 NEXT X
140 FOR X=1 TO 1000:NEXT X
150 POKE 62017,12
160 POKE 62018,15
170 DEF USR=56500
180 A=USR(0)
```



**Nombre: RUT7**

**Función:** Apelando a esta rutina todos los caracteres que están expuestos con un color específico de pluma volverán a exponerse con un nuevo color especificado.

Dirección de comienzo ..... 56550  
 Dirección de final ..... 56585  
 Suma total ..... 4556

**Requisitos:** La celdilla 62019 debe contener el número identificativo que representa el color de la pluma que deseamos sustituir.

La celdilla 62020 debe contener el número que identifica el color de la nueva pluma. (Consulta el apéndice para ver cuáles son los números que identifican los diversos colores).

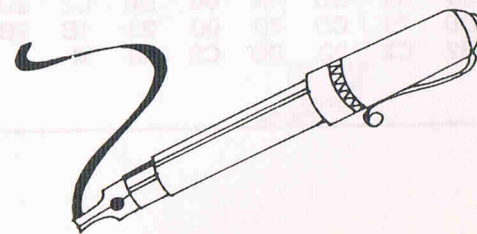
**Limitaciones:** Se usa en el modo 1 de pantalla.

ED	4B	43	F2	21	00	20	3E
20	F5	CD	4A	00	E6	F0	B8
C2	02	DD	CD	4A	00	E6	0F
B1	CD	4D	00	23	F1	3D	C2
EF	DC	C9	07	M			

El siguiente programa en BASIC puede aprovecharse para demostrar la rutina anterior. El programa rellena primero la pantalla con caracteres que están expuestos con diferentes colores de pluma. Después de una breve pausa, todos los caracteres expuestos en rojo volverán a ser expuestos usando una pluma de tinta blanca.

```

10 SCREEN 1
20 LET X=0
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=256 THEN LET X=0
70 NEXT A
80 LET X=21
90 FOR A=0 TO 14
100 VPOKE(8192+A),X
110 LET X=X+16
120 NEXT A
130 LET X=21
140 FOR A=0 TO 14
150 VPOKE(8208+A),X
160 LET X=X+16
170 NEXT A
180 FOR X=1 TO 1000:NEXT X
190 POKE 62019,240
200 POKE 62020,96
210 DEF USR=56550
220 A=USR(0)
  
```





**Nombre: RUT8****Función:**

Quando se apela a esta rutina se sustituirán todas las apariciones de un determinado carácter con un nuevo carácter especificado. Por ejemplo, cada vez que aparece la letra "E" en la pantalla será sustituida por la letra "X".

Dirección de comienzo ..... 56600

Dirección de final ..... 56629

Suma total ..... 3047

**Requisitos:** La celdilla 62021 debe contener el código ASCII del carácter que deseamos sustituir.

La celdilla 62022 debe contener el código ASCII de la letra reemplazante de la anterior. (La lista completa de los caracteres y su código ASCII se da en el apéndice).

**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

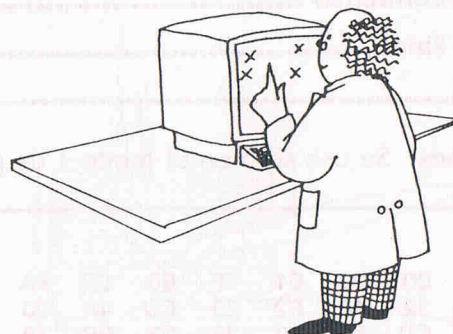
21	00	18	ED	4B	45	F2	11
00	03	CD	4A	00	B8	C2	2D
DD	79	CD	4D	00	23	1B	7B
B2	C2	22	DD	C9	08	M	

El siguiente programa en BASIC puede aprovecharse para demostrar el funcionamiento de la rutina anterior. La pantalla se rellenará primero con diversos caracteres; luego, después de una breve pausa, cada aparición de la letra "A" será reemplazada con la letra "X".

```

10 SCREEN 1
20 LET X=65
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=90 THEN LET X=65
70 NEXT A
80 FOR X=1 TO 1000:NEXT X
90 POKE 62021,88
100 POKE 62022,65
110 DEF USR=56600
120 A=USR(0)

```





**Nombre: RUT9****Función:**

En el modo 1 de pantalla es posible disfrutar de 32 combinaciones de colores simultaneamente para la pluma y el papel del texto. Estas 32 combinaciones están depositadas en 32 celdillas contiguas de la memoria de vídeo. La primera celdilla determina el color para los caracteres de código 0 a 7. La segunda determina el color para los caracteres de código 8 a 15; y así sucesivamente hasta la celdilla 32 que determinará el color de los caracteres con código 248 a 255. Esta rutina hará efectivamente que se corran los colores correspondientes a cada bloque de 8 caracteres. El primer bloque de 8 caracteres adoptará por tanto el color del segundo bloque de ocho caracteres. El segundo bloque adoptará el color del tercer bloque de 8 caracteres, y así sucesivamente.

Dirección de comienzo ..... 56650  
 Dirección de final ..... 56686  
 Suma total ..... 3150

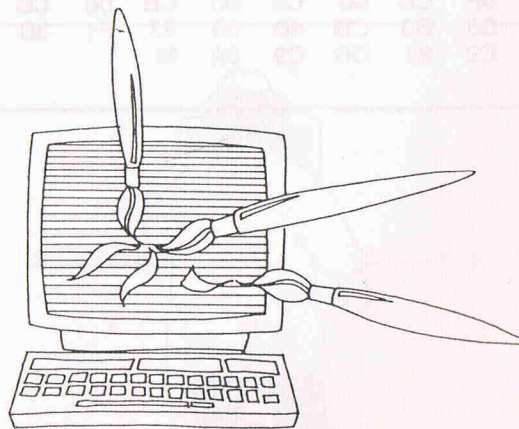
**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

21	00	20	01	1F	00	CD	4A
00	32	40	F2	23	CD	4A	00
2B	CD	4D	00	23	23	0B	78
B1	C2	57	D0	2B	3A	40	F2
CD	4D	00	C9	09	M		

El siguiente programa en BASIC demostrará el funcionamiento de la rutina anterior. En la pantalla se exponen primeramente diversos caracteres con colores de pluma diferentes. Luego, después de una breve pausa se apelará a la subrutina y se efectuará el corrimiento de los colores.

```

10 SCREEN 1
20 LET X=0
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=256 THEN LET X=0
70 NEXT A
80 LET X=17
90 FOR A=0 TO 31
100 VPOKE(8192+A),X
110 X=X+3
120 NEXT A
130 FOR X=1 TO 1000:NEXT X
140 DEF USR=56650
150 A=USR(0)
160 GOTO 150
  
```



## Nombre: RUT10

**Función:** Esta rutina canjea los colores de tinta-pluma (vanguardia) y de papel (retaguardia) de todos los caracteres. De esta manera, si se expone un carácter con tinta blanca sobre un fondo negro, después de apelar a esta rutina, el carácter se expondrá con tinta negra sobre un fondo blanco.

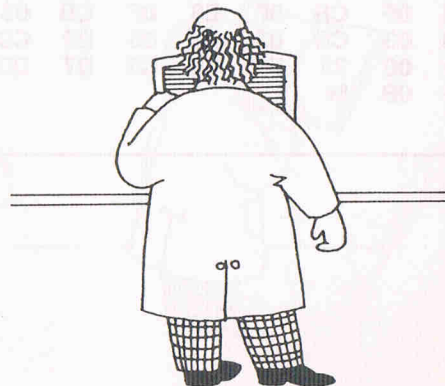
Dirección de comienzo ..... 56700  
Dirección de final ..... 56744  
Suma total ..... 4757

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

21	00	20	3E	20	F5	CD	4A
00	E6	0F	47	CD	4A	00	CB
0F	CB	0F	CB	0F	CB	0F	E6
0F	CB	00	CB	00	CB	00	CB
00	B0	CD	4D	00	23	F1	3D
C2	81	DD	C9	0A	M		

El siguiente programa en BASIC puede aprovecharse para demostrar el funcionamiento de la rutina anterior. La pantalla se rellenará con diversos caracteres de varios colores; luego, después de un breve momento, se apelará a la subrutina y verás cómo los colores de la pluma y del papel son intercambiados.

```
10 SCREEN 1
20 LET X=0
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=256 THEN LET X=0
70 NEXT A
80 LET X=17
90 FOR A=0 TO 31
100 VPOKE(8192+A),X
110 X=X+3
120 NEXT A
130 FOR X=1 TO 1000:NEXT X
140 DEF USR=56700
150 A=USR(0)
160 FOR X=1 TO 300:NEXT X
170 GOTO 150
```





**Nombre: RUT11**

**Función:** Esta rutina es muy similar a la RUTINA 10. Pero en lugar de canjear los colores de pluma y de papel para todos los caracteres, esta rutina sólo canjeará aquéllos que tengan un color específico de papel, de fondo.

**Dirección de comienzo** ..... 56750

**Dirección de final** ..... 56807

**Suma total** ..... 6823

**Requisitos:** La celdilla 62024 debe contener el número identificativo del color de fondo que deseamos intercambiar con el color del frente, de la pluma.

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

21	00	20	ED	4B	47	F2	3E
20	F5	CD	4A	00	E6	0F	B8
C2	E0	DD	CD	4A	00	E6	0F
5F	CD	4A	00	CB	0F	CB	0F
CB	0F	CB	0F	E6	0F	CB	03
CB	03	CB	03	CB	03	B3	CD
4D	00	23	F1	3D	C2	B7	DD
C9	0B	M					

El siguiente programa en BASIC puede usarse para comprobar la rutina anterior. Rellenará la pantalla con diversos caracteres de varios colores y luego, después de una breve pausa, los caracteres con un fondo blanco tendrán el color de la pluma y el color del papel canjeados.

```

10 SCREEN 1
20 LET X=0
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=256 THEN LET X=0
70 NEXT A
80 LET X=1
90 FOR A=0 TO 15
100 VPOKE(8192+A),X
110 X=X+1
120 NEXT A
130 FOR X=1 TO 1000:NEXT X
140 POKE 62023,240
150 DEF USR=56820
160 A=USR(0)

```





Nombre: RUT12

**Función:**

Esta rutina es muy similar a la rutina 10, pero en lugar de canjear los colores de pluma y papel de todos los caracteres, esta rutina sólo intercambiará los colores de aquéllos que tengan un color específico de pluma, de frente.

Dirección de comienzo ..... 56820  
Dirección de final ..... 56877  
Suma total ..... 6902

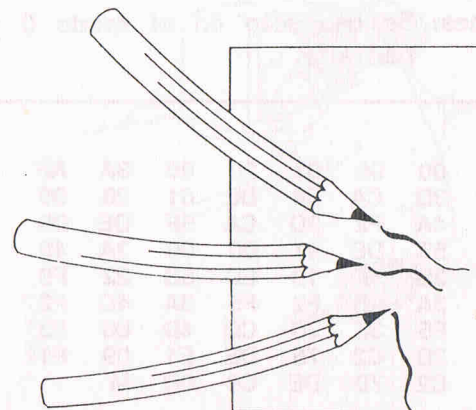
**Requisitos:** La celdilla 62023 debe contener el número que representa el color de la pluma que deseamos intercambiar con el color del papel subyacente.

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

21	00	20	ED	4B	47	F2	3E
20	F5	CD	4A	00	E6	F0	B9
C2	26	DE	CD	4A	00	C6	0E
5F	CD	4A	00	CB	0F	CB	0F
CB	0F	CB	0F	E6	0F	CB	03
CB	03	CB	03	CB	03	B3	CD
4D	00	23	F1	3D	C2	FD	DD
C9	0C	M					

El siguiente programa en BASIC demostrará el funcionamiento de la rutina anterior. La pantalla se rellenará con diversos caracteres de colores diferentes. Luego, después de una breve pausa, los caracteres que están expuestos con pluma blanca tendrán los colores de la pluma y del papel intercambiados.

```
10 SCREEN 1
20 LET X=0
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=256 THEN LET X=0
70 NEXT A
80 LET X=17
90 FOR A=0 TO 31
100 VPOKE(8192+A),X
110 X=X+1
120 NEXT A
130 FOR X=1 TO 1000:NEXT X
140 POKE 62024,15
150 DEF USR=56750
160 A=USR(0)
```



Nombre: RUT13

Función: Deja en claro un recinto de pantalla.

Dirección de comienzo ..... 56900

Dirección de final ..... 56969

Suma total ..... 7829

**Requisitos:** La celdilla 62025 debe contener el número de columna correspondiente a la esquina superior izquierda del recuadro que va a aclararse.

La celdilla 62026 debe contener el número de fila correspondiente a la esquina superior izquierda del recinto que va a quedar en claro.

La celdilla 62027 debe contener la altura del recuadro concerniente.

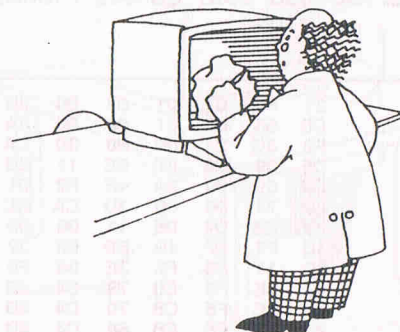
La celdilla 62028 debe contener la anchura del recuadro que va a aclararse.

**Limitaciones:** Se usa sólo en el modo 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	54	DE	01	28	00
3A	4A	F2	3D	CA	5F	DE	09
C3	57	DE	11	00	00	3A	49
F2	3D	5F	19	ED	5B	22	F9
19	3A	4B	F2	F5	3A	4C	F2
E5	F5	3E	20	CD	4D	00	23
F1	3D	C2	75	DE	E1	09	F1
3D	C2	70	DE	C9	0D	M	

El siguiente programa en BASIC demostrará el funcionamiento de la rutina que aclara un panel en la pantalla. La pantalla se rellenará con caracteres y luego, después de una breve pausa, se apelará a la rutina para aclarar un determinado recuadro de pantalla, que en este caso es el que corresponde a lo que se ha especificado en las líneas 100 a 115.

```
10 SCREEN 1
20 LET X=0
30 FOR A=6144 TO 6911
40 VPOKE(A),X
50 LET X=X+1
60 IF X=256 THEN LET X=0
70 NEXT A
100 POKE 62025,10
105 POKE 62026,15
110 POKE 62027,8
115 POKE 62028,5
120 DEF USR=56900
130 A=USR(0)
```





**Nombre: RUT14**

**Función:** Expone caracteres de un tamaño igual a ocho veces el tamaño normal.

**Dirección de comienzo** ..... 57000

**Dirección de final** ..... 57147

**Suma total** ..... 15560

**Requisitos:** La celdilla 62029 debe contener el número de columna donde quiere exponerse el carácter ampliado.

La celdilla 62030 debe contener el número de fila donde va a exponerse el carácter ampliado.

La celdilla 62031 debe contener el valor ASCII correspondiente al carácter que se desea exponer. (Véase el apéndice para la lista de los códigos ASCII de los caracteres).

**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

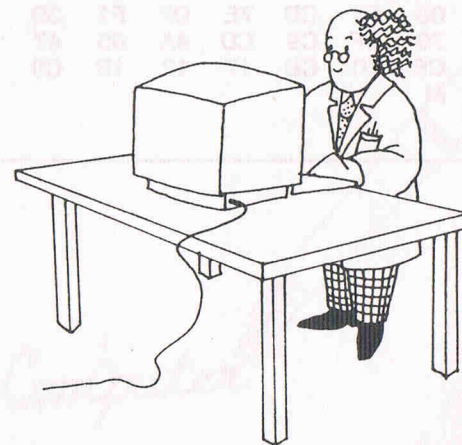
21	F0	07	01	08	00	3E	FF
CD	56	00	01	08	00	3A	4F
F2	3C	21	00	00	3D	CA	C5
DE	09	C3	BD	DE	11	5B	F2
CD	59	00	3A	4E	F2	01	20
00	21	00	00	3D	CA	DC	DE
09	C3	D4	DE	11	00	00	3A
4D	F2	5F	19	ED	5B	22	F9
19	11	5B	F2	3E	08	F5	1A
47	3E	FE	CB	78	C4	4D	00
23	3E	FE	CB	70	C4	4D	00
23	3E	FE	CB	68	C4	4D	00
23	3E	FE	CB	60	C4	4D	00
23	3E	FE	CB	58	C4	4D	00
23	3E	FE	CB	50	C4	4D	00
23	3E	FE	CB	48	C4	4D	00
23	3E	FE	CB	40	C4	4D	00
13	01	19	00	09	F1	3D	C2
EE	DE	C9	0E	M			

El siguiente programa en BASIC puede utilizarse para demostrar el funcionamiento de la rutina que expone caracteres con un tamaño ocho veces su tamaño normal.

```

5 SCREEN 1
10 DEF USR=57000
20 POKE 62030,5
30 POKE 62029,3
40 POKE 62031,77
50 A=USR(0)
60 POKE 62029,13
70 POKE 62031,83
80 A=USR(8)
90 POKE 62029,23
100 POKE 62031,88
110 A=USR(0)

```





Nombre: RUT15

Función: Efectúa una rotación de 90° sobre un carácter específico.

Dirección de comienzo ..... 57150

Dirección de final ..... 57222

Suma total ..... 8135

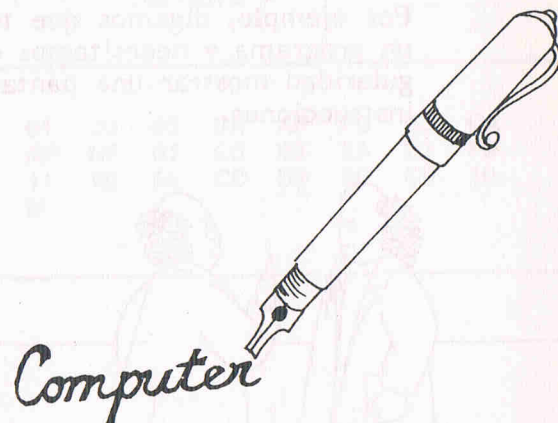
Requisitos: La celdilla 62032 debe contener el valor ASCII del carácter que se desea girar 90°.

Limitaciones: Se usa sólo en el modo 1 de pantalla.

01	08	00	21	00	00	3A	50
F2	3C	3D	CA	50	DF	09	C3
48	DF	E5	3E	08	F5	CD	67
DF	F1	23	3D	C2	53	DF	E1
13	EB	01	08	00	CD	5C	00
C9	11	62	F2	CD	4A	00	47
3E	08	F5	CD	7E	DF	F1	3D
C2	70	DF	C9	CD	4A	00	47
1A	CB	10	CB	17	12	1B	C9
0F	M						

El siguiente programa en BASIC puede usarse para demostrar el funcionamiento de la rutina anterior. El programa expondrá la palabra "COMPUTADORA" y luego girará 90° las letras C, P, U, y R.

```
10 SCREEN 1
20 DEF USR=57150
30 LOCATE 10,10
40 PRINT "COMPUTADORA"
50 POKE 62032,67
60 A=USR(0)
70 POKE 62032,80
90 A=USR(0)
100 POKE 62032,85
110 A=USR(0)
120 POKE 62032,82
130 A=USR(0)
```



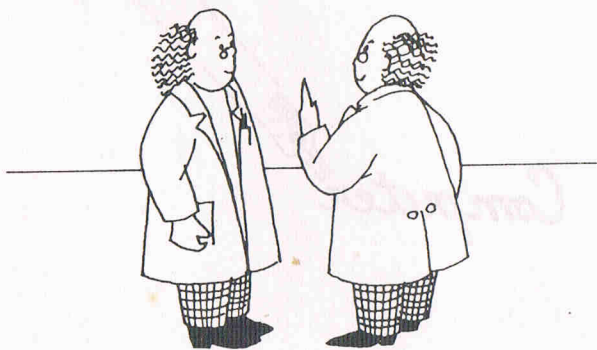
## Nombre: RUT16

### Función:

Esta rutina copiará lo expuesto en ese momento en pantalla sobre un área específica de memoria. Una vez que hayas efectuado esta copia serás capaz de hacer que se **guarde** en cinta y que vuelva a traerse de la cinta en una fecha posterior, usando la rutina 17. Cuando llames a esta rutina, la imagen presente en pantalla será copiada instantáneamente. Puedes luego guardarla en cinta usando el comando para que **guarde** en cinta un **bloque** de memoria dando las correspondientes direcciones; por ejemplo con el comando:

**BSAVE "CAS:IMAGEN",61000,61999**

Esta rutina también sigue siendo útil aunque no queramos guardar en cinta la imagen mostrada en pantalla. Por ejemplo, digamos que tenemos un programa y necesitamos con regularidad mostrar una pantalla con instrucciones.



En lugar de tener que exponer una a una las instrucciones -lo que sería relativamente lento y además consumiría mucha memoria-simplemente se apela a la rutina 17 una vez que hayas expuesto tus instrucciones de la manera normal y de haber aprovechado esta rutina 16 para guardar las instrucciones mostradas en pantalla en ese momento sobre un área específica de la memoria.

Posteriormente describiremos cómo usar la rutina 17 que es la contrapartida de esta rutina 16.

Dirección de comienzo .....	57250
Dirección de final .....	57273
Suma total .....	2509

**Limitaciones:** Se usa sólo en el modo 0 ó 1 de pantalla.

01	00	03	3A	AF	FC	3D	CA
AF	DF	01	C0	03	2A	22	F9
11	48	EE	CD	59	00	C9	10
M							



**Nombre: RUT17**

**Función:** Esta rutina transferirá a la pantalla lo contenido en el bloque de memoria donde se depositó la copia de la imagen mediante la rutina 16. Si hemos usado la rutina 16 para hacer que guarde el bloque de memoria correspondiente a lo expuesto en pantalla, podemos mandar que **traiga** de la cinta dicho bloque usando el comando: **BLOAD "CAS:IMAGEN"** habiendo copiado el bloque correspondiente a la imagen que habíamos vertido sobre la cinta con la rutina 16, podemos apelar a esta rutina 17 para que vuelva de nuevo a ser mostrada esa imagen en pantalla. Las rutinas 16 y 17 son pues extremadamente útiles cuando se utilizan combinadamente. Puedes por ejemplo, depositar en memoria una plana de instrucciones presentes en pantalla y mostrarla de nuevo en pantalla rápidamente sin más que apelar a esta rutina 17.

Dirección de comienzo ..... 57300  
Dirección de final ..... 57324  
Suma total ..... 2865

**Limitaciones:** Se usa sólo en el modo 0 y 1 de pantalla. Sé cuidadoso y no intentes mostrar la imagen contenida en el bloque cuando estes en el modo 0 de pantalla y la hayas copiado en el modo 1 o viceversa.

01	00	03	3A	AF	FC	3D	CA
E1	DF	01	CO	03	ED	5B	22
F9	21	48	EE	CD	5C	00	C9
11	M						

**Nombre: RUT18**

**Función:** Esta rutina es muy similar a la rutina 16, pero en lugar de copiar en un bloque de memoria toda la imagen mostrada en pantalla, únicamente copiará un determinado panel, un recuadro específico de la imagen.

Dirección de comienzo ..... 57350  
Dirección de final ..... 57424  
Suma total ..... 8130

**Requisitos:** La celdilla 62033 debe contener el número de columna correspondiente a la esquina superior izquierda del recuadro de imagen que va a copiarse.

La celdilla 62034 debe contener el número de fila correspondiente a la esquina superior izquierda de dicho recuadro de imagen que va a copiarse.

La celdilla 62035 debe contener la altura de dicho recuadro.

La celdilla 62036 debe contener la anchura de dicho recuadro.

**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	16	E0	01	28	00
3A	52	F2	3D	CA	21	E0	09
C3	19	E0	11	00	00	3A	51
F2	3D	5F	19	ED	5B	22	F9
19	11	48	EE	3A	53	F2	F5
C5	D5	E5	01	00	00	3A	54
F2	4F	CD	59	00	E1	D1	C1
09	EB	09	EB	F1	3D	C2	35
E0	C9	12	M				

**Nombre:** RUT19

**Función:** Esta rutina debe usarse en conjunción con la rutina 18. Esta rutina hace que vuelva a mostrarse el recuadro de imagen que fue previamente copiado sobre un bloque de memoria usando la rutina 18.

**Dirección de comienzo** ..... 57450

**Dirección de final** ..... 57525

**Suma total** ..... 8786

**Limitaciones:** Se usa sólo en el modo 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	7A	E0	01	28	00
3A	52	F2	3D	CA	85	E0	09
C3	7D	E0	11	00	00	3A	51
F2	3D	5F	19	ED	5B	22	F9
19	EB	21	48	EE	3A	53	F2
F5	C5	D5	E5	01	00	00	3A
54	F2	4F	CD	5C	00	E1	D1
C1	09	EB	09	EB	F1	3D	C2
9A	E0	C9	13	M			

El siguiente programa en BASIC puede usarse para demostrar la utilidad de las rutinas 18 y 19. El programa contiene un cierto número de instrucciones de **memorandum** que te permitirán saber exactamente qué es lo que está haciendo cada rutina.



```

5 SCREEN 1
10 REM UN MENSAJE SALE EN PANTALLA
20 LOCATE 10,9
30 PRINT "*****"
40 LOCATE 10,10
50 PRINT "*ESTA ZONA*"
60 LOCATE 10,11
70 PRINT "* SERA *"
80 LOCATE 10,12
90 PRINT "*RESERVADA*"
100 LOCATE 10,13
110 PRINT "*ENMEMORIA*"
120 LOCATE 10,14
130 PRINT "*****"
140 REM AHORA DEFINE LA ZONA A RESERVAR
150 POKE 62033,13
160 POKE 62034,10
170 POKE 62035,6
180 POKE 62036,11
190 REM AHORA DARAS UN MARGEN DE TIEMPO PARA LA LECTURA DEL
MENSAJE
200 FOR X=1 TO 1000:NEXT X
210 REM AHORA TIENES ACCESO A LA RUTINA DE SALVADO
220 DEF USR=57350
230 A=USR(0)
240 REM AHORA ESCRIBE EL MENSAJE
250 LET X=0
260 FOR A=6144 TO 6911
270 VPOKE A,X
280 LET X=X+1
290 IF X=256 THEN LET X=0
300 NEXT A
310 REM AHORA OTRA ACCION DE RETARDO
320 FOR X=1 TO 1000:NEXT X
330 CLS
340 REM AHORA TIENES ACCESO A LA RUTINA QUE RECUPERA
EL MENSAJE SALVADO.
350 DEF USR=57450
360 A=USR(0)

```

**Nombre:** RUT20

**Función:**

Tabla de puntuación. ¿Te has sentido alguna vez envidioso de las tablas de puntuación que aparecen en los programas de juegos hechos por compañías profesionales? Si así es, esta rutina será de gran provecho para ti. Debes apelar a esta subrutina al final del juego. La subrutina mostrará entonces los puntos alcanzados, y decidirá si esa puntuación es digna de ser insertada en la tabla de ganadores máximos. Luego permite al jugador que imponga por teclado su nombre antes de mostrar la nueva tabla de ganadores.

**Dirección de comienzo** ..... 57550  
**Dirección de final** ..... 58412  
**Suma total** ..... 95996

**Requisitos:**

A lo largo del juego debes usar la variable denominada SC para seguir el rastro de la puntuación (score). Obviamente, al comienzo del juego debes asignarle el valor 0. A lo largo del juego puedes tanto agregar como restar puntos al valor de la variable SC, dependiendo de cada juego en concreto. El valor de esta variable no debe sobrepasar nunca la gama de 0 a 65535. Al final del juego debes tener una instrucción BASIC para que vaya y venga (GOSUB..) a la rutina en BASIC que te hemos preparado.

Estas pocas líneas en BASIC son las que efectúan el traspaso del valor de la variable SC a la rutina en Código Máquina que presenta la tabla de puntuación.

```

10 X=INT(SC/10000)
20 POKE 57640,X
30 X=SC-(X*10000)
40 SC=INT(X/1000)
50 POKE 57641,SC
60 SC=X-(SC*1000)
70 X=INT(SC/100)
80 POKE 57642,X
90 X=SC-(X*100)
100 SC=INT(X/10)
110 POKE 57643,SC
120 SC=X-(SC*10)
130 POKE 57644,SC
140 CLS
150 DEF USR=57550
160 A=USR(0)
170 RETURN

```

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

C3	58	E1	53	4F	4E	59	20
20	20	20	20	20	2E	2E	54
4F	53	48	49	42	41	20	20
20	2E	2E	53	41	4E	59	4F
20	20	20	20	20	2E	2E	59
41	4D	41	48	41	20	20	20
20	2E	2E	47	4F	4C	44	53
54	41	52	20	20	2E	2E	30
30	30	30	39	30	30	30	30
38	30	30	30	30	37	30	30
30	30	36	30	30	30	30	35
00	00	00	00	00	00	09	00
48	49	47	48	20	53	43	4F
52	45	53	2E	59	4F	55	20
53	43	4F	52	45	44	20	31
32	33	34	35	2E	00	00	00
00	00	00	00	00	00	00	00
00	00	21	28	E1	7E	C6	30
77	23	7E	C6	30	77	23	7E
C6	30	77	23	7E	C6	30	77
23	7E	C6	30	77	21	28	E1
11	45	E1	01	05	00	ED	BC

01	10	00	11	A8	18	21	3A
E1	CD	5C	00	3E	14	11	29
E1	CD	5D	E3	22	26	E1	3A
28	E1	21	0D	E1	46	90	CA
A6	E1	FA	B9	E1	C3	56	E2
11	0E	E1	CD	5D	E3	ED	4B
26	E1	A7	ED	42	FA	56	E2
CA	63	E2	3A	28	E1	21	12
E1	46	90	CA	CA	E1	FA	DD
E1	C3	63	E2	11	13	E1	CD
5D	E3	ED	4B	26	E1	A7	ED
42	FA	63	E2	CA	70	E2	3A
28	E1	21	17	E1	46	90	CA
EE	E1	FA	01	E2	C3	70	E2
11	18	E1	CD	5D	E3	ED	4B
26	E1	A7	ED	42	FA	70	E2
CA	7D	E2	3A	28	E1	21	1C
E1	46	90	CA	12	E2	FA	25
E2	C3	7D	E2	11	1D	E1	CD
5D	E3	ED	4B	26	E1	A7	ED
42	FA	7D	E2	CA	8A	E2	3A
28	E1	21	21	E1	46	90	CA
36	E2	FA	46	E2	C3	8A	E2
11	22	E1	CD	5D	E3	ED	4B
26	E1	A7	ED	42	FA	8A	E2
3E	14	F5	3E	C8	CD	EB	E3
F1	3D	C2	48	E2	C3	8A	E3
01	14	00	CD	94	E2	11	0D
E1	E5	C3	C1	E2	01	0F	00
CD	94	E2	11	12	E1	E5	C3
C1	E2	01	0A	00	CD	94	E2
11	17	E1	E5	C3	C1	E2	01
05	00	CD	94	E2	11	1C	E1
E5	C3	C1	E2	11	21	E1	21
00	E1	E5	C3	C1	E2	C5	21
20	E1	11	25	E1	ED	B8	E1
29	7D	01	02	00	09	FE	0A
CA	B6	E2	09	FE	14	CA	B6
E2	09	FE	1E	CA	B6	E2	09
E5	C1	21	00	E1	11	0C	E1
ED	B8	C9	21	28	E1	01	05
00	ED	B0	01	2E	00	21	4B
E1	3E	0C	71	23	3D	C2	D1



E2	11	43	19	21	F8	E3	01
19	00	CD	5C	00	11	63	19
21	11	E4	01	1B	00	CD	5C
00	3E	0A	32	2D	E1	2E	0F
26	07	CD	C6	00	21	4B	E1
CD	9F	00	FE	0D	CA	4F	E3
F5	3A	AB	FC	3C	CA	1E	E3
F1	FE	2E	CA	1F	E3	FE	20
CA	1F	E3	D6	20	C3	1F	E3
F1	FE	41	FA	2C	E3	FE	5B
FA	40	E3	C3	31	E3	FE	2E
CA	3B	E3	3E	1C	CD	A2	00
3E	00	C3	40	E3	3E	5C	C3
40	E3	77	23	CD	A2	00	3A
2D	E1	3D	32	2D	E1	C2	FE
E2	E1	23	EB	01	0C	00	21
4B	E1	ED	B0	C3	8A	E3	21
00	00	1A	01	E8	03	CD	80
E3	13	1A	01	64	00	CD	80
E3	13	1A	01	0A	00	CD	80
E3	13	1A	01	01	00	CD	80
E3	C9	D6	2F	3D	C8	09	3D
C2	84	E3	C9	CD	A7	E3	21
2E	E1	01	0C	00	11	CA	1A
CD	5C	00	3E	32	F5	3E	C8
CD	EB	E3	F1	3D	C2	9B	E3
C9	3E	20	21	00	18	01	00
03	CD	56	00	01	0C	00	11
28	19	21	D1	E0	3E	05	F5
CD	5C	00	01	40	00	09	EB
01	0C	00	F1	3D	C2	BD	E3
01	05	00	11	34	19	21	0D
E1	3E	05	F5	CD	5C	00	01
40	00	09	EB	01	05	00	F1
3D	C2	D9	E3	C9	F5	3E	FF
3D	C2	EE	E3	F1	3D	C2	EB
E3	C9	45	4E	54	45	52	20
59	4F	55	52	20	4E	41	4D
45	20	55	53	49	4E	47	20
54	48	45	4B	45	59	42	4F
41	52	44	2E	20	54	48	45
4E	20	50	52	45	53	53	20
45	4E	54	45	52	2E	14	M

Nombre: RUT21

Función:

Ya te he mostrado dos rutinas que pueden aprovecharse para rotar un ángulo de 90° o bien un carácter individual, o bien todo el repertorio completo de caracteres. Las limitaciones principales de ambas rutinas es que no puedes mostrar a la vez en pantalla un carácter normal y la versión girada 90° de ese mismo carácter. Por ejemplo, si apelas a la subrutina para girar 90° la letra E, entonces cada vez que aparece dicha letra en la pantalla aparecerá girada 90°. Para algunas aplicaciones puede que quieras exponer un renglón de caracteres normales a lo ancho de la pantalla y luego tener algunas líneas con caracteres girados 90° en los laterales de la pantalla. Esta rutina te permitirá hacer exactamente ese tipo de acciones.

Si observas la lista de códigos ASCII del apéndice, verás que el repertorio de caracteres principales comienza con el código 48 -que representa el número 0- y sigue hasta el código 90 que representa la letra Z en mayúsculas. Esta rutina funciona generando otras tres versiones de este subgrupo principal del repertorio de caracteres, pero cada una de ellas girada 90, 180 y 270° respectivamente, con relación a la versión normal.

Con eso obtienes cuatro subgrupos de caracteres: uno normal, uno boca abajo, uno rotado 90° a la izquierda y otro a la derecha 90°. Una desventaja de esta rutina es que serás incapaz de usar las letras minúsculas además de algunos otros caracteres; sin embargo, no considero que eso pueda suponer un problema para la mayoría de las aplicaciones.

Dirección de comienzo ..... 59200  
 Dirección de final ..... 59353  
 Suma total ..... 16826

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

21	80	01	11	48	EE	01	58
01	CD	59	00	21	48	EE	11
D8	02	01	58	01	CD	5C	00
21	48	EE	11	30	04	01	58
01	CD	5C	00	21	48	EE	11
88	05	01	58	01	CD	5C	00
21	D8	02	CD	95	E7	21	30
04	CD	95	E7	21	30	04	CD
95	E7	21	88	05	CD	95	E7
21	88	05	CD	95	E7	21	88
05	CD	95	E7	C9	3E	2B	F5
E5	3E	08	F5	CD	BA	E7	F1
23	3D	C2	9B	E7	E1	E5	13
EB	01	08	00	CD	5C	00	E1
01	08	00	09	F1	3D	C2	97
E7	C9	11	62	F2	CD	4A	00
47	3E	08	F5	CD	D1	E7	F1
3D	C2	C3	E7	C9	CD	4A	00
47	1A	CB	10	CB	17	12	1B
C9	15	M					

**Nombre:** RUT22

**Función:** Esta rutina te dará un repertorio de caracteres alternativo al repertorio que está normalmente prefijado en el ordenador y es el utilizado cuando se enciende. La rutina sólo necesita ser citada una vez al comienzo de tu programa y con ella redefinirás los tipos, las formas de los caracteres cuyos códigos ASCII están comprendidos entre el 48 y el 90.

Dirección de comienzo ..... 59400  
 Dirección de final ..... 59757  
 Suma total ..... 46374

**Limitaciones:** Se usa sólo en el modo 1 de pantalla. No olvides colocar el teclado en el modo de "enclavamiento de mayúsculas" (CAPS LOCK) a la hora de teclear el nuevo repertorio de caracteres.

21	15	E8	01	58	01	11	80
01	CD	5C	00	C9	38	4C	CE
D6	E6	64	38	00	78	18	18
18	18	18	7E	00	7C	C6	66
1C	30	62	FE	00	7E	8C	18
3C	06	86	7C	00	1C	3C	6C
CC	FE	0C	0C	00	7E	C0	C0
FC	06	C6	7C	00	1C	30	60
FC	C6	C6	7C	00	FE	86	0C
18	30	60	C0	00	7C	C6	C6
7C	C6	C6	7C	00	7C	C6	C6
7E	06	8C	78	00	38	44	92
A2	A2	92	44	38	00	DF	6F
37	39	0F	03	01	70	60	18
38	70	60	18	38	FF	FF	FF
FF	FF	F9	E7	83	FF	FF	FD
F3	E7	CF	8F	04	FF	FF	FF
FF	FF	FE	ED	FF	FF	FF	FF
FF	FF	FF	FF	FF	3A	46	C6



C6	C6	CE	76	00	DC	E6	C6
FC	C6	C6	FC	00	3C	42	C0
C0	C0	66	3C	00	0E	76	CE
C6	C6	46	3C	00	3C	66	C6
FC	C0	66	3C	00	5C	66	60
F8	60	60	60	80	3A	66	C6
C6	4E	76	86	7C	C0	FC	E6
C6	C6	CC	DE	00	7C	B8	30
30	30	34	FC	00	7F	06	06
46	C6	6C	38	00	C6	CC	D8
F0	F0	D8	CE	00	F8	30	60
60	60	66	7C	00	AC	D6	D6
D6	D6	D6	D4	00	DC	F6	E6
C6	C6	CC	DE	00	38	4C	C6
C6	C6	64	38	00	DC	EE	C6
C6	C6	EC	D8	C0	38	4C	C6
C6	C6	64	38	1E	DC	E6	C6
C4	F8	CC	C6	00	7C	C2	C0
7C	06	C6	7C	00	FE	18	30
30	30	30	18	00	F6	66	46
C6	C6	EE	76	00	E6	46	C6
C6	66	3C	18	00	CC	46	D6
D6	D6	D6	6C	00	86	CC	78
38	3C	66	C2	00	C6	66	66
66	3C	8C	CC	78	3E	46	0C
18	30	62	FC	00	16	M	



Nombre: RUT23

Función: Esta rutina te proporcionará otro repertorio alternativo de caracteres para usarlo en el modo 1 de pantalla.

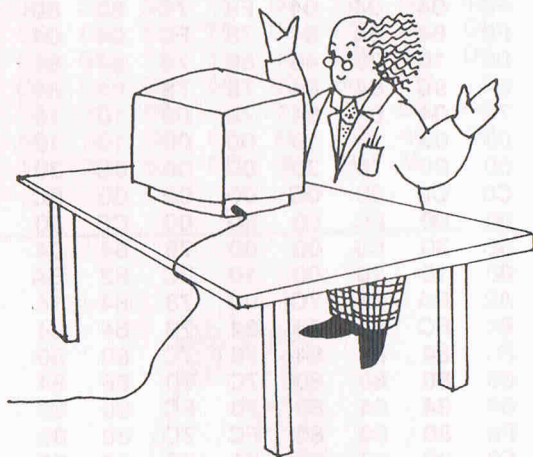
Dirección de comienzo ..... 59800

Dirección de final ..... 60157

Suma total ..... 1188

21	A5	E9	01	58	01	11	80
01	CD	5C	00	C9	78	84	84
84	84	84	84	78	20	60	A0
20	20	20	20	70	78	84	04
08	10	20	40	FC	78	84	04
1C	1C	04	84	78	0C	14	24
44	84	FC	04	04	7C	80	80
F8	04	04	04	F8	7C	80	80
F8	84	84	84	78	FC	04	04
08	10	20	40	80	78	84	84
E4	9C	84	84	78	78	84	84
7C	04	04	84	78	00	10	10
00	00	10	10	00	00	10	10
00	00	10	30	00	00	0C	30
C0	C0	30	0C	00	00	00	FC
00	00	FC	00	00	00	C0	30
0C	30	C0	00	00	78	84	04
08	10	10	00	10	7C	82	BA
A2	BA	82	7C	00	78	84	84
84	FC	84	84	84	78	84	84
F8	84	84	84	78	7C	80	80
80	80	80	80	7C	70	88	84
84	84	84	88	70	FC	80	80
F8	80	80	80	FC	7C	80	80
F0	80	80	80	80	7C	80	80
80	9C	84	84	78	84	84	84
FC	FC	84	84	84	FC	30	30
30	30	30	30	FC	FC	10	10
10	10	90	90	60	88	90	A0
C0	C0	A0	90	88	80	80	80
80	80	80	80	F8	84	CC	B4
84	84	84	84	84	84	C4	A4

A4	94	94	8C	84	78	84	84
84	84	84	84	78	78	84	84
F8	80	80	80	80	78	84	84
84	A4	94	8C	78	78	84	84
F8	A0	90	88	84	78	84	80
60	18	04	84	78	FC	30	30
30	30	30	30	30	84	84	84
84	84	84	84	78	84	84	48
48	48	30	30	30	84	84	84
84	84	B4	B4	48	84	48	48
30	30	48	48	84	84	84	48
48	30	30	30	30	FC	04	08
10	20	40	80	FC	17	M	



Nombre: RUT24

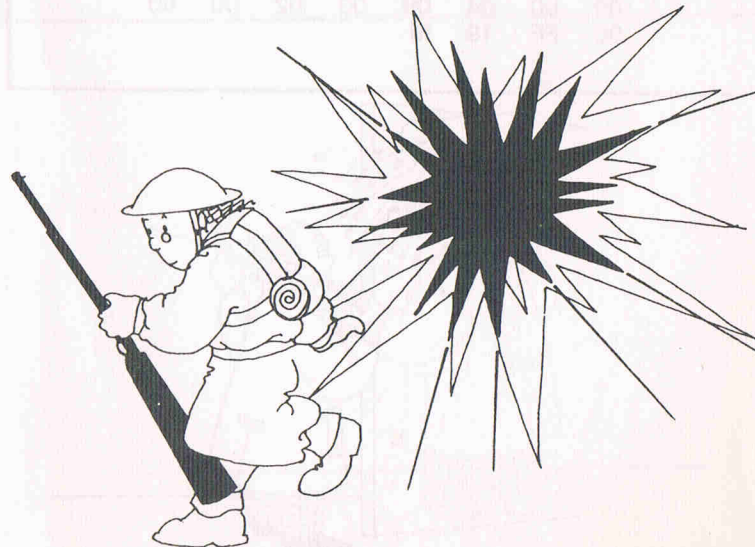
**Función:** Cuando se apela a esta subrutina, se logra el efecto sonoro de una explosión.

Dirección de comienzo ..... 60200

Dirección de final ..... 60301

Suma total ..... 9080

3E	07	1E	F5	CD	93	00	21
71	EB	22	6A	EB	01	D0	07
0B	79	B0	C2	38	EB	3A	6E
EB	D6	01	38	06	32	6E	EB
C3	35	EB	2A	6A	EB	7E	FE
FF	C8	32	6E	EB	23	5E	3E
06	CD	93	00	23	5E	3E	08
CD	93	00	23	22	6A	EB	C3
35	EB	70	EB	70	EB	00	00
FF	09	14	13	05	12	0B	0A
0F	0C	02	02	0D	08	09	0E
0C	06	0C	07	04	09	03	02
05	00	00	00	FF	18	M	





Nombre: RUT25

**Función:** Cuando se cita esta subrutina conseguimos el efecto sonoro de un rociador de agua.

Dirección de comienzo ..... 60350  
Dirección de final ..... 60472  
Suma total ..... 8588

3E	07	1E	F5	CD	93	00	21
07	EC	22	00	EC	01	D0	07
0B	79	B0	C2	CE	EB	3A	04
EC	D6	01	38	06	32	04	EC
C3	CB	EB	2A	00	EC	7E	FE
FF	C8	32	04	EC	23	5E	3E
06	CD	93	00	23	5E	3E	08
CD	93	00	23	22	00	EC	C3
CB	EB	37	EC	06	EC	00	00
FF	00	05	02	00	05	04	00
05	06	01	05	08	01	04	0A
00	04	0C	00	04	0E	02	03
0F	05	03	0E	06	02	0C	08
02	0A	08	01	08	09	01	06
09	00	04	09	00	02	00	00
00	FF	19	M				

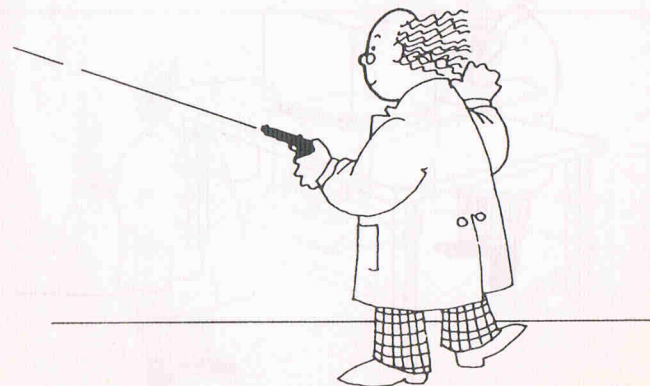


Nombre: RUT26

**Función:** Cuando se pone en marcha esta subrutina produce el efecto sonoro del disparo de una pistola.

Dirección de comienzo ..... 60500  
Dirección de final ..... 60601  
Suma total ..... 9581

3E	07	1E	F5	CD	93	00	21
9D	EC	22	96	EC	01	D0	07
0B	79	B0	C2	64	EC	3A	9A
EC	D6	01	38	06	32	9A	EC
C3	61	EC	2A	96	EC	7E	FE
FF	C8	32	9A	EC	23	5E	3E
06	CD	93	00	23	5E	3E	08
CD	93	00	23	22	96	EC	C3
61	EC	B8	EC	9C	EC	00	00
FF	01	14	0A	01	12	0B	01
0F	0C	02	0C	0D	02	09	0E
02	06	0C	02	04	09	03	02
05	00	00	00	FF	1A	M	



Nombre: RUT27

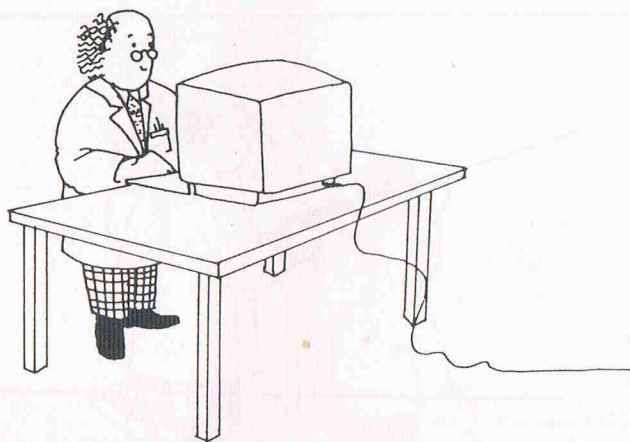
**Función:** Esta rutina reproduce el efecto sonoro tantas veces encontrado en las películas del tipo "La Guerra de las Galaxias".

Dirección de comienzo ..... 60650

Dirección de final ..... 60725

Suma total ..... 7620

3E	07	1E	F5	CD	93	00	3E
08	1E	0F	CD	93	00	3E	07
1E	FE	CD	93	00	3E	00	1E
6E	CD	93	00	3E	01	1E	00
CD	93	00	1E	6E	3E	00	CD
93	00	3E	C8	F5	3E	0A	3D
C2	19	ED	F1	3D	C2	16	ED
7B	C6	06	CA	2D	ED	3D	5F
C3	0F	ED	3E	07	1E	FF	CD
93	00	C9	1B	M			



Nombre: RUT28

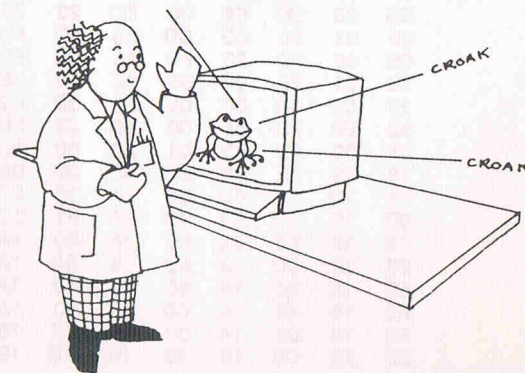
**Función:** Esta es otra rutina para efectos sonoros que sólo puedo describir como el tipo de sonido que una rana mecánica emitiría. Ensayala y verás lo que quiere decir.

Dirección de comienzo ..... 60750

Dirección de final ..... 60872

Suma total ..... 11141

3E	07	1E	F5	CD	93	00	21
9E	ED	22	99	ED	01	D0	07
0B	79	B0	C2	5E	ED	3A	9C
ED	D6	01	38	06	32	9C	ED
C3	5B	ED	2A	99	ED	7E	FE
FF	C8	32	9C	ED	23	3E	02
5E	CD	93	00	23	3E	03	5E
CD	93	00	23	3E	09	5E	CD
93	00	23	22	99	ED	C3	5B
ED	9D	ED	BE	ED	00	00	FF
01	00	04	0F	01	96	03	0F
01	32	03	0F	00	C8	02	0F
00	64	02	0F	00	C8	01	0F
00	32	01	0F	00	00	00	00
FF	00	00	0C	0F	00	00	00
00	FF	1C	M				





Nombre: RUT29

**Función:** Esta rutina consigue el efecto visual de que la imagen en pantalla desaparezca por un agujero situado en el centro de la pantalla.

Dirección de comienzo ..... 58900

Dirección de final ..... 59128

Suma total ..... 16986

**Requisitos:** Si deseas volver a ver lo que estaba mostrado en pantalla antes de apelar a esta subrutina, debes primero utilizar la rutina 16, que como recordarás deposita una copia de la imagen en un bloque de memoria. Luego apelarás a la rutina 17 -para devolver la imagen a partir del bloque, cuando desees volver a mostrar lo originalmente expuesto en pantalla.

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

DD	21	88	E6	FD	21	B8	E6
3E	0C	F5	DD	6E	00	DD	23
DD	66	00	DD	23	3E	20	01
20	00	CD	56	00	DD	6E	00
DD	23	DD	66	00	DD	23	3E
20	01	20	00	CD	56	00	FD
6E	00	FD	23	FD	66	00	FD
23	01	20	00	3E	18	F5	3E
20	CD	4D	00	09	F1	3D	C2
52	E6	FD	6E	00	FD	23	FD
66	00	FD	23	01	20	00	3E
18	F5	3E	20	CD	4D	00	09
F1	3D	C2	6D	E6	01	10	27
0B	79	B0	C2	7C	E6	F1	3D
C2	1E	E6	C9	00	18	E0	1A
20	18	C0	1A	40	18	A0	1A
60	18	80	1A	80	18	60	1A
A0	18	40	1A	C0	18	20	1A
E0	18	00	1A	00	19	E0	19
20	19	C0	19	40	19	A0	19

60	19	80	19	00	18	1F	18
01	18	1E	18	02	18	1D	18
03	18	1C	18	04	18	1B	18
05	18	1A	18	06	18	19	18
07	18	18	18	08	18	17	18
09	18	16	18	0A	18	15	18
0B	18	14	18	0C	18	13	18
0D	18	12	18	0E	18	11	18
0F	18	10	18	1D	M		

60	19	80	19	00	18	1F	18
01	18	1E	18	02	18	1D	18
03	18	1C	18	04	18	1B	18
05	18	1A	18	06	18	19	18
07	18	18	18	08	18	17	18
09	18	16	18	0A	18	15	18
0B	18	14	18	0C	18	13	18
0D	18	12	18	0E	18	11	18
0F	18	10	18	1D	M		

Nombre: RUT30

**Función:** Esta rutina es exactamente la opuesta a la rutina 29: consigue el efecto de una imagen que surge y se agranda de un agujero situado en el centro de la pantalla.

Dirección de comienzo ..... 58700

Dirección de final ..... 58869

Suma total ..... 15234

**Requisitos:** Con el fin de utilizar esta subrutina debes primeramente haber utilizado la rutina 16 para depositar en un bloque de memoria una copia de la imagen. Esta rutina construirá luego la imagen de manera tal que parece estar surgiendo de un hueco situado en el centro de la pantalla.

**Limitaciones:** Se usa sólo en el modo 1 de pantalla.

3E	0C	11	C1	E5	F5	1A	32
F0	E0	13	1A	00	5A	60	93
36	E5	03	1A	40	BF	F0	80
80	00	00	80	80	3A	AF	FC
3D	CA	77	E5	01	28	00	3A
F2	E5	3D	CA	82	E5	09	03
DC	F9	84	80	00	3A	F1	E5
3D	5F	19	ED	5B	22	F9	19
EB	21	48	F4	9D	F3	C5	F5
C0	D4	E5	01	00	00	3A	F4
E5	4F	CD	5C	00	E1	D1	C1
09	EB	09	EB	F1	3D	C2	97
E5	01	10	27	0B	79	B0	C2
B4	E5	D1	F1	3D	C2	51	E1
81	08	0C	02	0A	0B	0B	04
0C	0A	0A	06	0E	09	09	08
10	08	08	0A	12	07	07	0C
14	06	06	0E	16	05	05	10
18	04	04	12	1A	03	03	14
1C	02	02	16	1E	01	01	18
20	01	01	18	20	1E	M	

Nombre: RUT31

**Función:** Esta rutina escribirá al revés lo que aparece expuesto en un recinto específico de la pantalla. Por lo tanto, si en dicho recinto estuviera expuesta la palabra "COMPUTADORA" después de apelar a esta subrutina, dicha palabra aparecería en la pantalla como "ARODATUPMOC".

Dirección de comienzo ..... 58550

Dirección de final ..... 58657

Suma total ..... 14068

**Requisitos:** La celdilla 62039 debe contener el número de columna correspondiente a la esquina superior izquierda del recuadro donde va a invertirse la forma de escribir. La celdilla 62040 debe contener el número de fila correspondiente a dicha esquina superior izquierda. La celdilla 62041 debe contener la altura, y la celdilla 62042 la anchura, del recuadro de pantalla donde se va a invertir la forma de escribir.

**Limitaciones:** Se usa sólo en los modos 0 y 1 de pantalla.

21	00	00	01	20	00	3A	AF
FC	3D	CA	C6	E4	01	28	00
3A	58	F2	3D	CA	D1	E4	09
C3	C9	E4	11	00	00	3A	57
F2	3D	5F	19	ED	5B	22	F9
19	C5	01	00	00	3A	5A	F2
3D	4F	E5	D1	09	C1	3A	59
F2	F5	D5	E5	3A	5A	F2	CB
0F	CB	BF	F5	CD	4A	00	F5
EB	CD	4A	00	EB	CD	4D	00
EB	F1	CD	4D	00	EB	13	2B
F1	3D	C2	F9	E4	E1	D1	09
E5	EB	09	EB	E1	F1	3D	C2
EF	E4	C9	1F	M			



Supongamos ahora que ya has ingresado por teclado y comprobado todas las rutinas. Ahora, queda claro que es bastante inconveniente tener agrupadas y guardadas en cassette más de treinta rutinas en un sólo bloque, por no mencionar el problema de encontrar dónde comienza cada una de las subrutinas. Te presento pues un truco que te hará la vida mucho más fácil. Primeramente, manda a la computadora que anule los datos anteriores, reserve un espacio de 200 octetos para almacenar los datos literales, y establezca la cima de la memoria utilizable por el BASIC en la dirección 56000. Es decir, teclea:

```
CLEAR 200,56000
```

Ahora ya puedes mandarle que traiga del cassette una a una las rutinas, utilizando el comando para que **cargue un bloque** de memoria, en la forma:

```
BLOAD "CAS:RUT1"  
BLOAD "CAS:RUT2"
```

y así sucesivamente hasta:

```
"  
"  
BLOAD "CAS:RUT31"
```

Ahora ya puedes mandarle que guarde en cinta todas las rutinas como un único bloque, usando la instrucción:

```
BSAVE "CAS:RUTODO",56000,60872
```

En este momento, en lugar de tener que buscar una rutina individual, ya puedes mandar que traiga del cassette todas las rutinas y las cargue en el bloque correspondiente usando el siguiente comando:

```
BLOAD "CAS:RUTODO"
```

La idea de tener todas las rutinas en memoria también te impide que quieras apelar a una que no haya sido previamente traída del cassette y cargada en memoria.

Ya te he mostrado cómo se apela, cómo se cita, cómo se pone en marcha, la rutina deseada aprovechando la definición de rutina de usuario para dar la dirección de comienzo, con la clave "DEF USR" y luego -en el momento oportuno- apelando a ella con una instrucción de asignación  $A = \text{USR}(0)$ . Si estás confeccionando un programa largo -con frecuentes apelaciones a muchas de las rutinas- puede que el repetido uso de las dos instrucciones anteriores ocupe un montón de memoria. En tal situación, deberías adoptar el siguiente método de programación:

El método involucra agregar diversas líneas en BASIC al final de tu programa. Digamos que de manera regular deseas apelar a tres de las rutinas, por lo que necesitarás añadir las siguientes instrucciones:

```
10000 DEF USR=56000  
10005 A=USR(0)  
10010 RETURN  
10015 DEF USR=56600  
10020 A=USR(0)  
10025 RETURN  
10030 DEF USR=57000  
10035 A=USR(0)  
10040 RETURN
```

Si quieres apelar a cualquiera de estas tres subrutinas en Código Máquina, puedes hacerlo directamente mandando que **vaya y venga** (GOSUB...) al número de línea correspondiente. Así la instrucción:

```
GOSUB 10015
```

hará que vaya y ejecute la segunda de las rutinas en Código Máquina y luego vuelva para continuar con lo que estaba haciendo.

1. The first part of the report is a general introduction to the project. It describes the objectives of the study and the scope of the work. It also provides a brief overview of the methodology used in the research.

2. The second part of the report is a detailed description of the methodology used in the study. This includes a discussion of the data sources, the data collection methods, and the data analysis techniques used.

3. The third part of the report is a presentation of the results of the study. This includes a discussion of the findings of the research and a comparison of these findings with the results of previous studies.

4. The fourth part of the report is a discussion of the implications of the findings of the study. This includes a discussion of the practical implications of the findings and a discussion of the theoretical implications of the findings.

5. The fifth part of the report is a conclusion. This summarizes the main findings of the study and provides a final assessment of the value of the research.

1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

The following table shows the results of the study for the years 1988 to 2025. The table is organized into columns representing the years and rows representing the different variables measured. The data shows a general upward trend in most variables over the period, with some fluctuations in the later years.

1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

## CONTENTS

- 1) Table de Contenu
- 2) Liste des Pages
- 3) Liste des Figures
- 4) Liste des Références
- 5) Liste des Annexes
- 6) Liste des Bibliographies
- 7) Liste des Résumés
- 8) Liste des Mots-clés
- 9) Liste des Auteurs
- 10) Liste des Éditeurs
- 11) Liste des Distributeurs
- 12) Liste des Revendeurs
- 13) Liste des Abonnés
- 14) Liste des Membres
- 15) Liste des Partenaires
- 16) Liste des Sponsors
- 17) Liste des Collaborateurs
- 18) Liste des Consultants
- 19) Liste des Experts
- 20) Liste des Témoins
- 21) Liste des Jurés
- 22) Liste des Arbitres
- 23) Liste des Médiateurs
- 24) Liste des Conciliateurs
- 25) Liste des Médiateurs
- 26) Liste des Conciliateurs
- 27) Liste des Médiateurs
- 28) Liste des Conciliateurs
- 29) Liste des Médiateurs
- 30) Liste des Conciliateurs

## Apéndices



---

# Apéndices

---

- 1) Tabla de Colores Pluma/Papel
- 2) Modo Cero de Pantalla
- 3) Modo Uno de Pantalla
- 4) Código ASCII
- 5) Nota del Autor

**TABLA DE COLORES PLUMA/PAPEL**

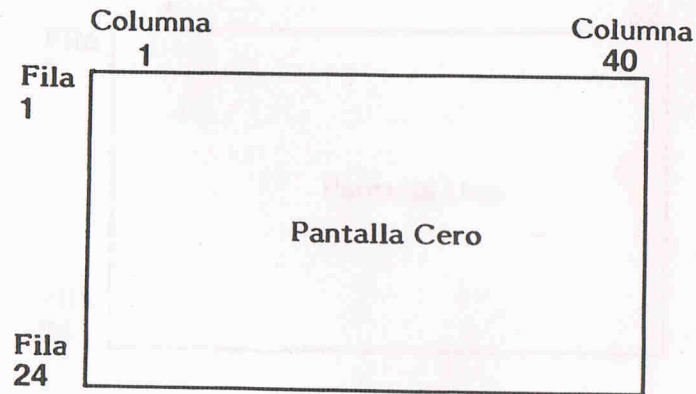
Colores	Código de Papel	Código de Pluma
NEGRO	1	16
VERDE MEDIO	2	32
VERDE PALIDO	3	48
AZUL OSCURO	4	64
AZUL PALIDO	5	80
ROJO OSCURO	6	96
CIANO	7	112
ROJO MEDIO	8	128
ROJO PALIDO	9	144
AMARILLO OSCURO	10	160
AMARILLO PALIDO	11	176
VERDE OSCURO	12	192
MAGENTA	13	208
GRIS	14	224
BLANCO	15	240



Código de Pluma	Código de Papel	Color
10	1	NEGRO
25	2	VERDE MEDIO
40	3	VERDE PÁLIDO
45	4	AZUL OSCURO
50	5	AZUL PÁLIDO
60	6	ROJO OSCURO
110	7	GRANDE
120	8	ROJO ULTRIO
140	9	ROJO PÁLIDO
160	10	AMARILLO OSCURO
170	11	AMARILLO PÁLIDO
190	12	VERDE OSCURO
200	13	MAGENTA
250	14	GRIS
260	15	BLANCO

## MODO CERO DE PANTALLA

El siguiente diagrama muestra la numeración de filas y columnas en el modo cero de pantalla:

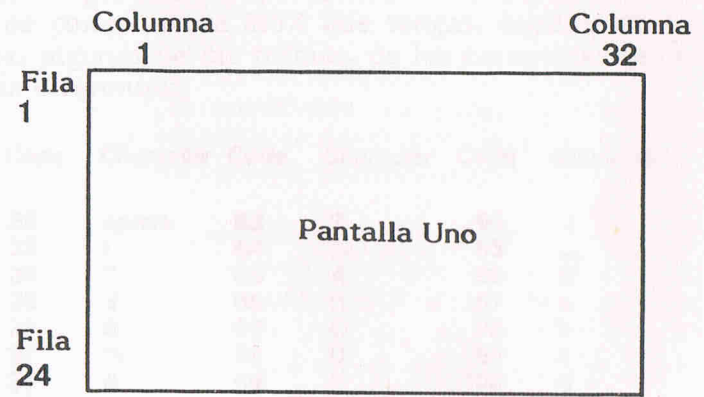


El siguiente diagrama muestra la numeración de las filas y columnas en el modo cero de pantalla.



MODO UNO DE PANTALLA

El siguiente diagrama muestra la numeración de las filas y columnas en el modo 1 de pantalla:





# ASCII

La siguiente tabla muestra los códigos ASCII que corresponden a los 256 caracteres que pueden exponerse en pantalla. Dependiendo de la marca particular de computadora MSX que tengas, algunos de los tipos, algunas de las formas, de los caracteres reales serán diferentes.

Code	Character	Code	Character	Code	Character
32	space	63	?	94	↑
33	!	64	@	95	—
34	"	65	A	96	£
35	#	66	B	97	a
36	\$	67	C	98	b
37	%	68	D	99	c
38	&	69	E	100	d
39	'	70	F	101	e
40	(	71	G	102	f
41	)	72	H	103	g
42	*	73	I	104	h
43	+	74	J	105	i
44	,	75	K	106	j
45	-	76	L	107	k
46	.	77	M	108	l
47	/	78	N	109	m
48	0	79	O	110	n
49	1	80	P	111	o
50	2	81	Q	112	p
51	3	82	R	113	q
52	4	83	S	114	r
53	5	84	T	115	s
54	6	85	U	116	t
55	7	86	V	117	u
56	8	87	W	118	v
57	9	88	X	119	w
58	:	89	Y	120	x
59	;	90	Z	121	y
60	<	91	[	122	z
61	=	92	/		
62	>	93	]		

## NOTA DEL AUTOR

Espero que las rutinas de este libro demuestren que te han sido de alguna utilidad en tus programas.

Si tienes algunas cuestiones relacionadas específicamente con las rutinas de este libro, o algunas cuestiones sobre la programación en MSX BASIC en general, no dudes en escribirme a través de los editores de este libro (pero no olvides incluir un sobre con el sello correspondiente para mi contestación).

También agradecería tus comentarios sobre el tema general de este libro. Por ejemplo ¿te gustaría ver el volumen 2 de "MSX 30 Rutinas de Utilidad en Código Máquina", y si así es, ¿qué clase de subrutinas te gustaría ver incluidas en tal libro?

Buena suerte con tus programas.

Steve Webb  
Londres  
Marzo 1985