Precio exterior us$ 3,00

programación popular

N 8 $ 180

REVISTA MENSUAL DE COMPUTACION

PROGRAMAS PARA:

COMMODORE 64
CZ 1000/1500
SPECTRUM
TI 99/4A
TK 83/85
TK 90X
TS 2068

MEGA - BASIC

Ampliamos su máquina



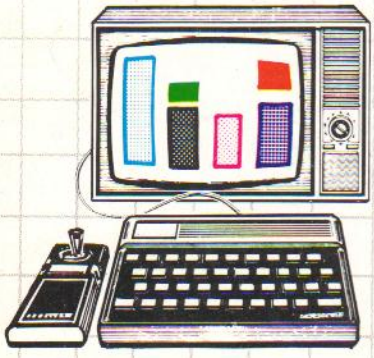
MICROCOMPUTADOR MICRODIGITAL TK-90X

Color y sonido a través del T.V.

16K y 48 K



EL MICROCOMPUTADOR CON MILES DE PROGRAMAS



GARANTIA 6 MESES

En venta en comercios de microcomputadores, artículos del hogar, electrónica, fotografía y librerías.

SOFTWARE Y PERIFERICOS TOTALMENTE COMPATIBLES CON ZX SPECTRUM +" ®

- Control del volumen del sonido a través del TV (sintetizador operado por BASIC)
- Interface incorporado para joystick
- Mensajes de ejecución y código de reportes de errores en castellano.
- TRACE: Comando de seguimiento de programas, permitiendo la rápida corrección de errores de lenguaje.
- UDG: Comando de editor de caracteres especiales definidos por el usuario (acentos, Ñ, etc.).
- Feedback sonoro del teclado
- Fuente de alimentación con interruptor.
- Ameno, fácil y completo manual de instrucciones en castellano.

MICRODIGITAL

Importa, distribuye y garantiza:
ARVOC s.a.i.c.f.i.
Tte. Gral. Juan D. Perón 1563 (ex Cangallo) C.P. 1037 - Capital. Tel. 35-2400 / 2511 / 8241.

ZX SPECTRUM + " ALL RIGHTS RESERVED SINCLAIR RESEARCH LTD

Ref 1

Dirección General
Dr. Héctor J. Silva
Dirección Financiera
Enrique H. Ramírez
Dirección Comercial
Ricardo J. Baiele

Colaboran en este número:
Abbot Soft (Marcelo)
Juan Carlos Bruni
Norberto H. Domizzi
Eduardo A. Grosclaude
Raúl D. Lambertucci
Gerardo Javier Mateos
Rodolfo Miranda
Héctor R. Molena
Gustavo C. Petrasso
Mario V. Simes
Ing. Raúl D. Vera
Iván F. Sarachman
Gabriel Schnitman
Arte y Diagramación:
Daniel P. Unamuno
Francisco J. Fenili

programación popular

Publicidad (Sr. Baiele)
393-8456
Suscripciones Telefónicas
(Srta. Adriana)
393-8560
Administración (Srta. Alicia):
Maipú 464, Piso 8, Of 801
(1006) Capital Federal
393-8560

Kioscos Capital y Gran Bs. As.
Prícolo S.A.
Kioscos Interior del País
Ryela SAIC y F
Impresión
Imp. Gráficas Tabare SAIC
Fotocromía
Gráfica Boedo

Número 8
Los ejemplares atrasados se venden al precio de tapa del último ejemplar

Programación Popular es una publicación de **Ediciones del Futuro S.A.** con domicilio en la calle Maipú 464, piso 8, oficina 801, (1006) Capital Federal - T.E. 393-8456 y 393-8560. Registro Nacional de la Propiedad Intelectual 311.847. Marca Registrada 1.477.565. Editor responsable: **Héctor J. Silva**. Queda hecho el depósito que marca la Ley 11.723, todos los derechos reservados. Prohibida la reproducción total o parcial del material publicado. Todas las menciones de marcas, modelos, etc., excepción hecha de las incluidas en los avisos comerciales, se realizan con fines de servicios, informativos y técnicos, no siendo en ningún caso pagadas por persona o empresa alguna. Al ser informativa esta publicación no se responsabiliza por cualquier problema que pueda plantear la fabricación, el funcionamiento y/o la aplicación de los sistemas y los dispositivos que en ella se describen. Los artículos no reflejan necesariamente la opinión de la dirección de la publicación siendo la responsabilidad de los mismos exclusivamente de sus autores.

EDITORIAL

Se acerca el comienzo de un nuevo año: 1986. Es época de festejos, de reencuentros, de nostalgias y de "balances".

Y de esto último se trata, de detenernos por un momento a recordar las cosas más importantes que sucedieron en 1985 y formularnos propósitos y deseos para el año entrante.

En este año que se va sucedieron cosas muy importantes en nuestro querido país: reafirmamos el ejercicio de la democracia, se restableció el valor de nuestra moneda y comenzó a nacer así la confianza en nuestro país en el exterior y, es bueno decirlo, también en muchos argentinos. En lo que a nuestra actividad se refiere, se clarificó el panorama de desarrollo del área informática y se realizaron las preadjudicaciones de la famosa Resolución 44, aunque aún queda mucho por hacer en este terreno.

*También hubo otro acontecimiento fundamental en esta materia y supongo que sabrán a qué me refiero: sí, efectivamente !!! ... nació la primera y también, a juzgar por lo que dicen nuestros lectores, la mejor revista argentina de microcomputación: **Programación Popular**. En lo que a nosotros se refiere, nos hacemos el firme propósito de seguir creciendo en servicios a ustedes, lectores, nuestros únicos destinatarios, tanto en cantidad de páginas como en calidad de software y hardware. Amén.*

*Y ahora me pongo serio, sólo para formular el no por conocido menos sentido augurio de **MUCHAS FELICIDADES EN 1986**, amigos.*

Hasta el número 9...

Dr. Héctor J. Silva

NOVEDADES

Noticias, anticipos y comentarios
de nuestro mundo de la microcomputación 50

PROGRAMACION

Curso Basic (5)
En este número: **Diagramación IF/THEN**.....12
Lenguaje de Máquina (3)
Veremos: Codificación hexadecimal Desensamble 46
Programas en **"High Speed"**
Comenzamos a develar el secreto de la carga y lectura de Programas en "alta velocidad" en la **TK85**.....48

PROGRAMAS

C64	1000	1500	Spectrum	TI	TK 83	TK 85	TK 90X	TS 2068	
●			●	●			●	●	Aritmética espacial.....36
									Ataque Espacial 29
									Calendario 6
X									Colores al azar.....42
	●	●			●	●			Costos de Fabricación (I).....30
			●				●	●	Cubo mágico.....18
	●	●			●	●			El Cazador 22
	●	●			●	●			Galactic Mover 21
X									Locación de Programas 42
			●				●	●	**Monitor-LM-1**: Para cargar Programas en LM desde nuestra Rev. **Programación Popular**40
			●				●		Mega-Basic.....25
●									Motocicleta..... 24
				●					Oso Perezoso (cont) 17
X									Protección de Programas.....42
X									Utilizando la Impresora 42
				●					Veneno.....38

VARIOS

AVISOFTI: Bolsa de Usados y Empleos 8
Concurso de Programas: Ganadores y Premios
(1er. Premio: una computadora).....10
SoftiClub: Club de lectores.....4

SOFTI CLUB

Queridos amigos:

Veo con una enorma satisfacción (pero con espanto y horror, por otra parte) como crece en forma más que geométrica (exponencial, yo diria), la pila de sus cartas que no hemos podido responder por falta de espacio. He apelado a distintos métodos: achicar la letra, agrandar las "columnas" tipográficas, "recortar" las cartas eliminando las cuestiones que puedan no interesar a los demás y los elogios (no las críticas), sintetizando mis respuestas a punto tal de parecer parco y hasta a veces descortés, etc. Pero a pesar de todo, el problema se me sigue agravando...

También me resulta muy dificil seleccionar cuáles cartas responder pues quedan muchas otras muy buenas e interesantes en el tintero. Además me siento muy mal porque me parece que estoy dejándolos de lado a muchos de ustedes, aunque juro por mis Softitos que no es así!!!

No les pido que me den la solución. Sólo quiero que sepan esto y que no se sientan mal ni crean que no interesan sus opiniones ni mucho menos que no tengo la más amplia disposición de atender sus inquietudes.

Por ello voy a proponerme a partir del próximo número nuevos métodos: intentaré agrupar preguntas, mencionar a los que no pueda contestar; no sé, ya veré qué puedo hacer... Si se les ocurre algo no duden en escribirme.

Muchas gracias a todos, y perdonen ...

Softi

Me complace enviarles estas líneas no sólo para felicitarles por las ideas que vuelcan en su revista, sino por la signular manera de tocar los temas "Tabú" de la Computación y que son fáciles de entender por nosotros los neófitos.

Como recién empiezo mis estudios, a la par de seguir el curso de Uds. en la revista creo conveniente presentarles algunas dudas y exponerles otras ideas que puedan llegar a beneficiar el entendimiento que tendría que existir entre Uds. y los lectores, teniendo en cuenta lo específico que busca un medio de comunicación social: la comunicación mutua de las partes. Por supuesto que no voy a caer en lo teórico dejando de lado los aspectos comerciales. Una me inspira confianza esta buena idea de Uds. y convencido de ello trato a través de ésta no sólo colaborar sino quedar "enganchado con Uds.", es decir hacerme socio.

1. Ideas (colaboración):

- tener en cuenta qué es lo que piensa alguien que le encante esta temática, y no entiende nada... pero quiere entender.
- Aquel que quiere empezar está mareado con la propaganda. Cómo y qué le conviene hacer para comenzar, qué cursos y en qué orden (programador, perfoverificador, etc.)
- En dónde recomiendan Uds. (con vuestra experiencia) hacer esos cursos. Cuándo normalmente aparece la revista? Por ej. Hoy no conseguí la de agosto
- Específicamente, en qué horarios se puede utilizar el T.E. al que Uds. hacen mención (las 5 veces que intenté, no contestó nadie).

2. Dudas del curso Basic:

a. Del tema 1, ¿cuáles son específicamente los términos que hay que retener? Seña importante, terminar la unidad didáctica con una síntesis que luego sea reflejada en el test.

b. Hay algunos términos técnicos que los neófitos no manejamos pero que Uds. hacen mención en el curso y luego el que quiere profundizar un poco más no encuentra la explicación. Sería bueno que se extendieran: Ej.: Microcoordenadores, Programa "Objeto", etc.

c. Cómo puedo hacer para conseguir los tres primeros números que no tengo? Sé que con el N° 2 hay problemas por la demanda, por lo que les pido que me agreguen a la lista de espera.

Néstor Montero
Capital

1. Gracias por tus sugerencias, tomamos buena nota de las mismas. En cuanto al TE (tu carta es del 13/8) ya sabrás que a la fecha nos hemos mudado y hemos mejorado notoriamente el servicio de atención telefónica. De todos modos, informo nuevamente que la vía más efectiva para comunicarnos es a través de la correspondencia que contestamos en esta sección.

2.a. Muy buena tu pregunta, pero ya teníamos pensada la respuesta. Resulta, mi estimado *Néstor*, que buscamos que nuestro lector "piense" y deba esforzarse para "aprehender" los conceptos de programación. Caso contrario, seguramente no podrá llegar a ser más que un programador frustrado. Por eso no nos parece adecuado hacer nosotros una síntesis o resumen de cada unidad didáctica. Sí nos parece muy bien que ese resumen lo haga cada uno de nuestros lectores/alumnos, eso forma parte del proceso de aprendizaje (aprender cuesta).

2.b. Respecto de los nuevos términos, buscamos que queden explicados a lo largo del curso, pero de todos modos vamos a repasar el tema.

2.c. Ahora podés solicitar con el talón adjunto la totalidad de los números atrasados disponibles.

Gracias *Nestor* y hasta la próxima ...

Me llamo Gonzalo Rafael Fernández, ante todo quiero contarles que fui (creo que desde este momento no lo seré más) asiduo lector de las revistas españolas Micro Hobby, Mi Computer y Tu Micro. Lástima que todas las ofertas, novedades y demás yerbas son para los españoles, está bien que con ellos nos actualizamos pero siempre es mucho mejor tener una revista que nos hable de cosas que estén en nuestro mercado, y es así que leyendo Programación Popular encontré la revista justa para nosotros.

Yo soy poseedor de una Sinclair 1500, pero lo más pronto posible quiero comprarme una Spectrum o una Commodore 64.

Demás está decirles que su publicación es buenísima pero comparando con otras revistas (las gallegas) veo que ustedes o la gente que publica en su revista no saca los precios de las máquinas, periféricos, etc., aquí eso nos sería de gran utilidad, para comprar y para otras muchas cosas.

Pregunta: ¿Hay alguna diferencia entre la Spectrum de Czerweny y la Spectrum de Timex-Sinclair?

Gonzalo R. Fernández
Resistencia, Chaco

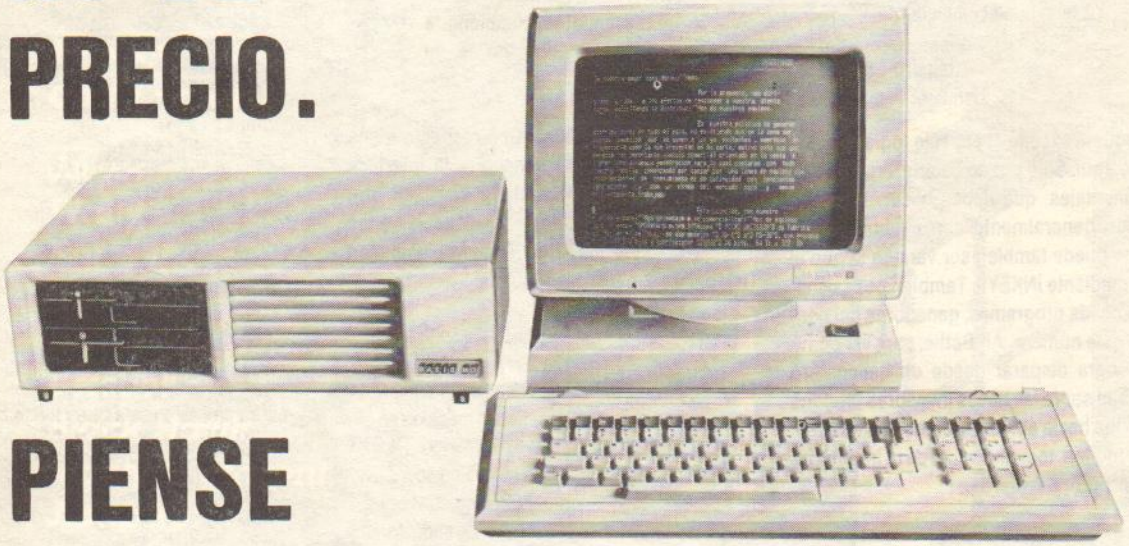
La *Spectrum CZ2000* (Czerweny) es igual a la *ZX Spectrum* europea de 48 K RAM.

Gracias por tus conceptos, *Gonzalo*. Adiós...

Aprovecho la oportunidad para felicitarlos por la revista, muy buena la presentación y muy bueno el contenido y si algo le faltara, sería la inclusión de artículos sobre hardware que uno mismo pueda hacer, aun aquellos que no tengan profundos conocimientos de electrónica. Otra cosa interesante sería la publicación de avisos publicitarios de casas que comercialicen Software y Hardware para las distintas marcas de M.C. y si fuera posible con precios.

Cándido Osvaldo Sáez
Concepción, Tucumán

ENTRE EL MAS GRANDE DE LOS MICROS Y EL MAS CHICO DE LOS BASIS HAY GRANDES DIFERENCIAS. NO DE PRECIO.



PIENSE EN GRANDE LLEGUE A BASIS.

BASIS

Un corazón argentino

BASIS S.A. - INFORMATICA PARA EMPRESAS ARGENTINAS - PASEO COLON 823 8º P.
(1063) BUENOS AIRES - TEL. 362-6048/6043/6051-361-1718

eco

viene de pág. 4

Realmente no pensábamos que a muchos lectores pueda interesarle meter mano en los circuitos, máxime habiendo buenas revistas técnicas específicas sobre electrónica. De todos modos, tomamos nota de tu idea y veremos si hay otros amigos que también quieran este tipo de notas en nuestra *PP*.

Respecto de los avisos con precios, nosotros también quisiéramos lograrlos, pero la mentalidad (y la realidad, por supuesto) inflacionaria de tantos años, inhibe generalmente anuncios de este tipo. Esperemos que esto vaya cambiando con una mayor estabilidad monetaria.

Gracias por todo, *Cándido*...

Me dirijo a ustedes con el fin de solicitarles un informe acerca de las instrucciones necesarias para que de la parte inferior de la pantalla del televisor, cuando yo toque una tecla desde ese lugar, salgan como tiros. Yo tengo una computadora **TK 85**.

Edelmiro Centorani
Ushuaia, T. del Fuego

Tu pregunta excede el espacio lógico de que dispongo aquí. Sólo puedo decirte, en términos muy elementales, que debes hacer un PRINT controlado generalmente con un bucle FOR-NEXT que puede también ser variado según el teclado, mediante INKEY$. También podés analizar uno de los programas, ganadores del concurso de este número, Air Battle, para ver cómo *marcelo* logra disparar desde un helicóptero sus bombas sobre las naves invasoras (aunque desde arriba hacia abajo, no como vos querés). De todos modos, tomamos nota de tu inquietud.

Hasta pronto, *Edelmiro*...

Yo tengo 11 años, y estudio inglés, mi hermana se llama Patricia Andrea y estudia conmigo inglés; y Gonzalo, mi hermanito Gonzalo Gastón que tiene 4 años cumplidos en agosto (ya conoce cual es enter y shift) conoce también, de memoria los nombres de distintas clases de computadoras.

Graciela Liliana García
Com. Rivadavia, Chubut

Estamos trabajando en una tabla de conversión para facilitar (cuando sea posible) la reconversión de programas. El tema no es nada fácil...

Gracia por todo *Graciela* y felicitalo a *Gonzalo* de mi parte. Chau ...

Como Ud. sabrá, estoy suscripto a ella desde su primer ejemplar y en esta ocasión le envío un cheque para ser suscripto nuevamente a 6 números más, a contar desde el N° 8.

Si Ud. analiza mi situación, verá que en menos de tres meses, me he suscripto dos veces a vuestra revista (12 ejemplares). Esto demuestra mi interés por ella y lo que ésta significa para todos los que de una u otra manera somos poseedores de un ordenador personal, a la vez que para todos los que aún no lo poseen.

Es por esto que le pido tenga a bien analizar la posibilidad de asociarme sin cargo al tan prestigioso **Softi Club**.

Le comento que soy poseedor de un ordenador **TS 2068** –con mucho orgullo–, me interesan los programas didácticos y de juegos y aprovecho esta oportunidad para preguntarles si en realidad existe la posibilidad (como se dice por allí) de construir un sencillo periférico que permita adaptarle a la **TS 2068**, una impresora de papel común o un microdrive.

Víctor R. Ravera
Córdoba

Todo es posible, *Víctor*. Con respecto a una impresora, se trata de conseguir la interface adecuada, cosa no demasiado fácil, por otra parte. El microdrive sólo sería utilizable en los últimos modelos de *TS2068*. Podés consultar en Tisinc, Datex o Valro tus dudas.

Hasta pronto, *Víctor* ...

Tengo 11 años, poseo una computadora **TK 90 X** y me gustaría que pongan un programa sobre alarmas para mi computadora y también canciones.

Quería preguntarles si disponen de impresoras y cuanto están, también me gustaria ser socio del **Softi Club**.

Los felicito por la revista, es muy importante y ayuda a aprender cosas nuevas que yo no conocía.

Además solicito información y precios de computadoras y cassettes con juegos para microcomputadoras.

Marcelo Pablo Villagra
Rio Grande - Tierra del Fuego

Te informo que nuestra Revista PP no vende productos de computación, logrando de esta forma una independencia de criterio para brindar un buen servicio a nuestros lectores. La información comercial la recibirás dentro de unos meses a través del *Softi Club*.

Gracias por tus conceptos y tomamos nota de tus necesidades. Hasta pronto, *Marcelo* ...

***A continuación les ofrecemos la adaptación del programa "Calendario" para TI99/4A* que nos enviara el amigo *Iván F. Sarachman* (*P.P.* Nro. 5, pág. 6) y que recién ahora hemos podido hacer un lugacito para publicar:**

```
10 REM TRADUCIDO PARA TI 99/
4A POR IVAN SARACHMAN 1985
20 REM CALENDARIO
30 CALL CHAR(129,"0010301010
10103C")
40 CALL CHAR(130,"001C220204
08103E")
50 CALL CHAR(131,"001C22020C
02221C")
60 CALL CHAR(132,"00040C1424
3E0404")
70 CALL CHAR(133,"007C407804
044438")
80 CALL CHAR(134,"000C10203C
22221C")
90 CALL CHAR(135,"003E020408
101010")
100 CALL CHAR(136,"003844443
8444438")
110 CALL CHAR(137,"003844443
C040830")
120 DIM M$(12)
130 CALL CHAR(128,"003844444
4444438")
140 CALL CHAR(138,"007C22222
222227C")
150 CALL CHAR(139,"007E42424
242427E")
160 CALL CHAR(140,"00665A5A4
2424242")
170 CALL CLEAR
180 CALL SCREEN(12)
190 CALL CHAR(144,"AAAAAAAAA
AAAAAAA")
200 CALL COLOR(15,5,16)
210 CALL COLOR(13,16,7)
220 CALL COLOR(14,16,7)
230 GOSUB 1050
240 GOSUB 680
250 GOSUB 870
260 GOSUB 980
270 L=8
280 FOR C=1 TO CD
290 P=6+4*D-(LEN(STR$(C)))-2
300 IF P>29 THEN 550
310 IF P<6 THEN 580
320 XLL=P
330 YLL=L
340 LL$=STR$(C)
350 GOSUB 1130
360 D=D+1
370 NEXT C
380 CALL HCHAR(1,2,144,30)
390 CALL VCHAR(1,31,144,22)
400 CALL HCHAR(22,2,144,30)
410 CALL VCHAR(1,2,144,22)
420 XLL=4
430 YLL=1
440 LL$=M$(M)
450 GOSUB 1130
460 XLL=25
470 LL$=STR$(B)
480 GOSUB 1130
490 PRINT "OTRO CALENDARIO (
S/N)? ";
500 CALL KEY(0,KEY,STS)
510 IF STS=0 THEN 500
520 PRINT CHR$(KEY)
530 IF KEY=ASC("S")THEN 10
540 STOP
550 L=L+2
560 D=0
570 GOTO 290
580 N$=" "
590 C$=STR$(C)
600 FOR Q=1 TO LEN(C$)
610 N$=N$&(CHR$(ASC(SEG$(C$,
```

sigue en la pág. 8

CIBERNE



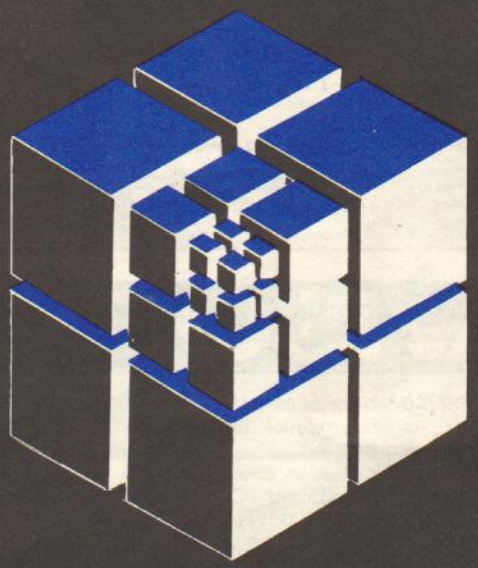
SOFTWARE

Las últimas novedades europeas
Imaginación en los juegos
Inteligencia en los utilitarios
de microcomputadores

Soft educativos
Análisis combinatorio
Estadística

Astrología
Carta natal

Juegos
Wally
Babaliba
Allien 8
Rocky
Béisbol
Black Hawk
Monty is inocent
Jump set
Starstrike



Utilitarios
Beta Basic
Master Diseño

Solicite nuestro catálogo de juegos y utilitarios para la línea Sinclair, Microdigital y TS 2068 de 2K, 16K y 48K

Balcarce 1053 - L.10 - T.E. 362-0086 - C.P. 1064 Cap. Fed.

Distribuye:
Microvideo: Sarmiento 1568, 6° "B", 1042 Cap. Fed. T.E. 35-0164

Ref. 16

viene de la pág. 6

```
Q,1))+80))
620 NEXT Q
630 XLL=P
640 YLL=L
650 LL$=N$
660 GOSUB 1130
670 GOTO 360
680 REM ---INGRESO MES/ANO--
-
690 CALL CLEAR
700 PRINT "INGRESE ANO DESEA
DO"
710 PRINT "CON TODAS SUS CIF
RAS"
720 INPUT A
730 IF A<1582 THEN 690
740 CALL CLEAR
750 PRINT "INGRESE MES DESEA
DO";"COMO NUMERO ENTRE 1 Y 1
2"
760 INPUT M
770 IF (M<1)+(M>12)THEN 740
780 CALL CLEAR
790 FOR T=1 TO 3
800 CALL HCHAR(5,3+T,137+T)
810 NEXT T
820 XLL=5
830 YLL=5
840 LL$=" LUN MAR MIE JUE VI
E SAB "
850 GOSUB 1130
860 RETURN
870 REM ---FACTOR---
880 F=365*A+31*(M-1)+1
890 B=A
900 IF M>=3 THEN 960
910 A=A-1
920 F=F+INT(A/4)-INT(.75*(A/
100)+1)
930 D=F-INT(F/7)*7
940 D=ABS((D-1)*(D>0)+6*(D=0
))
950 RETURN
960 F=F-INT(.4*M+2.3)
970 GOTO 920
980 REM ---CANT DE DIAS---
990 CD=31
1000 IF (M=9)+(M=4)+(M=6)+(M
=11)THEN 1010 ELSE 1020
1010 CD=30
1020 IF M=2 THEN 1030 ELSE 1
040
1030 CD=28+ABS((INT(B/4)=B/4
)-(INT(B/100)=B/100))
1040 RETURN
1050 REM  ---CARGA DE MESES-
-
1060 CALL CLEAR
1070 RESTORE
1080 FOR I=1 TO 12
1090 READ M$(I)
1100 NEXT I
1110 DATA ENERO,FEBRERO,MARZ
O,ABRIL,MAYO,JUNIO,JULIO,AGO
STO,SEPTIEMBRE,OCTUBRE,N
OVIEMBRE,DICIEMBRE
1120 RETURN
1130 FOR LL=1 TO LEN(LL$)
1140 CCL=ASC(SEG$(LL$,LL,1))
1150 CALL HCHAR(YLL,XLL+1+LL
,CCL)
1160 NEXT LL
1170 RETURN
```

Nuevamente, gracias, *Iván*...

Mi saludo y mi deseo de que esta revista se convierta con el tiempo en la mejor revista de computación en el mercado nacional y por qué no del mundo (mucho, ¿no?). Bueno, sin exagerar realmente es buena pues se ha encarado el tema con un criterio didáctico amplio y claro lo cual no es muy común en las publicaciones de este tipo, pues o son demasiado técnicas o muy superficiales, esta claridad permite al recién iniciado como es mi caso comprender bien la estructura del programa con sus variables correspondientes.

Tengo dos sugerencias que efectuarles: yo tengo una **TK 85** y mi mayor problema es uno que tal vez les suceda a muchos, la compatibilidad entre programas; a menudo sucede que nos interesa un programa y entonces nos encontramos con sentencias que no figuran en nuestra máquina de manera que mi solicitud se centra sobre una especie de tabla de equivalencias y sentencias entre las diferentes marcas del mercado. La segunda sugerencia sería que incluyan en próximos números un curso de Fortran y las posibilidades de adaptación de este lenguaje a las diferentes máquinas en plaza.

Carlos José Alberola
Mar del Plata, PBA

1. Estamos preparando una nota sobre el tema de equivalencias de programación entre las máquinas de nuestro mercado, la que pondremos a tu disposición y de nuestros demás amigos lectores muy pronto.

2. Tomamos nota de tu inquietud sobre un curso de Fortran.

Gracias, *Carlos*, y hasta pronto ...

AVISOFTIAVISOFTIAVI

OFREZCO servicios de computación p construccion costos. 86-9230.

EMPLEADO administrativo, conoc. computación, operador IBM. PC, graboverificador IBM 3742. Estudiante analista programador full time. T.E. 921-2791.

VENDO Seikosha GP 100. u$s 350. Interfase TK u$s 70. 35-9360. Angel.

VENDO Sinclair CZ 1000 con manual, transform., cableado y dos cassettes de juegos. 784-0623. Luis o Juan.

VENDO TI99 4A con modulador, cajas de periféricos, controlador de diskettes, drive diskettes, tarjeta RS232, Parsec Basic II u$s 1.250 798-4350. Sr. Llorens.

COMPRO prog. para TI99. 781-0518. Sebastian.

COMPRO Extended Basic para TI99 4A. T.E. 248-6384. Marcelo.

VENDO TI99 4A Pal N Basic extendido; mini memory; caja periféricos; disketera; expansión de memoria; compilador lenguaje forth; editor assembler; 50 juegos en disquette. T.E. 46-0979. Enrique.

COMPRO VOL 2 TK Computadores=Programação. Processamento de dados" de Flavio Rossini - Edit Moderna - AMO, Belgrano 626, (4105) Manantial, Tucumán.

INTERCAMBIO experiencias TS 2068 e impresora Latindata GP-100 c RS 232. Carlos E. Silva - Uruguay 672 - (3509) Gral. San Martín, Chaco.

INTERCAMBIO listados. Ruego a D. Weselka y A. Cano me escriban. Carlos Vecchi - Mendoza 266 - Resistencia, Chaco (3500).

VENDO Sinclair 1000 con ex. 16 KB, manual castellano, libros, programas de juegos y cables. T.E. 253-7111.

CLUB TK85 con 20 programas y manual. Fco. Acuña de Figueroa 450, 1° "F".

SE OFRECE programador en Basic (IBM Televideo TS 2068 NCR) y Cobol (Televideo NCR). 801-7229.

INTERCAMBIO comentarios usuarios TK85. 207-8882. Darío Gustavo.

COMPRO usado o nuevo cartucho "Basic Extendido" para TI 99 4 A - Escribir Francisco Puente B° YPF Cañadón Seco (9013) Casa 6246 - Santa Cruz.

INTERCAMBIO programas y dudas con usuarios TK 2000. Mario Aníbal 572-3419.

VENDO programas para CZ 1000/1500 TK 83/85 CZ 2000 TK90 795-4544.

PERMUTO Por Atari o vendo ~~A~~ 100, CZ 1000 16K Enciclopedia computación Nueve casetes, Manual Cableado Colección revista 551-1206.

VENDO TK 85 casi nueva con garantía, 30 juegos y Joysticks ~~A~~ 180 654-4729.

VENDO TIMEX SINCLAIR 100 con barra de expansión A 16 KB Manuales y programas 768-6551.

VENDO Casetes con juegos grabados Juan Manuel 785-7292.

CANJEO O VENDO bicicleta H. France por TI 99 - A Arreglo diferencia 769-4554.

VENDO TK 2000 y TK 85 como nuevas con programas 542-3625. Después de las 19 hrs.

VENDO TK 85 16 KB con fuente, manual, 20 programas, 1 joystick Oscar - F.A. de Figueroa 450, 1° "F" Capital.

VENDO Sinclair 1500 con 12 casetes y jeugos ~~A~~ 120 571-9928.

COMPRO Extended Basic para TI 99 4A Marcelo 248-6384.

PERMUTO TK 83 con expansor memoria impecable por TK 90 X. Commodore 64, TI 99 4A, Spectrum, TS 2068 - Diferencia a convenir. 0388-21726 Lucio (Gral. Villegas).

INTERCAMBIO nformación, programas, ideas con usuarios Commodore 64 Lucy Pravia - Av. San Martín 1744 (5174) Huerta Grande - Córdoba.

SOLICITO me escriban pasándome programas y trucos para TS 2068 Javier C. Gallinar - H. Quiroga 4555 (7600) Mar del Plata.

INTERCAMBIO Programas con usuarios CZ 1500 y compatibles. Diego A. Langes - Pringles 965 P.B. (1183) Capital Federal.

AYUDO a aclarar dudas usuarios Micro-computadoras. Soy técnico electrónico y programo en Assembler y Basic - Poseo conocimientos Logo y Fortram. Gustavo G. Mislavsky - Aguirre 691 - 6° A (1414) Cap. 771-2084.

latindata
MPF-3

Después del primer paso, la elección más inteligente para aprender a caminar en computación.



Si usted necesita un microcomputador que le brinde posibilidades de trabajar **en serio**, que ayude a un aprendizaje más rápido y efectivo y cuente también con los más espectaculares entretenimientos del mundo, la elección es una:

latindata MPF-3, la compatibilidad total con los más populares y útiles programas de software moderno y la más conveniente relación entre prestaciones y precio

latindata MPF-3 TOTALMENTE COMPATIBLE CON apple II

Consulte a nuestros agentes de todo el país.

latindata s.a.

Pte. R.S. Peña 628 piso 1° (1035) Cap. Teléfonos: 30-8943 33-7993 34-0959/7860

Key Advertising

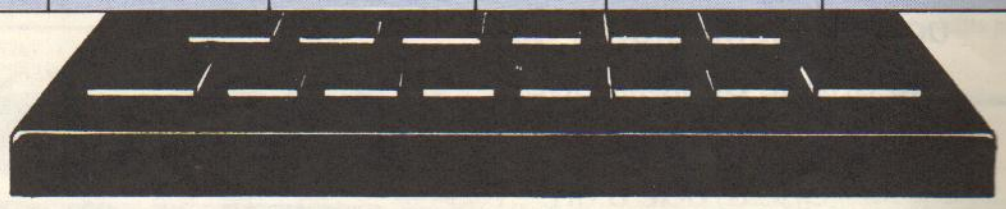
ref 17

AHORA PODES GANAR UNA CZ 2000 (SPECTRUM) TODO LOS MESES

CONCURSO NACIONAL DE PROGRAMAS

Ganadores del mes (PP Nro.8)

Premio	Ganador	Programa	Premio	Ganador	Programa
₳ 25.- en efectivo	**Juan C. Bruni** La Reja, PBA	Ataque Espacial TS2068	₳ 25.- en efectivo	**Mario V. Simes** Colonia Caroya, Córdoba	Programas en High Speed TK85
	Raul D. Lambertucci Pinamar, PBA	El Cazador TS1500		**Ing. Raúl D. Vera** Sgo. del Estero	Costos de Fabricación TK85
	Gerardo J. Mateos Paraná, Entre Ríos	Galactic Mover ZX81	Una Beca de Computación Basic en EPI	**Luis Briones** Bahía Blanca, PBA	Procesador de Textos TI99-4A
	Rodolfo Miranda La Plata, PBA	Veneno TI99-4A		**Orlando Cerocchi** Chubut	Tierra S.O.S. TK85
	Héctor R. Molena San Martín, PBA	Cubo Mágico Spectrum		**Diego H. Conde** Santa Rosa, La Pampa	Sas-Man TS2068



Premios

- ₳ **25.- (Veinticinco australes) en efectivo** por cada programa publicado en **PP**
- Una Computadora **CZ2000 (Spectrum) de 48 K**, donada por **Czerweny Electrónica SA**, para ser entregada todos los meses al autor del mejor programa de los publicados en los últimos 12 (doce) números de nuestra Revista **Programación Popular**. La selección del mejor programa del mes se hará con la participación de los **lectores** (ver **Bases** y **Talón de Voto** adjunto) quienes al votar intervendrán gratuitamente en un sorteo especial de cada mes (cursos de computación y otros importantes premios)
- Participación en el concurso mensual, semestral y anual, con importantes premios para todos los programas del ranking
- Inscripción del autor en la lista de **Colaboradores** de tu Revista **Programación Popular** y en el **Cuadro de Honor**

Bases

- Programas **INEDITOS** en casete o diskete (se admiten más de 1 por autor). El casete/diskete se devolverá al autor a partir de los 30 días de recibido en la Editorial
- Máquinas: **Commodore 16/64, CZ1000/1500/Spectrum, MSX, TI99/4A, TK83/85/90X/2000** y **TS2068** (lista de microcomputadoras mas corrientes en el país, no excluyente)
- Acompañar, además del casete o diskete, listado del programa (emitido por impresora, a máquina o manuscrito con letra de imprenta) y texto explicativo, todo **FIRMADO POR SU AUTOR**. Es aconsejable (no imprescindible) remitir también la estructura del programa o diagrama de logica y la tabla o lista de las variables. También deben acompañarse los datos personales, domicilio completo, edad y computadora en la que fue realizado o el programa y grabado el soporte magnético enviado (casete o diskete) como asimismo los K de memoria de la máquina.
- Remitir el material **por Correo Certificado** a:

Ediciones del Futuro S.A.
Revista **Programación Popular**
Maipú 464, Piso 8, Oficina 801
(1006) Capital Federal

- **Jurado:** en general, la selección la realizará personal técnico altamente calificado de la Editorial.
  Para elegir el mejor Programa del mes cuyo autor ganará una **CZ2000 (Spectrum) de 48K**, el Jurado lo formarán: **1 voto** de los **lectores** de nuestra revista **PP, 1 voto** de **Czerweny Electrónica SA** y **1 voto** de **Programación Popular**. Resultará ganador el programa que tenga mayoría de votos. Si los 3 votos difieren, ganará el seleccionado por los lectores. Se labrará un acta ante **Escribano Público** de los resultados.
  Para el próximo mes, se computarán los votos recibidos hasta el dia **10.1.86** y la selección se realizará el día 15.1.86, informándose el ganador en nuestro próximo ejemplar de **PP** (Nro. 9).
- Regirán las restantes bases publicadas en el Nro. 5 de **PP**

A SU COMPUTADORA EQUIPELA CON LA LINEA DE ACCESORIOS MAS COMPLETA.

Usted puede adquirir la línea más completa de accesorios, programas y periféricos, con la experiencia y el respaldo de los especialistas, en las mejores condiciones y con nuestro exclusivo plan de financiación de 3 a 12 cuotas.



- **Computadoras.**
- **Interface para joysticks.**
- **Joysticks.**
- **Juegos en Cassettes.**
- **Interface para Impresoras.**
- **Impresoras.**
- **Lápiz Optico.**
- **Accesorios.**







A SU COMERCIO SUMELE LAS VENTAJAS DE VICONEX

- Amplio surtido.
- Stock permanente.
- Entrega inmediata.
- Los mejores precios.
- Planes de financiación hasta 12 meses.

Lo esperamos porque jamás la computación estuvo tan cerca suyo.

TARJETAS DE CREDITO

VICONEX

SU ALIADO EN COMPUTACION

Av. de Mayo 767 (1084) Capital.
Tel. 30-3301 / 33-2106 / 33-8371.
Lun. a vier. abierto hasta las 20 hs.; sábados abierto medio día.

CURSO BASIC V

DIAGRAMACION - CONDICIONALES - IF/THEN

5.1. ENTENDER EL PROBLEMA

Antes de seguir adelante, insistamos en lo importante de la primera de las seis fases en las que, según dijimos, se divide la tarea del programador:

1. Entender el problema

Con el mayor de los énfasis aseguramos que, si no se cumple este requisito,el programa que estamos a punto de diseñar está condenado al fracaso.

''Entender el problema'' involucra el conocimiento detalladao de tres cosas:

a) Los **datos** que será necesario proporcionar a la computadora.

b) El **proceso** de transformación que debe aplicarse a los mismos.

c) La naturaleza y cantidad de la información requerida como **resultado**.

En a), en el momento de considerar qué datos vamos a entregar al ordenador, es cuando más que nunca debemos recordar que, pese a su aspecto inteligente, este es *un ciego seguidor de instrucciones; no es capaz de descubrir por sí solo qué parte de la información necesaria falta entregarle, ni de descartar la que sobre*. Es tarea nuestra decidir qué datos son relevantes para el tratamiento del problema particular que nos enfrentamos. Al respecto recordar 4.2. donde un ligero cambio en los datos a manejar obligó a una serie de modificaciones en el programa que culminó con la aparición de la instrucción **INPUT**.

Los mismo vale para b). El procesamiento de la información se hará a través de un *algoritmo que debe estar precisado en todos sus detalles*; en general, es tanto más valioso un algoritmo cuanto más problemas distintos resuelve. Pero la versatilidad en este

RESPUESTAS AL TEST DE LA LECCION IV

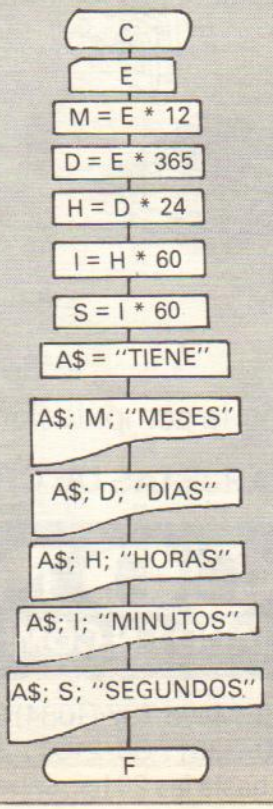
1 - Generalmente se realiza en seis pasos, que son los siguientes:
a) Entender el problema.
b) Diagramar un algoritmo que lo resuelva.
c) Traducir el algoritmo al lenguaje de computación que se va a utilizar.
d) Introducirlo en el ordenador.
e) Probarlo y optimizarlo.
f) Documentarlo.

2 - **INPUT** es una instrucción BASIC que nos permite introducir desde el exterior datos a la computadora, mientras está corriendo un programa.

3 - Están bien: C, D, G.

Están mal: A, B, E, F, H.

4 - Diagrama



Basic

```
100 PRINT "QUE EDAD TIENE?"
110 IMPUT E
120 LET M = E * 12
130 LET D = E * 365
140 LET H = D * 24
150 LET I = H * 60
160 LET S = I * 60
170 LET A$ = "TIENE"
180 PRINT A$; M * "MESES"
190 PRINT A$; D * "DIAS"
200 PRINT A$; H * "HORAS"
210 PRINT A$; I * "MINUTOS"
220 PRINT A$; S * "SEGUNDOS"
230 STOP
```

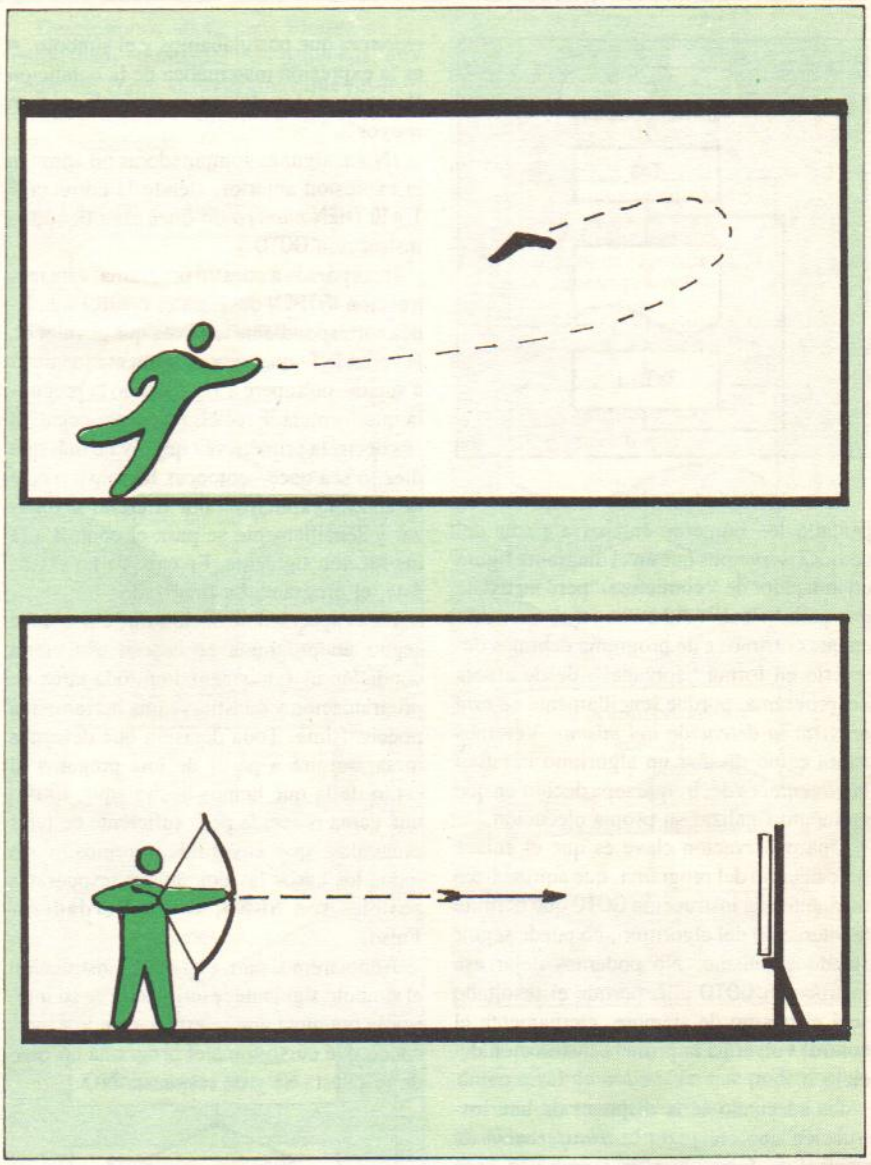
■ En caso de responder mal alguna pregunta

Error en	Repasar
1	4.1. - P.P. Nº 7
2	4.2 al 4.4. - P.P. Nº 7
3	2.5. / 2.7. / 2.8. - P.P. Nº 5 4.2 al 4.4. - P.P. Nº 7
4	P.P. Nº 7 (y anteriores, si fuera necesario)

sentido es a veces bastante difícil de lograr y consume tiempo de programación que quizás –y según el tipo de problema– no valga la pena gastar. Lo ideal es obtener una solución de compromiso, intermedia entre un programa "especializado" que resuelva un único caso particular y uno "enciclopédico", dedicando el tiempo de optimización que merezca la importancia del problema.

Una vez más, es necesario recordar aquí que la construcción del algoritmo pasa por considerar detalles que parecerían nimios y que uno tiende a dar por sobreentendidos: ver, por ejemplo, 3.9, donde era necesario especificar cómo generar la serie de los naturales, o, en 4.2., la "adaptación" a la computadora de la "regla de tres simple".

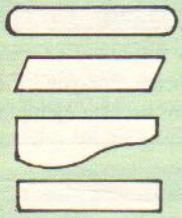
c. finalmente, es igualmente importante pensar en la *forma de presentación de la información obtenida*: a quién está dirigida, qué documentación hace falta de la misma.



5.2. DIAGRAMAS DE FLUJO

En un capítulo anterior (3.4.) apelamos a la intuición de nuestros lectores presentando en forma sumamente breve los **diagramas de flujo**, especie de representación gráfica de un algoritmo cualquiera que frecuentemente sirve para aclarar nuestras ideas acerca de cómo trabaja un futuro programa. Es obvio que el paso "entender el problema" es previo a la confección de un tal diagrama de flujo, puesto que éste no es más que una representación simbólica de algo que ya debe existir en nuestro pensamiento. Vamos a hacer un repaso de la simbología utilizada entonces y la ampliaremos.

Hasta el momento nos hemos encontrado con:

SIMBOLOS	REPRESENTA A:	BASIC
 *	INICIO/FIN DE PROGRAMA	**NEW-END/STOP**
	INGRESO DE DATOS	**INPUT**
	SALIDA DE DATOS	**PRINT**
	OPERACION DE ASIGNACION	**LET** (algunas veces opcional)

(*) A veces la instrucción **INPUT** suele representarse mediante el símbolo



que obedece a una razón histórica; recuerda a una tarjeta perforada, y ello es porque las tarjetas perforadas fueron el primer medio de ingreso masivo de datos a un computador. Este símbolo es, probablemente, más clásico que el que damos en la tabla, y lo encontraremos en muchos libros y manuales.

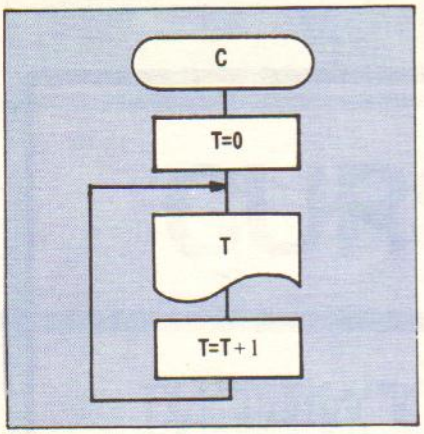
El símbolo correspondiente a la instrucción PRINT tiene una historia similar; antes de que se implementara el tubo de video como medio de salida de datos, el habitual era la impresora de papel (precisamente **PRINT** significa Imprimir) y la forma elegida para este símbolo recuerda a una hoja de papel rasgado saliendo de una impresora.

Como nuestro medio de salida información va a ser normalmente el video, nada impediría que nuestro símbolo en cuestión fuera el siguiente:



5.3. IF/THEN

Vamos a considerar una nueva situación a partir de un ejercicio ya visto en 3.8., el programa con estructura de **contador**. Recordémoslo:



Este algoritmo generaba e imprimía en pantalla los números enteros a partir del cero. Observemos que en el diagrama figura un indicador de "comienzo" pero no existe uno para "final". Además, en el momento en que corrimos este programa debimos detenerlo en forma "forzada", desde afuera del programa, porque sencillamente *no está prevista la detención del mismo*. Veremos ahora cómo diseñar un algoritmo iterativo *inteligente*, es decir, que sepa decidir en qué momento finalizar su propia ejecución.

Una observación clave es que el enlace fin/principio del programa, que aquí se hace mediante una instrucción **GOTO** que permite la interación del algoritmo, no puede seguir siendo el mismo. No podemos dejar esa instrucción **GOTO** allí, porque el resultado será el mismo de siempre: eternamente el control volverá a la primera instrucción del ciclo.

Lo adecuado sería disponer de una instrucción que efectuara la *transferencia de control* en forma similar a como lo hace **GOTO**, pero sujeta a una cierta condición.

Supongamos querer modificar el programa contador de más arriba para obtener en pantalla la sucesión de los números enteros a partir de cero, *pero hasta el número 10*.

En ese caso, lo que deseamos es que el ciclo de impresión e incremento se repita exactamente once veces (una para el número cero, diez veces más para los números de uno a diez).

La condición a que nos referíamos antes podría extraerse del mismo planteo del problema: lo que deseamos es que el ciclo se repita *mientras la variable* **T** *no supere un valor de 10*. Nuestra instrucción de *transferencia de control condicional*, hasta ahora potética, debería tener un aspecto más o menos así:

Si **T** *es menor o igual que* **10** *entonces* **GOTO número de línea correspondiente**.

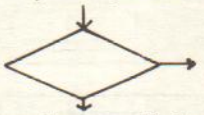
Precisamente; una tal instrucción existe y se escribe en BASIC del siguiente modo: **IF T =10 THEN GOTO** *número de línea*, donde **IF** y **THEN** son traducciones literales del *si* y *entonces* que postulábamos y el símbolo = es la expresión matemática de la condición **"menor o igual"**, que equivale a **"no mayor"**.

(Nota: algunas computadoras no admiten la expresión anterior, siendo la correcta **IF T =10 THEN** *número de línea*, omitiendo la instrucción **GOTO**.)

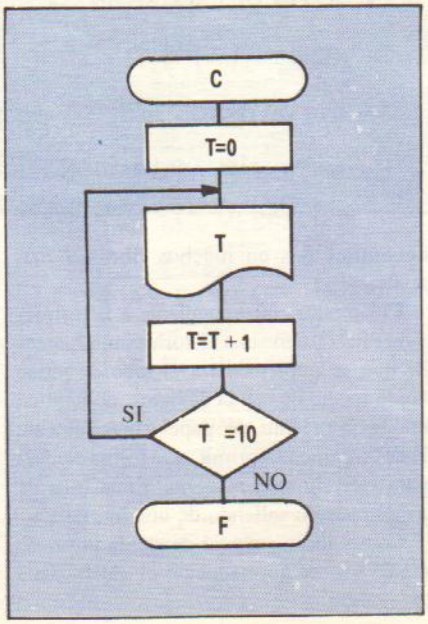
Incorporada a nuestro programa, esta instrucción **IF/THEN** desviará el control a la línea correspondiente toda vez que el valor de la variable **T** –que se ve incrementado vuelta a vuelta– no supere a 10. Cuando la pregunta que formula **IF** recibe respuesta negativa –es decir, la primera vez que **T** vale más que diez, o sea once– entonces la transferencia de control explicitada por **THEN** no se realiza, y sencillamente se pasa el control a la instrucción siguiente. En caso de no existir ésta, el programa ha finalizado.

Esta capacidad de decidir qué curso debe seguir un programa en base a una cierta condición es fundamental en toda tarea de programación y constituye una herramienta poderosísima. Toda decisión que debamos tomar surgirá a partir de una pregunta al estilo de la que hemos hecho aquí. Existe una gama reducida pero suficiente de tales preguntas, que enseguida veremos, y en todos los casos las dos únicas respuesatas posibles son **Sí-No**, o sea **Verdadero-Falso**.

Adoptaremos para esta nueva instrucción el símbolo siguiente, escribiendo en su interior la pregunta que se está haciendo e indicando qué curso toma el programa en caso de respuesta **SI** y de respuesta **NO**.

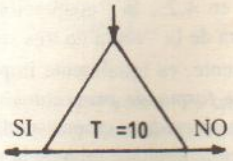


Ahora nuestro diagrama del algoritmo interativo controlado se verá así:



...diagrama que ahora **sí** incluye un indicador de final, porque esta posibilidad ha sido contemplada. Notemos que, de todos los símbolos de diagramación, el recién visto es el único al cual el control entra por un lado y sale por **dos** distintos. No olvidemos, al usarlo, prever qué curso daremos al programa en uno y otro caso.

(Nota al margen: si las preguntas que formulemos en Basic tienen sólo dos respuestas posibles, ¿por qué tenemos en el símbolo correspondiente **tres** puntos de salida, vértices del rombo? ¿No alcanzaría con el símbolo



?

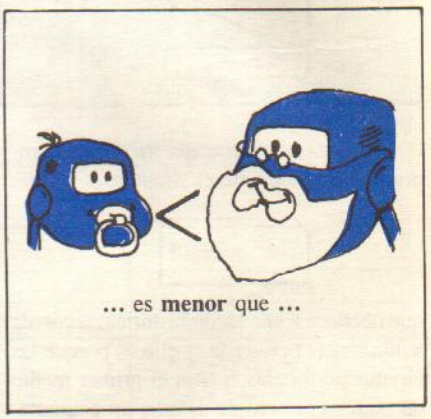
Es cierto, de hecho alcanza con este símbolo, y se utiliza a menudo, aunque el anterior es más clásico: proviene de lenguajes en que las respuestas posibles a una instrucción de transferencia de control condicional son tres. Como el lector despierto supondrá, no se trata de las respuestas SI, NO y MAS O MENOS, sino de MAYOR QUE CERO, IGUAL A CERO y MENOR QUE CERO.)

5.4. COMPARACIONES

Veamos el repertorio de "preguntas" que podemos incluir en una instrucción **IF/THEN**. Todas ellas hacen uso de uno o más de estos **operadores de comparación**:

>	**Mayor que**
<	**Menor que**
=	**Igual a**

(Si tenemos dificultad en recordar cuál símbolo representa "mayor" y cuál "menor", fijémonos en este ejemplo:



La **abertura** del símbolo apunta al lado de la desigualdad donde está el **mayor** de los comparandos.)

Con estos tres **comparadores elementales** podemos construir otros más, simplemente colocando uno al lado del otro en todas las maneras posibles:

>=	**Mayor o Igual que** (Es decir, **No Menor**.)
<=	**Menor o Igual que** (Es decir, **No Mayor**.)
<>	**Menor o Mayor que** (Es decir, **Distinto de**.)

Toda vez que escribamos dos de estos "comparadores" juntos, nuestra computadora interpretará que en medio de ambos figura una invisible palabra **"o"**.

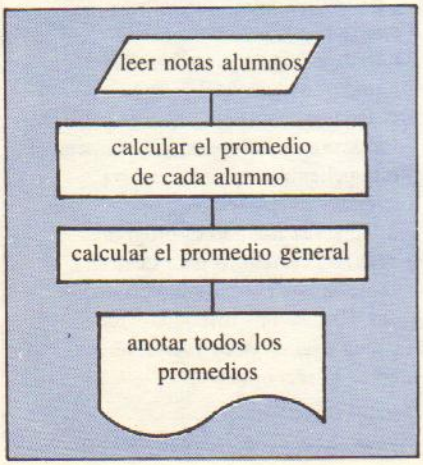
5.5. DIAGRAMAS DE BLOQUES

Ahora tenemos un conjunto completo de símbolos para representar todas las situaciones de interés en programación. Podemos introducir una idea nueva, en la que haremos uso de estos mismos símbolos; se trata del **diagrama de bloques**, que es una representación gráfica de un algoritmo parecida al diagrama de flujo pero mucho menos detallada.

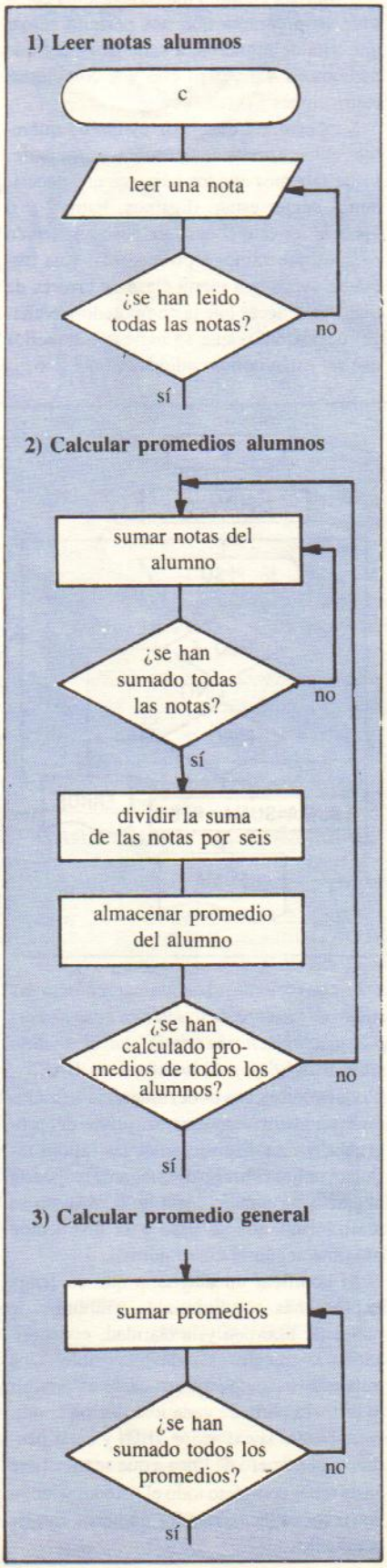
En general un diagrama de bloques *sirve como primerísima aproximación al problema, y consiste en un boceto a muy grandes rasgos del conjunto de procesos a cumplir*. La finalidad del mismo no es tanto develar la estructura del **programa** buscado, como *aclarar la naturaleza y vinculación de las grandes unidades funcionales del algoritmo*. Si el problema considerado es extremadamente sencillo, ambas clases de diagrama tienden a confundirse.

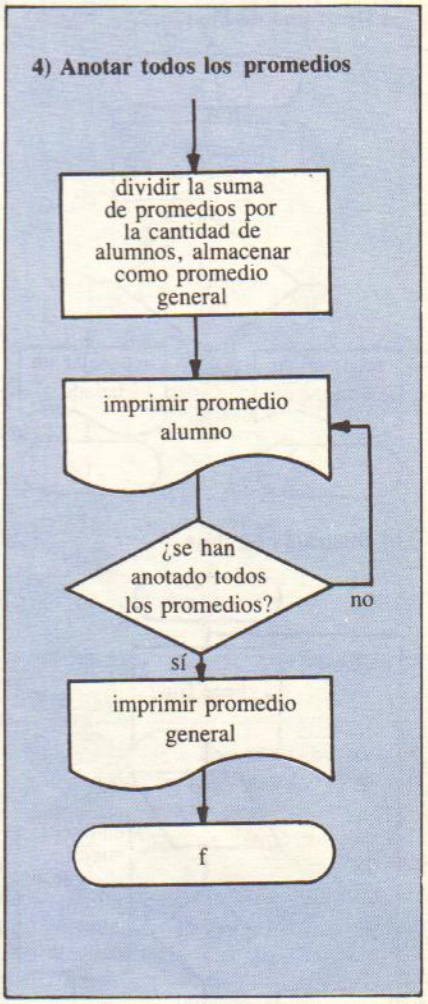
Situémonos en el caso de un profesor de escuela secundaria que desea obtener una evaluación final de una clase a partir de seis notas que ha obtenido cada alumno.

Ya sea si decide hacer las cuentas a mano, ya sea si piensa usar su computadora al efecto, las operaciones a realizar serán en líneas generales las siguientes:



Desglosemos ahora cada **bloque** de los enunciados en bloques más pequeños, procediendo como si tuviéramos que resolver cuatro problemas sencillos en lugar de uno solo, complicado:





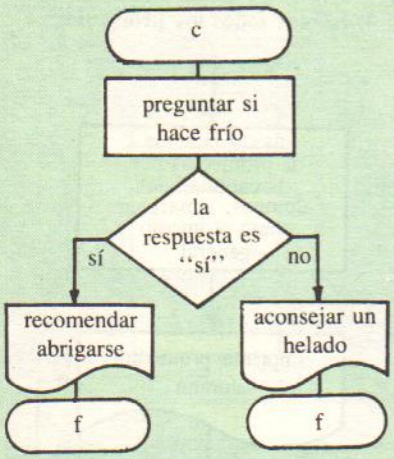
De lo visto se desprende que no existe un único nivel de análisis en que podemos hacer un diagrama de bloques, y que su minuciosidad puede aumentarse sin dificultad con sólo tomar cada bloque como si éste fuera el problema original y repetir la tarea de anáisis, esta vez en un nivel superior de profundidad. Si continuamos con este procedimiento, llegaremos tarde o temprano a un punto crítico en que ya no se puede ser más explícito acerca de las operaciones involucradas en cada bloque, quedando éstas expresadas por una única instrucción BASIC. Lo que tendremos en ese momento no es otra cosa que el viejo y querido **diagrama de flujo**, listo ya para codificar y correr alegremente en nuestro ordenador.

5.6. EJEMPLOS DE IF/THEN

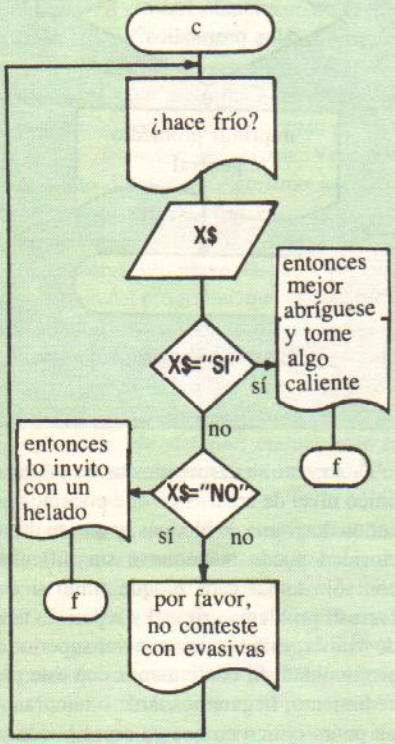
5.6.1. Veamos un ejemplo sencillo de manejo de condicionales, referidos esta vez a constantes y variables **alfanuméricas**.

Nuevamente incursionaremos en el tipo de programas de estilo *"conversación"* a que somos tan afectos. Nuestro programa no será más hábil que la mayoría de los humanos para iniciar una conversación, y optará por hablar del tiempo.

a) Diagrama de bloques



b) Diagrama de flujo



c) Codificación BASIC.

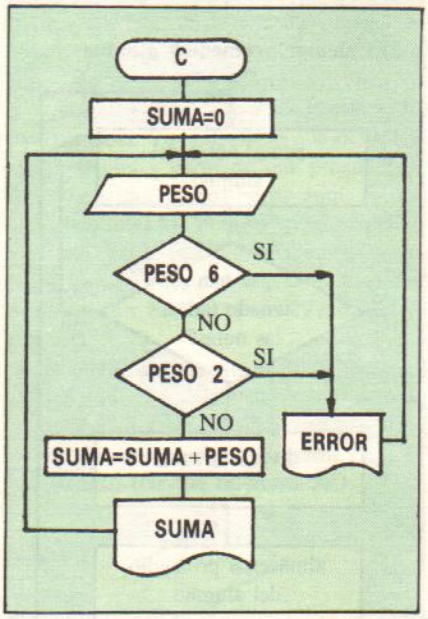
```
100 PRINT "¿HACE FRIO?"
110 INPUT X$
120 IF X$="SI" THEN GOTO 160
130 IF X$="NO" THEN GOTO 180
140 PRINT "POR FAVOR, NO CONTESTE CON EVASIVAS"
150 GOTO 100
160 PRINT "ENTONCES MEJOR ABRIGUESE Y TOME ALGO CALIENTE"
170 STOP
180 PRINT "ENTONCES LO INVITO CON UN HELADO"
190 STOP
```

(Recuérdese que, en algunas computadoras, la instrucción **GOTO** de las líneas **120** y **130** deberá suprimirse.)

Dejamos como ejercicio para el lector explicar la presencia de las líneas **140** y **150** en este listado. ¿Por qué es necesaria la instrucción **STOP** de línea **170**?

5.6.2. Un ejemplo más: Deseamos construir un programa que nos permita sumar una lista de números, a semejanza del que hicimos en 4.4., pero esta vez con ciertas restricciones.

Supongamos que, por ejemplo, queremos evitar errores de tipeo de nuestra parte, y que sabemos que los números que necesitamos sumar están, digamos, entre 2 y 6 –podría ser éste el caso de pesos de recién nacidos que vamos a promediar–. Una forma de evitar una cierta clase de errores de tipeo sería hacer que la computadora rechace, de los datos que se ingresen, aquellos que no estén comprendidos entre 2 y 6.



Es conveniente, al diagramar, *colocar las salidas de los condicionales en caso de respuesta "no" hacia abajo, en el sentido normal del flujo*. Al codificar en BASIC, bloques consecutivos del diagrama se traducirán en instrucciones consecutivas del programa, y esta disposición de las salidas reflejará mejor el hecho de que, ante respuesta negativa, la transferencia de control no se realiza sino que se pasa a la instrucción próxima según el curso normal.

Al codificar un diagrama que contenga expresiones condicionales, tratemos de mantener el orden y la claridad, economizando instrucciones todo lo posible. Una treta muy usada para estos casos es escribir la primera parte de cada instrucción condicional hasta la expresión **THEN** y dejar pendiente el número de línea a que se transfiere hasta tener completo todo el listado, y luego llenar los espacios con los números correspondientes

```
100 LET SUMA=0
110 INPUT PESO
120 IF PESO  6 THEN GOTO...
130 IF PESO  2 THEN GOTO...
140 LET SUMA=SUMA+PESO
150 PRINT SUMA
160 GOTO 110
```

Aquí, en las líneas **120-130** hemos dejado abierta la transferencia condicional a la línea donde ordenaremos imprimir el mensaje de error hasta tener todo más "armado". Ahora se ve claramente que si agregamos

```
170 PRINT "ERROR"
180 GOTO 110
```

y completamos los puntos suspensivos con el número de línea **170** obtenemos una reproducción exacta del flujo del diagrama anterior. Para comprobarlo basta hacer la prueba de escritorio correspondiente (ver 3.5. en caso de duda).

Nota: la computadora no admitirá –salvo raras excepciones– que ingresemos a la memoria de programación una línea incompleta, tal como

```
120 IF PESO  6 THEN GOTO
```

Todo lo dicho recién vale, por lo tanto, para el momento de la codificación tentativa, con papel y lápiz, de nuestro programa.

Esta y otras mañas diversas de la computadora se dominarán peleándonos con el **TEST** de la **LECCION 5**.

TEST DE LA LECCION V

Diagramar y pasar a Basic:

a) Dados tres números A, B y C, imprimir el mayor de ellos.
b) Dados diez números, imprimir el promedio.
c) Confeccionar un programa "portero de cine" que nos pregunte nuetra edad (recordar INPUT) y si nuestra respuesta consiste en un número mayor que 18, responda con el mensaje "adelante, puede pasar". En otro caso, conteste "Lo lamento, esta película es prohibida para menores de 18 años".
d) Una vez logrado el programa anterior, corregirlo para que, si contestamos con un número mayor que 100, se nos responda "Esta película es un poco fuerte, abuelo, yo no se la recomiendo."

EL OSO PEREZOSO



En este número finalizamos con el análisis de este programa que nos enviara el amigo Gustavo.

ESTRUCTURAS DEL PROGRAMA

● **Programa principal: 100/470**
Consta de presentación e inicialización de variables llamando a subprograma de los dos niveles.

● **Subprogramas:** son 15
–NIVEL 1: utiliza **PATOS, LECHUZAS, GALLOS, DISPARO, FLASH, BALIN** y **PUNTOS**
–NIVEL 2: utiliza **DISP, BALIN, PUNTOS**
–DISPARO: utlizado por NIVEL 1, a su vez usa **LECHPUN, GALLPUN, PATPUN** y **PUNTOS**
–INSTRUCCIONES

VARIABLES DEL PROGRAMA PRINCIPAL

RECO Contiene al iniciar el programa un récord conseguido, luego valdrá el máximo puntaje logrado. Para cambiarlo, modifique la línea **380**.

PRIM Llave auxiliar que sirve para determinar si es la primera vez que se supera el récord.

PALAS. Cantidad total de balas durante el juego.

EXIT. Llave de condición de fin de juego.

PUM. Contiene el puntaje conseguido.

COMENTARIO GENERAL SOBRE EL PROGRAMA

100/230 Presentación e instrucciones.

240/370 Definición de caracteres a utilizar.

380/390 Inicialización de variables.

400/420 Presentación de la pantalla.

430/470 Programa principal: tiene una bifurcación a la subrutina de **NIVEL 1** y luego **NIVEL 2**, y así sucesivamente (la condición de fin del juego está en **NIVEL 2**).

480/660 Subprograma **NIVEL 1**: es el primer nivel del juego, con 12 sprites en pantalla, revólver y balas. Utiliza otros subprogramas (para dar movimiento a los animales y carteles, etc.).

670/840 Subprograma **DISPARO**: contiene una buena alternativa de testeo de coincidencias por intervalos. NOTA: los intervalos donde se preguntan las coincidencias no contienen a los sprites, esto se debe a la velocidad de la bala, más rápida que la velocidad de ejecución del Basic Extendido. La velocidad de la bala puede ser cambiada, sin embargo aconsejo utilice la que figura en el programa, dado que podría perder exactitud al aumentarla.

850/930 Subprogramas **PATOS, LECHUZAS, GALLOS**: les da velocidad a cada uno respectivamente.

940/1060 Subprogramas **PATPUN, LECHPUN, GALLPUN**: despliega puntajes en pantalla y/o suma o resta balas al cargador.

1070/1090 Subprograma **BALIN**: encargado de la visualización en pantalla de las balas.

1100/1160 Subprograma **FLASHEO**: ilumina y apaga en pantalla cartel de **GAME OVER**.

1170/1190 Subprograma **FLASH**: ilumina y apaga carteles en pantalla.

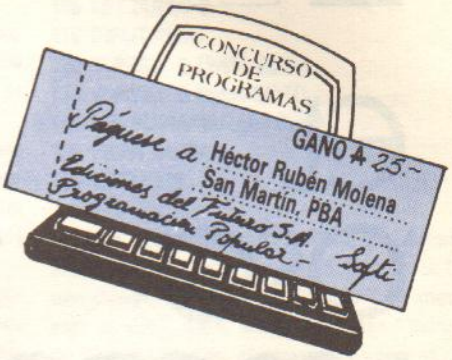
1200/1360 Subprograma **NIVEL 2**: aparece el oso en pantalla, es el segundo nivel del juego que consiste en sumar balas para enfrentar nuevamente al **NIVEL 1**.

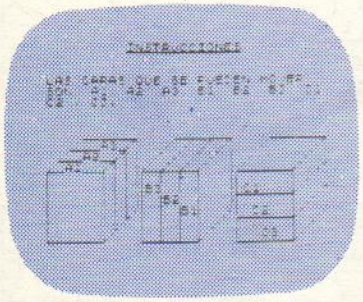
1370/1460 Subprograma **DISP**: se encarga de verificar el acierto o no de la bala sobre el oso, dándolo vuelta y haciéndolo ir más rápido cada vez.

1470/1570 Subprograma **INSTRUCCIONES**: instrucciones del juego.

1580/1600 Subprograma **PUNTOS**: flasheo de los puntos obtenidos, en pantalla.

CUBO MAGICO









Este programa del amigo **Héctor** es uno de los mejores, recibidos para nuestro **Concurso**. Muy buena la recreación de un popular juego de ingenio, excelente gratificación en Pantalla, pulida y cuidadosa programación y completa documentación técnica hacen de este programa un muy divertido y atractivo entretenimiento que servirá para atrapar en la pantalla a nuestros amigos lectores y a su familia. Pese a que es bastante extenso, recomendamos cargarlo y no se arrepentirán.

El ''leit-motiv'' del ''juego'' es reordenar los colores de las caras de un ''cubo mágico'' que presenta 9 (3 * 3) cuadrados por cara y que tiene una movilidad muy peculiar, similar al del popular juego de estos últimos años.

Tiene además todos los ingredientes para adaptarlos a nuestras necesidades (Instrucciones muy claras y fijación de hasta 4 niveles de dificultad).

A tipear entonces el programa y divertirnos con esta tortura del maldito **Cubo Mágico**...

```
  10 REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
# - C U B O - M A G I C O -  3 #
@ Hector R Molina-S.Martin.PBA @
@ Rev.Programacion Popular-N° 8 @
@ CZSpectrum-TK90X-TS2068  48K @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  100 GO SUB 9900
  140 CLS : PAPER 5: BORDER 5: IN
K 0
  150 PRINT AT 2,5;"***  CUBO  MA
GICO  ***": PRINT AT 5,9;"DESEA
VER LAS": PRINT AT 7,4;"INSTRUCC
IONES ( S / N )"
  160 IF INKEY$="S" OR INKEY$="s"
 THEN GO SUB 8000: GO TO 500
  170 IF INKEY$="N" OR INKEY$="n"
 THEN GO TO 500
  180 GO TO 150
  500 DIM A(60)
  510 FOR I= 1 TO  9: LET A(I)=1
 NEXT I
  520 FOR I=11 TO 19: LET A(I)=2
 NEXT I
  530 FOR I=21 TO 29: LET A(I)=4
 NEXT I
  540 FOR I=31 TO 39: LET A(I)=3
 NEXT I
  550 FOR I=41 TO 49: LET A(I)=7
 NEXT I
  560 FOR I=51 TO 59: LET A(I)=5
 NEXT I
  570 GO TO 600
  580 CLS : PAPER 5: BORDER 5
  600 PRINT AT 11,7;"SELECCIONAR
 NIVEL";AT 13,9;"DE DIFICULTAD";A
T 15,13;"1 / 4"
  610 INPUT ,H
  620 IF H>4 THEN GO TO 600
  650 GO SUB 7000
  940 REM
  960 REM ******************
  980 REM ***              ***
 1000 REM *** PRINT CUBO  ***
 1020 REM ***              ***
 1040 REM ******************
 1060 REM
 1090 BORDER 0: PAPER 0: CLS
 1100 PRINT INK A(2);AT 4,15;"ABC
";AT 5,14,"DEQFG";AT 6,16;"H"
 1110 PRINT INK A(9);AT 5,13;"I";
AT 6,11;"JKQLM";AT 7,12;"NOP"
 1120 PRINT INK A(3);AT 5,19;"I";
AT 6,17;"JKQLM";AT 7,18;"NOP"
 1130 PRINT INK A(8);AT 7,9;"ABC"
;AT 8,8;"DEQFG";AT 9,10;"H"
 1140 PRINT INK A(1);AT 7,15;"ABC
";AT 8,14;"DEQFG";AT 9,16;"H"
 1150 PRINT INK A(4);AT 7,21;"ABC
";AT 8,20;"DEQFG";AT 9,22;"H"
 1160 PRINT INK A(7);AT 8,13;"I";
AT 9,11;"JKQLM";AT 10,12;"NOP"
 1170 PRINT INK A(5);AT 8,19;"I";
AT 9,17;"JKQLM";AT 10,18;"NOP"
 1180 PRINT INK A(6);AT 10,15;"AB
C";AT 11,14;"DEQFG";AT 12,16;"H"
 1190 PRINT INK A(12);AT 9,8;"LC
";AT 10,8;"QQ";AT 11,8;"NE"
 1200 PRINT INK A(13);AT 10,11;"C
";AT 11,11;"QL";AT 12,11;"EQ";AT
 13,12;"N"
 1210 PRINT INK A(14);AT 12,14;"L
C";AT 13,14;"QQ";AT 14,14;"NE"
 1220 PRINT INK A(19);AT 12,8;"LC
";AT 13,8;"QQ";AT 14,8;"NE"
 1230 PRINT INK A(11);AT 13,11;"C
";AT 14,11;"QL";AT 15,11;"EQ";AT
 15,12;"N"
 1240 PRINT INK A(15);AT 15,14;"L
C";AT 16,14;"QQ";AT 17,14;"NE"
 1250 PRINT INK A(18);AT 15,8;"LC
";AT 16,8;"QQ";AT 17,8;"NE"
 1260 PRINT INK A(17);AT 16,11;"C
";AT 17,11;"QL";AT 18,11;"EQ";AT
 19,12;"N"
 1270 PRINT INK A(16);AT 18,14;"L
C";AT 19,14;"QQ";AT 20,14;"NE"
 1280 PRINT INK A(22);AT 13,17;"A
K";AT 13,17;"QQ";AT 14,17;"FP"
 1290 PRINT INK A(23);AT 10,21;"A
";AT 11,20;"KQ";AT 12,20;"QF";AT
 13,20;"P"
 1300 PRINT INK A(24);AT 9,23;"AK
";AT 10,23;"QQ";AT 11,23;"FP"
 1310 PRINT INK A(29);AT 15,17;"A
K";AT 16,17;"QQ";AT 17,17;"FP"
 1320 PRINT INK A(21);AT 13,21;"A
";AT 14,20;"KQ";AT 15,20;"QF";AT
 16,20;"P"
 1330 PRINT INK A(25);AT 12,23;"A
K";AT 13,23;"QQ";AT 14,23;"FP"
 1340 PRINT INK A(28);AT 18,17;"A
K";AT 19,17;"QQ";AT 20,17;"FP"
 1350 PRINT INK A(27);AT 16,21;"A
";AT 17,20;"KQ";AT 18,20;"QF";AT
 19,20;"P"
 1360 PRINT INK A(26);AT 15,23;"A
K";AT 16,23;"QQ";AT 17,23;"FP"
 1640 REM
 1660 REM ******************
 1680 REM ***              ***
 1700 REM *** MOVIMIENTOS ***
 1720 REM ***              ***
 1740 REM ******************
 1760 REM
 2100 INPUT ; FLASH 1;"** HAGA UN
 MOVIMIENTO **";A$
 2210 IF A$="A1D" OR A$="a1d" THE
N GO SUB 2510: GO TO 1100;
 2220 IF A$="A1I" OR A$="a1i" THE
N GO SUB 2520: GO TO 1100
 2230 IF A$="A2D" OR A$="a2d" THE
N GO SUB 2530: GO TO 1100
 2240 IF A$="A2I" OR A$="a2i" THE
N GO SUB 2540: GO TO 1100
 2250 IF A$="A3D" OR A$="a3d" THE
N GO SUB 2550: GO TO 1100
 2260 IF A$="A3I" OR A$="a3i" THE
N GO SUB 2560: GO TO 1100
 2270 IF A$="B1D" OR A$="b1d" THE
N GO SUB 2570: GO TO 1100
 2280 IF A$="B1I" OR A$="b1i" THE
N GO SUB 2580: GO TO 1100
 2290 IF A$="B2D" OR A$="b2d" THE
N GO SUB 2590: GO TO 1100
 2300 IF A$="B2I" OR A$="b2i" THE
N GO SUB 2600: GO TO 1100
 2310 IF A$="B3D" OR A$="b3d" THE
N GO SUB 2610: GO TO 1100
 2320 IF A$="B3I" OR A$="b3i" THE
N GO SUB 2620: GO TO 1100
 2330 IF A$="C1D" OR A$="c1d" THE
```

```
N GO SUB 2630: GO TO 1100
2340 IF A$="C1I" OR A$="c1i" THE
N GO SUB 2640: GO TO 1100
2350 IF A$="C2D" OR A$="c2d" THE
N GO SUB 2650: GO TO 1100
2360 IF A$="C2I" OR A$="c2i" THE
N GO SUB 2660: GO TO 1100
2370 IF A$="C3D" OR A$="c3d" THE
N GO SUB 2670: GO TO 1100
2380 IF A$="C3I" OR A$="c3i" THE
N GO SUB 2680: GO TO 1100
2390 IF A$="I" OR A$="i" THEN GO
 SUB 6000: GO TO 1000
2400 IF A$="XXX" OR A$="xxx" THE
N GO TO 580
2410 GO TO 2100
2440 REM
2460 REM ************************
2480 REM ***                  ***
2500 REM *** S/R MOVIMIENTOS ***
2502 REM ***                  ***
2504 REM ************************
2506 REM
2510 LET A(10)=A(6): LET A(20)=A
(7): LET A(30)=A(8): LET A(6)=A(
44): LET A(7)=A(45): LET A(8)=A(
46): LET A(44)=A(54): LET A(45)=
A(55): LET A(46)=A(56): LET A(54
)=A(28): LET A(55)=A(29): LET A(
56)=A(22): LET A(28)=A(10): LET
 A(29)=A(20): LET A(22)=A(30): LE
T A(40)=A(14): LET A(50)=A(13):
LET A(14)=A(12): LET A(13)=A(19)
: LET A(12)=A(18): LET A(19)=A(1
7): LET A(18)=A(16): LET A(17)=A
(15): LET A(16)=A(40): LET A(15)
=A(50): RETURN
2520 LET A(10)=A(8): LET A(20)=A
(7): LET A(30)=A(6): LET A(8)=A(
22): LET A(7)=A(29): LET A(6)=A(
28): LET A(22)=A(56): LET A(29)=
A(55): LET A(28)=A(54): LET A(56
)=A(46): LET A(55)=A(45): LET A(
54)=A(44): LET A(46)=A(10): LET
 A(45)=A(20): LET A(44)=A(30): LE
T A(40)=A(12): LET A(50)=A(13):
LET A(12)=A(14): LET A(13)=A(15)
: LET A(14)=A(16): LET A(15)=A(1
7): LET A(16)=A(18): LET A(17)=A
(19): LET A(18)=A(40): LET A(19)
=A(50): RETURN
2530 LET A(10)=A(5): LET A(20)=A
(1): LET A(30)=A(9): LET A(5)=A(
43): LET A(1)=A(41): LET A(9)=A(
47): LET A(43)=A(53): LET A(41)=
A(51): LET A(47)=A(57): LET A(53
)=A(27): LET A(51)=A(21): LET A(
57)=A(23): LET A(27)=A(10): LET
A(21)=A(20): LET A(23)=A(30): RE
TURN
2540 LET A(10)=A(9): LET A(20)=A
(1): LET A(30)=A(5): LET A(9)=A(
23): LET A(1)=A(21): LET A(5)=A(
27): LET A(23)=A(57): LET A(21)=
A(51): LET A(27)=A(53): LET A(57
)=A(47): LET A(51)=A(41): LET A(
53)=A(43): LET A(47)=A(10): LET
A(41)=A(20): LET A(43)=A(30): RE
TURN
2550 LET A(10)=A(4): LET A(20)=A
(3): LET A(30)=A(2): LET A(4)=A(
42): LET A(3)=A(49): LET A(2)=A(
48): LET A(42)=A(52): LET A(49)=
A(59): LET A(48)=A(58): LET A(52
)=A(26): LET A(59)=A(25): LET A(
58)=A(24): LET A(26)=A(10): LET
A(25)=A(20): LET A(24)=A(30): LE
T A(40)=A(32): LET A(50)=A(33)
LET A(32)=A(34): LET A(33)=A(35)
: LET A(34)=A(36): LET A(35)=A(3
7): LET A(36)=A(38): LET A(37)=A
(39): LET A(38)=A(40): LET A(39)
=A(50): RETURN
2560 LET A(10)=A(2): LET A(20)=A
(3): LET A(30)=A(4): LET A(2)=A(
24): LET A(3)=A(25): LET A(4)=A(
26): LET A(24)=A(58): LET A(25)=
A(59): LET A(26)=A(52): LET A(58
)=A(48): LET A(59)=A(49): LET A(
52)=A(42): LET A(48)=A(10): LET
A(49)=A(20): LET A(42)=A(30): LE
T A(40)=A(34): LET A(50)=A(33)
LET A(34)=A(32): LET A(33)=A(39)
: LET A(32)=A(38): LET A(39)=A(3
7): LET A(38)=A(36): LET A(37)=A
(35): LET A(36)=A(40): LET A(35)
=A(50): RETURN
2570 LET A(10)=A(4): LET A(20)=A
(5): LET A(30)=A(6): LET A(4)=A(
14): LET A(5)=A(15): LET A(6)=A(
16): LET A(14)=A(56): LET A(15)=
A(57): LET A(16)=A(58): LET A(56
)=A(38): LET A(57)=A(39): LET A(
58)=A(32): LET A(38)=A(10): LET
A(39)=A(20): LET A(32)=A(30): LE
T A(40)=A(24): LET A(50)=A(23)
LET A(24)=A(22): LET A(23)=A(29)
: LET A(22)=A(28): LET A(29)=A(2
7): LET A(25)=A(26): LET A(27)=A
(25): LET A(26)=A(40): LET A(25)
=A(50): RETURN
2580 LET A(10)=A(6): LET A(20)=A
(5): LET A(30)=A(4): LET A(6)=A(
32): LET A(5)=A(39): LET A(4)=A(
38): LET A(32)=A(58): LET A(39)=
A(57): LET A(38)=A(56): LET A(58
)=A(16): LET A(57)=A(15): LET A(
56)=A(14): LET A(16)=A(10): LET
A(15)=A(20): LET A(14)=A(30): LE
T A(40)=A(22): LET A(50)=A(23)
LET A(22)=A(24): LET A(23)=A(25)
: LET A(24)=A(26): LET A(25)=A(2
7): LET A(26)=A(28): LET A(27)=A
(29): LET A(28)=A(40): LET A(29)
=A(50): RETURN
2590 LET A(10)=A(3): LET A(20)=A
(1): LET A(30)=A(7): LET A(3)=A(
13): LET A(1)=A(11): LET A(7)=A(
17): LET A(13)=A(55): LET A(11)=
A(51): LET A(17)=A(59): LET A(55
)=A(37): LET A(51)=A(31): LET A(
59)=A(33): LET A(37)=A(10): LET
A(31)=A(20): LET A(33)=A(30): RE
TURN
2600 LET A(10)=A(7): LET A(20)=A
(1): LET A(30)=A(3): LET A(7)=A(
33): LET A(1)=A(31): LET A(3)=A(
37): LET A(33)=A(59): LET A(31)=
A(51): LET A(37)=A(55): LET A(59
)=A(17): LET A(51)=A(11): LET A(
55)=A(13): LET A(17)=A(10): LET
A(11)=A(20): LET A(13)=A(30): RE
TURN
2610 LET A(10)=A(2): LET A(20)=A
(9): LET A(30)=A(8): LET A(2)=A(
12): LET A(9)=A(19): LET A(8)=A(
18): LET A(12)=A(54): LET A(19)=
A(53): LET A(18)=A(52): LET A(54
)=A(36): LET A(53)=A(35): LET A(
52)=A(34): LET A(36)=A(10): LET
A(35)=A(20): LET A(34)=A(30): LE
T A(40)=A(42): LET A(50)=A(43)
LET A(42)=A(44): LET A(43)=A(45)
: LET A(44)=A(46): LET A(45)=A(4
7): LET A(46)=A(48): LET A(47)=A
(49): LET A(48)=A(40): LET A(49)
=A(50): RETURN
2620 LET A(10)=A(8): LET A(20)=A
(9): LET A(30)=A(2): LET A(8)=A(
34): LET A(9)=A(35): LET A(2)=A(
36): LET A(34)=A(52): LET A(35)=
A(53): LET A(36)=A(54): LET A(52
)=A(18): LET A(53)=A(19): LET A(
54)=A(12): LET A(18)=A(10): LET
A(19)=A(20): LET A(12)=A(30): LE
```


Ahora versión casete!!!

DEC ELECTRONICA & COMPUTACION

PRESENTA EL

lapiz optico

PRIMERO EN ARGENTINA

HECHO POR ARGENTINOS

DISTRIBUYEN:

PAPELSHOP

Av. Pueyrredon 658 CAP.

CP: 1032 TE: 88-8582

JCN S.a

LORIA 373 L. de Zamora

CP: 1832 TE: 243-4933



ref. 18 precio promocion: A 70.-

```
T A(40)=A(44): LET A(50)=A(43):
LET A(44)=A(42): LET A(43)=A(49)
: LET A(42)=A(48): LET A(49)=A(4
7): LET A(48)=A(46): LET A(47)=A
(45): LET A(46)=A(40): LET A(45)
=A(50): RETURN
2630 LET A(10)=A(12): LET A(20)=
A(13): LET A(30)=A(14): LET A(12
)=A(22): LET A(13)=A(23): LET A(
14)=A(24): LET A(22)=A(32): LET
A(23)=A(33): LET A(24)=A(34): LE
T A(32)=A(42): LET A(33)=A(43):
LET A(34)=A(44): LET A(42)=A(10)
: LET A(43)=A(20): LET A(44)=A(3
0): LET A(40)=A(2): LET A(50)=A(
9): LET A(2)=A(8): LET A(9)=A(7)
: LET A(8)=A(6): LET A(7)=A(5):
LET A(6)=A(4): LET A(5)=A(3): LE
T A(4)=A(40): LET A(3)=A(50): RE
TURN
2640 LET A(10)=A(14): LET A(20)=
A(13): LET A(30)=A(12): LET A(14
)=A(44): LET A(13)=A(43): LET A(
12)=A(42): LET A(44)=A(34): LET
A(43)=A(33): LET A(42)=A(32): LE
T A(34)=A(24): LET A(33)=A(23):
LET A(32)=A(22): LET A(24)=A(10)
: LET A(23)=A(20): LET A(22)=A(3
0): LET A(40)=A(6): LET A(50)=A(
7): LET A(6)=A(8): LET A(7)=A(9)
: LET A(8)=A(2): LET A(9)=A(3):
LET A(2)=A(4): LET A(3)=A(5): LE
T A(4)=A(40): LET A(5)=A(50): RE
TURN
2650 LET A(10)=A(19): LET A(20)=
A(11): LET A(30)=A(15): LET A(19
)=A(29): LET A(11)=A(21): LET A(
15)=A(25): LET A(29)=A(39): LET
A(21)=A(31): LET A(25)=A(35): LE
T A(39)=A(49): LET A(31)=A(41):
LET A(35)=A(45): LET A(49)=A(10)
: LET A(41)=A(20): LET A(45)=A(3
0): RETURN
2660 LET A(10)=A(15): LET A(20)=
A(11): LET A(30)=A(19): LET A(15
)=A(45): LET A(11)=A(41): LET A(
19)=A(49): LET A(45)=A(35): LET
A(41)=A(31): LET A(49)=A(39): LE
T A(35)=A(25): LET A(31)=A(21):
LET A(39)=A(29): LET A(25)=A(10)
: LET A(21)=A(20): LET A(29)=A(3
0): RETURN
2670 LET A(10)=A(18): LET A(20)=
A(17): LET A(30)=A(16): LET A(18
)=A(28): LET A(17)=A(27): LET A(
16)=A(26): LET A(28)=A(38): LET
A(27)=A(37): LET A(26)=A(36): LE
T A(38)=A(48): LET A(37)=A(47):
LET A(36)=A(46): LET A(48)=A(10)
: LET A(47)=A(20): LET A(46)=A(3
0): LET A(40)=A(52): LET A(50)=A
(53): LET A(52)=A(54): LET A(53)
=A(55): LET A(54)=A(56): LET A(5
5)=A(57): LET A(56)=A(58): LET A
(57)=A(59): LET A(58)=A(40): LET
A(59)=A(50): RETURN
2680 LET A(10)=A(16): LET A(20)=
A(17): LET A(30)=A(18): LET A(16
)=A(46): LET A(17)=A(47): LET A(
18)=A(48): LET A(46)=A(36): LET
A(47)=A(37): LET A(48)=A(38): LE
T A(36)=A(26): LET A(37)=A(27):
LET A(38)=A(28): LET A(26)=A(10)
: LET A(27)=A(20): LET A(28)=A(3
0): LET A(40)=A(54): LET A(50)=A
(53): LET A(54)=A(52): LET A(53)
=A(59): LET A(52)=A(58): LET A(5
9)=A(57): LET A(58)=A(56): LET A
(57)=A(55): LET A(56)=A(40): LET
A(55)=A(50): RETURN
6994 REM
6996 REM **********************
6998 REM ***                 ***
7000 REM *** S/R MEZCLADO    ***
7002 REM ***                 ***
7004 REM **********************
7006 REM
7100 CLS : PRINT AT 10,9; FLASH
1;"I MEZCLANDO I";AT 9,9;"+-----
------+";AT 11,9;"+-----------+"
7150 LET S=H*5
7200 FOR I=1 TO S
7210 LET B=1+INT (RND*18)
7220 IF B= 1 THEN GO SUB 2510
7230 IF B= 2 THEN GO SUB 2520
7240 IF B= 3 THEN GO SUB 2530
7250 IF B= 4 THEN GO SUB 2540
7260 IF B= 5 THEN GO SUB 2550
7270 IF B= 6 THEN GO SUB 2560
7280 IF B= 7 THEN GO SUB 2570
7290 IF B= 8 THEN GO SUB 2580
7300 IF B= 9 THEN GO SUB 2590
7310 IF B=10 THEN GO SUB 2600
7320 IF B=11 THEN GO SUB 2610
7330 IF B=12 THEN GO SUB 2620
7340 IF B=13 THEN GO SUB 2630
7350 IF B=14 THEN GO SUB 2640
7360 IF B=15 THEN GO SUB 2650
7370 IF B=16 THEN GO SUB 2660
7380 IF B=17 THEN GO SUB 2670
7390 IF B=18 THEN GO SUB 2680
7400 NEXT I
7500 RETURN
7994 REM
7996 REM **********************
7998 REM **                  **
8000 REM ** S/R INSTRUCCIONES **
8002 REM **                  **
8004 REM **********************
8006 REM
8020 BORDER 6: PAPER 6: INK 0
8030 CLS : PRINT AT 0,9;"INSTRUC
CIONES";AT 0,9; OVER 1;"_______
_____"
8040 PRINT AT 3,0;"LAS CARAS QUE
 SE PUEDEN MOVER   SON: A1, A2,
A3, B1, B2, B3, C1,C2 Y C3."
8050 PLOT 0,80: DRAW 51,0: DRAW
0,-51: DRAW -51,0: DRAW 0,51
8060 PLOT 0,80: DRAW 30,24: DRAW
 51,0: DRAW 0,-51: DRAW -30,-24
8070 PLOT 50,80: DRAW 30,24
8080 PLOT 10,88: DRAW 51,0: DRAW
 0,-51
8090 PLOT 20,96: DRAW 51,0: DRAW
 0,-51
8100 PRINT AT 11,2;"A1";AT 10,4;
"A2";AT 9,6;"A3"
8250 PLOT 85,80: DRAW 51,0: DRAW
 0,-51: DRAW -51,0: DRAW 0,51
8260 PLOT 85,80: DRAW 30,24: DRA
W 51,0: DRAW 0,-51: DRAW -30,-24
8270 PLOT 135,80: DRAW 30,24
8280 PLOT 102,29: DRAW 0,51: DRA
W 30,24
8290 PLOT 119,29: DRAW 0,51: DRA
W 30,24
8300 PRINT AT 13,11;"B3";AT 14,1
3;"B2";AT 15,15;"B1"
8450 PLOT 170,80: DRAW 51,0: DRA
W 0,-51: DRAW -51,0: DRAW 0,51
8460 PLOT 170,80: DRAW 30,24: DR
AW 51,0: DRAW 0,-51: DRAW -30,-2
4
8470 PLOT 220,80: DRAW 30,24
8480 PLOT 170,63: DRAW 51,0: DRA
W 30,24
8490 PLOT 170,46: DRAW 51,0: DRA
W 30,24
8500 PRINT AT 13,22;"C1";AT 15,2
3;"C2";AT 17,24;"C3"
8550 PAUSE 300
8600 PRINT AT 20,0; FLASH 1;"  O
PRIMA CUALQUIER TECLA PARA     S
EGUIR                         "
IF INKEY$="" THEN GO TO 8600
8610 CLS
8620 PRINT AT 1,0;"CADA CARA SE
PUEDE MOVER:";AT 4,5;"* A LA DER
ECHA_____(D)";AT 6,5;"* A LA IZQ
UIERDA____(I)";AT 8,0;"POR EJEMPL
O: A1D, A1I, B3I, C2D"
8630 PAUSE 300
8640 PRINT AT 12,0;"SE HACE UN S
OLO MOVIMIENTO POR  VEZ (PULSAND
O ENTER)."
8650 PAUSE 150
8660 PRINT AT 16,0;"EN CUALQUIER
 MOMENTO DURANTE EL JUEGO, PULSA
NDO I (ENTER), SE   VUELVE A LAS
 INSTRUCCIONES."
8670 PAUSE 200
8680 PRINT AT 20,0; FLASH 1;"  O
PRIMA CUALQUIER TECLA PARA     S
EGUIR                         "
IF INKEY$="" THEN GO TO 8680
8690 CLS
8700 PRINT AT 2,0;"SI QUIERE VER
 EL CUBO TOTALMENTEORDENADO, SEL
ECCIONE EL NIVEL   DE DIFICULTAD
 0."
8710 PAUSE 200
8720 PRINT AT 7,0;"EN CUALQUIER
MOMENTO,DURANTE EL JUEGO, PUEDE
MEZCLAR EL CUBO    INTRODUCIENDO
 EL MOVIMIENTO XXX."
8730 PAUSE 200
8780 PRINT AT 20,0; FLASH 1;"  O
PRIMA CUALQUIER TECLA PARA     S
EGUIR                         "
IF INKEY$="" THEN GO TO 8780
8790 CLS
8800 PRINT AT 10,0;"QUIERE VER L
AS INSTRUCCIONES";AT 12,0;"NUEVA
MENTE  (S / N)"
8810 IF INKEY$="S" OR INKEY$="s"
 THEN GO TO 8000
8820 IF INKEY$="N" OR INKEY$="n"
 THEN GO TO 8890
8830 GO TO 8800
8890 CLS
8900 RETURN
9894 REM
9896 REM **********************
9898 REM ***                 ***
9900 REM *** S/R GRAFICOS    ***
9902 REM ***                 ***
9904 REM **********************
9906 REM
9910 CLS : BORDER 5: PAPER 5: IN
K 0: PRINT AT 10,9; FLASH 1;"I U
N MOMENTO I";AT 9,9;"+----------
---+";AT 11,9;"+-----------+"
9920 RESTORE 9960
9930 FOR M=0 TO 135
9940 READ A: POKE USR "A"+M,A
9950 NEXT M
9960 DATA 0,0,0,0,1,7,31,127
9962 DATA 0,0,24,126,255,255,255
,255
9964 DATA 0,0,0,0,128,224,248,25
4
9966 DATA 1,7,7,1,0,0,0,0
9968 DATA 255,255,255,255,127,31
,7,1
9970 DATA 255,255,255,255,254,24
8,224,128
9972 DATA 128,224,224,128,0,0,0,
0
9974 DATA 126,24,0,0,0,0,0,0
9976 DATA 0,0,0,0,0,0,24,126
9978 DATA 0,0,0,0,1,7,7,1
9980 DATA 1,7,31,127,255,255,255
,255
9982 DATA 128,224,248,254,255,25
5,255,255
9984 DATA 0,0,0,0,128,224,224,12
8
9986 DATA 127,31,7,1,0,0,0,0
9988 DATA 255,255,255,255,126,24
,0,0
9990 DATA 254,248,224,128,0,0,0,
0
9992 DATA 255,255,255,255,255,25
5,255,255
9994 RETURN
 999 SAVE "CUBO" LINE 100
```

NOTAS GRAFICAS

Lineas 1100/1360: tipear en modo GRAPHICS las letras entre comillas

ESTRUCTURA DEL PROGRAMA

140 / 180 Presentación
500 / 560 Definición inicial de cada cara
1090 / 1360 Impresión del cubo en Pantalla
2100 / 2410 Determinación del movimiento a realizar
2510 / 2680 Rotación de las caras (subrutinas)
7100 / 7500 Subrutina de "mezclado"
3020 / 8900 Subrutina de instrucciones
9910 / 9994 Subrutina de definición de gráficos

TABLA DE VARIABLES

A( ) Guarda un valor numérico representantivo del color de cada una de las partes en que se dividen las caras
I Variable auxiliar para los procesos iterativos
H Guarda el valor solicitado como nivel de dificultad
A$ Variable alfanumérica que recepciona el código del movimiento a realizar
B Variable auxiliar usada para producir el mezclado del cubo

Por último, algunas aclaraciones que nos dice **Héctor**:

1. Las líneas **2510** a **2680** son muy largas y lentas para corregir. Por ello podremos subdividirlas en líneas más cortas, colocando el **RETURN** sólo en la última de las nuevas líneas

2. Una vez cargado, hacer **GO TO 9999** para tener autoejecución al recuperar lo grabado.

Felicitaciones, **Héctor** !!! ...

GALACTIC MOVER

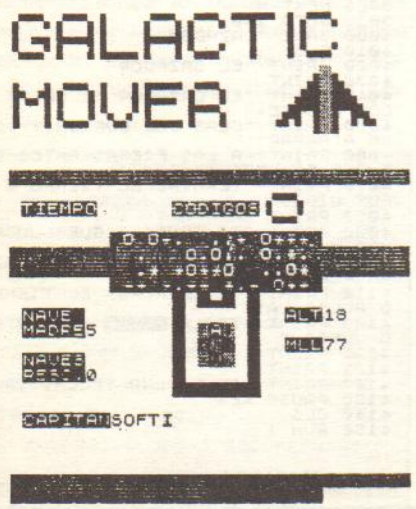



```
  1 REM
 GALACTIC MOVER
 GERARDO MATEOS-PARANA,ER
REV PROGRAMACION POPULAR-N°7
CZ1000/1500-TK83/85-ZX81 6K8

  2 REM JUEGOS
  6 DIM A$(60,100)
  7 LET ZX=0
  8 LET M=5
  9 LET NV=0
 10 LET X=1
 11 LET Y=1
 12 DIM T(1)
 14 GOSUB 800
 15 PRINT AT 0,1;"
                  ";AT 1,0;"
                    ";AT 2,0;"
                    ";AT 3,1;"
                    "
 20 PRINT AT 5,0;"
                  ";AT 6,0;"
                    ";AT 7,0;"
                    "
 25 PRINT AT 4,24;"  ";AT 5,23
;"  ";AT 6,22;"  ";AT 7,
21;"  ";AT 8,24;"  "
 30 PRINT AT 18,0;"
G.J.M.                  ";AT 17,0;"
        ARG.PAR.1985
 35 PRINT AT 21,0;"                DI
GITE ENTER                ";AT 21,0;"
        DIGITE ENTER
 40 IF INKEY$="" THEN GOTO 35
 45 PRINT AT 21,0;"        SIMULAC
RO DE DESPEGUE          "
 46 FOR N=1 TO 50
 47 NEXT N
 50 FOR A=1 TO 35
 55 PRINT AT 9,0;A$(A,1 TO 31);
 56 POKE 16424,10
 57 PRINT A$(A+1,1 TO 31);
 58 POKE 16424,10
 59 PRINT A$(A+2,1 TO 31);
 60 POKE 16424,10
 61 PRINT A$(A+3,1 TO 31);
 62 POKE 16424,10
 63 PRINT A$(A+4,1 TO 31);
 64 POKE 16424,10
 65 PRINT A$(A+5,1 TO 37);
 66 POKE 16424,10
 67 NEXT A
 70 CLS
 75 GOSUB 500
 80 PRINT AT 20,0;"
                    ";AT 21,0;"
                    ";AT 0,0
                    "
 85 PRINT AT 2,1;"TIEMPO       C
ODIGOS  ";AT 1,21;"  ";AT 3,21
;"  "
 90 PRINT AT 4,8;"
                    ";AT 5,0;"
                    ";AT 6,0;"
                    ";AT 7,8
;"                   "

 95 PRINT AT 8,13;"  "
100 PRINT AT 9,1;"NAVE
    ";AT 10,1;"MADRES";AT
10,13;"  ";AT 11,13;"
    ";AT 12,1;"NAVES
PERD ";AT 13,1;"PERD.0";AT 13,13
;"  ";AT 14,13;"
    "
105 PRINT AT 16,1;"CAPITAN";Z$
110 LET R=INT (RND*60)+1
115 LET S=INT (RND*100)+1
120 PRINT AT 9,25;R;AT 11,25;S
121 FOR N=200 TO 0 STEP -1
125 PRINT AT 4,9;A$(Y,X TO X+14
)
130 PRINT AT 5,9;A$(Y+1,X TO X+
14)
135 PRINT AT 6,9;A$(Y+2,X TO X+
14)
140 PRINT AT 7,9;A$(Y+3,X TO X+
14)
145 IF A$(Y+3,X+7)="" OR A$(Y+
3,X+7)="" THEN GOSUB 1000
150 LET Y=Y+(1 AND INKEY$="7")-
(1 AND INKEY$="6")
155 IF Y<6 THEN LET Y=Y+1
160 IF Y>56 THEN LET Y=Y-1
165 LET X=X+(1 AND INKEY$="5")-
(1 AND INKEY$="8")
170 IF X<2 THEN LET X=X+1
175 IF X+15>99 THEN LET X=X-1
180 IF N<10 THEN PRINT AT 2,7;N
185 IF X=S OR Y=R THEN GOTO 120
0
190 NEXT N
195 FOR N=16 TO 21
200 PRINT AT N,0;"
                                "
205 NEXT N
210 PRINT AT 17,0;"CAPITAN ";Z$
;AT 18,0;"MENSAJE :SE ACABO TU TI
EMPO..... ";"SOLO HAS GANADO U$S
";T(1)
215 PRINT AT 21,0;"DESEAS JUGAR
OTRA VEZ? (S-N)"
220 IF INKEY$="N" THEN STOP
225 IF INKEY$="S" THEN GOTO 235
230 GOTO 220
235 GOTO 1846
499 STOP
500 PRINT TAB 9;"GALACTIC MOVER
"
505 PRINT AT 2,2;"TU EMPRESA DE
MUDANZAS ESPACIAL"
510 PRINT
515 PRINT "LES ESTA AL BORDE DE
LA QUIEBRA."
520 PRINT
525 PRINT TAB 2;"UNA TERRIBLE G
UERRA AMENAZA A "
530 PRINT
535 PRINT "LA TIERRA Y FUISTE C
ONTRATADO    "
540 PRINT
545 PRINT "POR LA COMISION INTE
RGALACTICA   "
550 PRINT
555 PRINT "PARA TRAER A REFUGIO
A LAS REMA "
560 PRINT
565 PRINT "NENTES COLONIAS QUE'
ESTAN VARA  "
570 PRINT
575 PRINT "DAS EN EL ESPACIO."
580 PRINT
585 PRINT TAB 2;"GANARAS JUGOSA
S RECOMPENSAS"
590 PRINT
595 PRINT ."POR CADA NAVE QUE RE
SCATES,"
600 PRINT AT 21,0;"                DI
GITE ENTER                "
605 IF INKEY$="" THEN GOTO 600
610 CLS
615 PRINT TAB 9;"INSTRUCCIONES"
620 PRINT AT 2,2;"TE MUEVES CON
LAS FLECHAS="
625 PRINT
630 PRINT "      "
635 PRINT
640 PRINT TAB 2;"VISUALIZARAS E
L ESPACIO"
645 PRINT
650 PRINT "EN UNA PANTALLA."
655 PRINT
660 PRINT TAB 2;"TEN CUIDADO CO
N LAS ZONAS "
665 PRINT
670 PRINT "NEGRAS."
675 PRINT
680 PRINT TAB 2;"DEBES CUIDAR T
U TIEMPO Y TUS "
685 PRINT
690 PRINT "NAVES(5)."
695 PRINT
700 PRINT TAB 2;"LAS NAVES A RE
SCATAR SON="
705 PRINT
710 PRINT "S=U$S 4000","M=U$S 6
000"
715 PRINT AT 21,0;"INGRESA TU N
OMBRE="
720 INPUT Z$
721 IF LEN Z$>=8 THEN GOTO 720
729 PRINT AT 21,0;" * * * * *BU
ENA SUERTE* * * * * ";AT 21,0;"
* * * * *BUENA SUERTE* * * * *"
730 IF INKEY$="" THEN GOTO 725
735 CLS
740 RETURN
799 STOP
800 FAST
805 FOR A=1 TO 60
810 FOR B=1 TO 100
815 LET J=INT (RND*4)+1
820 IF J=1 THEN LET A$(A,B)="."
825 IF J=2 THEN LET A$(A,B)="+"
830 IF J=3 THEN LET A$(A,B)="*"
835 IF J=4 THEN LET A$(A,B)="0"
840 NEXT B
845 NEXT A
849 IF NV=15 THEN GOTO 890
850 FOR C=0 TO 15-ZX
855 LET J=INT (RND*60)+1
```

Continúa pág. 23



Compra tu **Commodore 16** ó **64** por el plan de ahorro previo de **Drean** en 20 cuotas mensuales de ~~A~~ 13.72 por sorteo o licitación.

Drean commodore

Llámanos a **TODO PLAN S.A.**
Te enviaremos un vendedor.

T.E. 313-1415/1515/1615 Int. 612.

ref. 21

EL CAZADOR



Este **"Cazador"** de **Raúl**, nos muestra una variante del "comecocos" o "Pacman" para las versiones "menores" de la línea **Sinclair**, en el cual el "cazador" debe apurarse a atrapar a la totalidad de sus "presas" en el tiempo disponible, para no perder una vida (tiene 3), avanzando entre las "calles" de un laberinto y perdiendo también si choca.



Para jugar, sólo tendremos que cargar el programa, hacer una copia de seguridad (**GO TO 4000**) y de inmediato veremos, las instrucciones en pantalla y podremos empezar la cacería...

TABLA DE VARIABLES

- **A** Bucle de impresión del "laberinto"
- **A$** Variable de respuestas a continuar o no el juego
- **B$** Cadena de borrado de avisos
- **BB** Variable que lee la pantalla para ver si ya está ocupada por una presa o un muro (evita la sobreimpresión)
- **C/D** Coordenadas del cazador
- **F** Bucle de impresión de las presas
- **K** Contador de puntos
- **L/M** Coordenadas aleatorias de impresión de cada presa
- **N** Bucle de pausa
- **P** Variable que controla si el cazador toma una presa o choca contra un muro
- **T** Cantidad a presas a imprimir
- **V** Vidas
- **W** Tiempo de juego
- **Y$/Z$** Cadenas de gráficos de pantalla
- **Z** Presas capturadas

El amigo **Raúl** nos mandó una muy completa documentación del programa (además de otras inquietudes, trucos y consultas que aún no hemos podido analizar. **Raúl**, gracias...), de las que destacamos éstas:

1. La línea **196** regula el tiempo de juego (**W**) y puede variarse a voluntad cambiando sus valores.

2. El programa no tiene una presentación especial pues prefirió que las líneas a teclear (pensando en el lector) se ocupen del juego en sí. Incluso pueden eliminarse las líneas **4010** a **4140** (instrucciones elementales).

3. **Raúl** dice que llegó a acumular 1.160 puntos (sin ninguna vida).

4. Para usar una **TK-85** con Joystick, adaptar las líneas **242** a **245** al Joystick (6,7,8,9) y las líneas **311** a **312** como sigue (para controlar la respuesta):

```
311 Let A$=INKEY$
312 IF A$=" "THEN GO TO 311
313 IF A$="0" THEN RUN 1
314 STOP
315 RUN 1
```

```
  1 REM
 EL CAZADOR
 RAUL LAMBERTUCCI-PINAMAR,PBA
 REV.PROGRAMACION POPULAR-Nº 8
 CZ1000/1500-TK83/85-ZX81 +2K
  10 LET K=0
  20 LET Z=0
  30 LET V=3
  35 LET B$="
  40 LET T=10
  50 CLS
  55 PRINT "      JUEGO DEL CAZ
ADOR      "
  56 PRINT "VIDAS= ";V;,"PUNTOS=
";K
  60 LET Z$="
  "
  65 LET Y$="
  "
  90 PRINT AT 2,0;
  95 FOR A=1 TO 9
 100 PRINT Z$+Y$
 105 NEXT A
 110 PRINT Z$
 145 FOR F=1 TO T
 150 LET M=INT (RND*16)+4
 160 LET L=INT (RND*20)+8
 170 PRINT AT M,L;
 175 LET BB=PEEK (PEEK 16396+256
*PEEK 16399)
 178 IF BB=23 OR BB=0 THEN GOTO
150
 180 PRINT "*"
 185 NEXT F
 191 PRINT AT 21,0;"PRESIONA UNA
TECLA PARA JUGAR  "
 193 PAUSE 4E4
 196 LET W=90+(T*3)
 200 LET C=10
 210 LET D=10
 230 LET W=W-1
 241 PRINT AT D,C;"
 242 LET C=C+(INKEY$="8")-(INKEY
$="5")
 245 LET D=D+(INKEY$="6")-(INKEY
$="7")
 248 IF C<0 THEN GOTO 1000
 250 PRINT AT D,C;
 260 LET P=PEEK (PEEK 16398+256+
PEEK 16399)
 270 IF P=23 THEN GOTO 2000
 275 IF P<=63 THEN GOTO 1000
 280 PRINT "0"
 300 IF W<>0 THEN GOTO 230
 305 PRINT AT 21,0;B$
 310 PRINT AT 21,0;"TERMINO TU T
IEMPO. OTRO ?"
 311 INPUT A$
 312 IF A$="S" OR A$="SI" OR A$=
"" THEN RUN 1
 313 STOP
 314 RUN 1
1050 PRINT AT 21,0;"CHOCASTE,TE
QUEDAN ";V-1;" VIDAS"
1051 PAUSE 120
1052 PRINT AT 21,0;B$
1053 LET V=V-1
1054 PRINT AT 1,7;V;"  "
1060 IF V=0 THEN GOTO 1100
1070 GOTO 200
1110 PRINT AT 21,0;"LO LAMENTO..
."
1120 PAUSE 90
1130 GOTO 305
2000 LET Z=Z+1
2001 LET K=K+1
2004 PRINT AT 1,23;K*10
2010 IF Z=T THEN GOTO 3000
2030 GOTO 240
3000 PRINT AT 21,0;"GANASTE Y SI
GUES JUGANDO
3010 LET Z=0
3020 LET T=T+5
3024 PRINT AT D,C;"
3025 FOR N=1 TO 200
3026 NEXT N
3030 GOTO 140
4000 SAVE "CAZADOR"
4010 CLS
4020 PRINT "EL CAZADOR"
4030 PRINT
4040 PRINT "EL CAZADOR SE MUEVE
CON LAS TE-"
4050 PRINT "CLAS DEL CURSOR,Y DE
BE ATRAPAR"
4060 PRINT "A LAS FIERAS ANTES D
E QUE SE LE"
4070 PRINT "TERMINE EL TIEMPO O
SUS VIDAS"
4080 PRINT
4090 PRINT "SI GANAS SIGUES JUGA
NDO Y HABRA"
4100 PRINT "MAS ANIMALES QUE CAZ
AR,PERO SI"
4110 PRINT "SE TE ACABA EL TIEMP
O PULSA LA"
4120 PRINT "TECLA ENTER PARA OTR
O JUEGO."
4121 PRINT
4122 PRINT
4127 PRINT "PULSA UNA TECLA..."
4130 PAUSE 4E4
4140 CLS
4150 RUN 1
```

ESTRUCTURA DEL PROGRAMA

10/40 Inicializa variables
50/110 Presenta la pantalla
145/185 Imprime las **"prensas"** a cazar
191/185 Espera que se pulse una tecla para iniciar el juego
196/230 Inicializa el contador de tiempo y las coordenadas iniciales del cazador.
240/300 **Bucle principal**: controla el movimiento del cazador y si se ha capturado una presa o chocado contra las paredes del laberinto
305/314 Si terminó el tiempo, da opción a otro juego, caso negativo **STOP**. Usando **CONT** el juego vuelve a comenzar
1050/1070 Subrutina de **choque**. Avisa el choque y descuenta 1 vida
1110/1130 Subrutina de **pérdida de la última vida**
2000/2030 Subrutina de **captura de presas**: cuenta las presas tomadas y otorga el puntaje
3000/3030 Subrutina de **fin de juego** por captura de todas las presas: inicializa el contador, aumenta la cantidad de presas a imprimir, borra el cazador y reinicia
4000/4150 **Auto RUN** e **Instrucciones** elementales

Viene de pág. 21

```
  856 IF J<=15 OR J>=65 THEN GOTO
  855
  860 LET K=INT (RND*100)+1
  861 IF K<=9 OR K>=65 THEN GOTO
860
  865 LET H=INT (RND*2)+1
  870 IF H=1 THEN LET A$(J,K)="O"
  875 IF H=2 THEN LET A$(J,K)="X"
  880 NEXT C
  885 SLOW
  890 RETURN
 1000 IF A$(Y+3,X+7)="O" THEN LET
  T(1)=T(1)+4000
 1010 IF A$(Y+3,X+7)="X" THEN LET
  T(1)=T(1)+8000
 1020 LET A$(Y+3,X+7)="■"
 1025 PRINT AT 21,26;T(1)
 1030 LET NV=NV+1
 1031 PRINT AT 13,6;NV
 1032 IF NV>=15 THEN GOTO 1800
 1035 RETURN
 1200 PRINT AT 2,21;"■■"
 1205 LET M=M-1
 1210 PRINT AT 9,6;M
 1215 IF M=0 THEN GOTO 1900
 1216 PRINT AT 2,21;"  "
 1220 GOTO 125
 1800 FOR N=16 TO 21
 1805 PRINT AT N,0;"
                    "
 1810 NEXT N
 1820 PRINT AT 17,0;"MENSAJE: HAS
  LOGRADO LA META... ";"TU RECOMP
ENSA SERA DE U$S";T(1)
 1825 PRINT "FELICITACIONES..."
 1830 PRINT AT 21,0;"DESEAS JUGAR
 OTRA VEZ? (S/N)"
 1835 IF INKEY$="N" THEN STOP
 1840 IF INKEY$="S" THEN GOTO 185
0
 1845 GOTO 1835
 1846 LET ZX=NV
 1850 LET NV=0
 1855 LET T(1)=0
 1860 LET M=5
 1861 LET X=1
 1862 LET Y=1
 1865 GOSUB 850
 1866 CLS
 1870 PRINT AT 21,24;"/
 1875 GOTO 80
 1900 FOR N=16 TO 21
 1905 PRINT AT N,0;"
                    "
 1910 NEXT N
 1915 PRINT AT 17,0;"MENSAJE: TU
MISION HA FRACASADO.";"NO TIENES
 MAS NAVES EN TU EMPRE ";"SA.HAN
 CAIDO EN LAS ZONAS NEGRAS"
 1920 PRINT AT 21,0;"QUERES JUGAR
 OTRA VEZ? (S/N)"
 1930 IF INKEY$="N" THEN STOP
 1935 IF INKEY$="S" THEN GOTO 194
5
 1940 GOTO 1930
 1945 GOTO 1846
 2000 CLEAR
 2005 SAVE "MOVER"
 2010 RUN
```

Este programa ganador de **Gerardo** consiste en encontrar en el espacio colonias de naves perdidas. Para ello el jugador cuenta con un control remoto (pantalla) para buscar las naves.

Visualizará el espacio en una pequeña pantalla y deberá llevar las naves hasta un punto medio, delimitado por un casillero en blanco.

El jugador dispone de 5 naves para cumplir con su misión, dentro de un tiempo máximo (el descuento del tiempo no aparece en pantalla, pero el programa nos avisará 10 segundos antes de finalizar el juego). Debemos sortear los obstáculos (agujeros negros espaciales) que se nos interpongan en nuestro camino de rescate. La presentación e instrucciones las veremos en pantalla.

```
GALACTIC MOVER

 TU EMPRESA DE MUDANZAS ESPACIA
LES ESTA AL BORDE DE LA QUIEBRA.
 UNA TERRIBLE GUERRA AMENAZA A
LA TIERRA Y FUISTE CONTRATADO
POR LA COMISION INTERGALACTICA
PARA TRAER A REFUGIO A LAS REMA
NENTES COLONIAS QUE ESTAN VARA
DAS EN EL ESPACIO.
 GANARAS JUGOSAS RECOMPENSAS
POR CADA NAVE QUE RESCATES.
          DIGITE ENTER
```

```
INSTRUCCIONES

 TE MUEVES CON LAS FLECHAS=
      5   8   7   6
 VISUALIZARAS EL ESPACIO
EN UNA PANTALLA.
 TEN CUIDADO CON LAS ZONAS
NEGRAS.
 DEBES CUIDAR TU TIEMPO Y TUS
NAVES(5).
 LAS NAVES A RESCATAR SON=
O=U$5 4000      X=U$5 8000
 * * * * *BUENA SUERTE* * * * *
```

Ganaremos el juego si rescatamos las 15 colonias varadas en el espacio antes del tiempo máximo. Atención, que **no es nada fácil!!!**...

Una vez cargado y peusto a correr el programa, éste entra en un **FAST** que dura alrededor de **4 minutos** (a no desesperarse), debido al cargado de la tabla de las líneas **800** a **890**. Por último, debemos aclarar que las "flechas" funcionan al revés de lo normal.

Suerte!!! (la necesitarán)...

CASETE VIRGEN
para COMPUTACION

- Fabricación propia
- Utilizamos cinta Ampex Ferrocobalto (USA)
- Las medidas se preparan en el día

Producciones ECCOSOUND S.A.

Tronador 611 - (1027) Capital
551-9489 553-5080/5063

Ofrecemos calidad y precio
al servicio de la informática.

- Consúltenos o haga su pedido.

ref. 12

MOTOCICLETA

Sos un motociclista que compites con otro en la pantalla de tu *C-64* para ver quién encierra a quién!!!...

Este es un divertido juego para dos jugadores, que piloteando una hipotética moto cada uno deben intentar encerrar al otro y, a su vez, no dejarse encerrar por el contrario.

Las motos van dejando tras de sí unas **"huellas"** que son fatales si te tocan. Tampoco puede volverse hacia atrás ni tocar los bordes de la pantalla.

Por supuesto, el primero que **"choca"** pierde y el otro sumará 1 punto en su score.

Para jugar debemos tipear el programa, luego **SAVE**'arlo, y hacer **RUN**.

Para mover la "moto" usaremos las siguientes **teclas**.

	Jugador 1	Jugador 2
Arriba	W	O
Abajo	X	M
Izquierda	A	J
Derecha	D	L

Y ahora, a divertirnos!!!...

TABLA DE VARIABLES

A$ Variable teclado para movimiento motos

CO Variable **POKE** (**1024**: dibuja línea moto)

DM Indicador de movimiento horizontal de la moto del jugador 2
- 1 = derecha
- 0 = queda donde está
- -1 = izquierda

DN Idem para el movimiento vertical del jugador 1
- 1 = sube
- 0 = queda donde está
- -1 = baja

DX Idem **DM** para el jugador 1

DY Idem **DN** para el jugador 1

I/J Variables **FOR/NEXT**

J$ Variable teclado para comienzo del juego

P1 Posición moto jugador 1 (para **POKE**'s)

P2 Idem para el jugador 2

S1 Puntaje del jugador 1

S2 Puntaje del jugador 2

SC Variable **POKE** (**55296**: detecta colisión)

X1 Posición coordenada horizontal del jugador 1

X2 Idem para el jugador 2

Y1 Posición coordenada vertical del jugador 1

Y2 Idem para el jugador 2

ESTRUCTURA DEL PROGRAMA

15/57 Inicialización y presentación en pantalla

70/135 Examina el teclado para detectar movimientos

155/171 Calcula el movimiento de la **"moto"**. Analiza si se choca con la pantalla

172/173 Control de choques

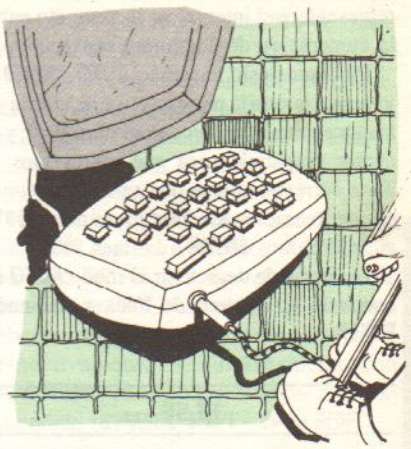
175/188 Dibuja la moto y recicla

1000/1020 Fin del juego

```
1 REM ************************************
2 REM *            MOTOCICLETA           *
3 REM * REV.PROGAMAMACION POPULAR N*8    *
4 REM *          COMMODORE  64           *
5 REM ************************************
15 POKE53281,0:POKE53280,4
20 SC=1024:CO=55296
25 PRINTCHR$(147)
30 X1=2:Y1=12:X2=37:Y2=12
35 DX=1:DM=-1:DN=0:DY=0
40 PRINTCHR$(19)CHR$(158)"JUGADOR 1:"S1
50 PRINTCHR$(19)TAB(27)"JUGADOR 2:"S2
52 FORI=1TO8:PRINTCHR$(17);:NEXT
54 PRINTTAB(14)"PULSE UNA TECLA"
55 GETJ$:IFJ$<>""THEN55
56 GETA$:IFA$=""THEN56
57 PRINTTAB(14)CHR$(145)"
70 GETA$
80 IFA$="W"THENDY=-1:DX=0
90 IFA$="X"THENDY=1:DX=0
95 IFA$="A"THENDX=-1:DY=0
100 IFA$="D"THENDX=1:DY=0
110 IFA$="O"THENDN=-1:DM=0
120 IFA$="M"THENDN=1:DM=0
130 IFA$="J"THENDM=-1:DN=0
135 IFA$="L"THENDM=1:DN=0
155 Y1=Y1+DY
156 IFY1<1ORY1>24THENF=0:GOSUB1000:GOTO25
157 X1=X1+DX
162 IFX1<0ORX1>39THENF=0:GOSUB1000:GOTO25
163 Y2=Y2+DN
165 IFY2<1ORY2>24THENF=1:GOSUB1000:GOTO25
167 X2=X2+DM
169 IFX2<0ORX2>39THENF=1:GOSUB1000:GOTO25
170 P1=X1+40*Y1
171 P2=X2+40*Y2
172 IFPEEK(SC+P1)=160THENF=0:GOSUB1000:GO
TO25
173 IFPEEK(SC+P2)=160THENF=1:GOSUB1000:GO
TO25
175 POKESC+P1,160
177 POKECO+P1,14
179 POKESC+P2,160
181 POKECO+P2,5
183 GOTO70
1000 REM
1010 IFF=1THENS1=S1+1:GOSUB2000:RETURN
1020 S2=S2+1:GOSUB2000:RETURN
2000 REM
2010 FORJ=1TO10
2020 FORI=0TO15
2030 POKE53281,I
2040 NEXTI,J
2050 POKE53281,0
2060 RETURN

READY.
```

MEGA-BASIC



Si bien el Basic de la **CZ Spectrum** y de la **TK90X** son bastante poderosos, **Softi** nos brinda aquí la posibilidad de nuevos comandos muy, pero muy importantes...

Estas máquinas vienen equipadas "de fábrica" con un Sistema Operativo que incluye entre sus diversas funciones un Intérprete Basic. Este Intérprete Basic compone parte del Hardware de nuestra computadora, residiendo en la ROM, y está dotado de una serie de "instrucciones' o "comandos", tal cual se encuentran especificados en el respectivo manual del equipo.

Ahora bien, tal como veremos seguidamente, es perfectamente factible (para un avezado programador, por supuesto) *modificar desde "afuera" de la máquina (es decir, por Software), estos comandos o instrucciones*, ya sea reformulándolos o bien agregándole otras funciones no previstas originalmente.

Por último, antes de entrar en detalles, destacamos que no resulta posible la aplicación de este importante programa a la **TS 2068**, dado que el programa utiliza e interactúa frecuentemente con rutinas incorporadas originariamente en la ROM de la **Spectrum**, las que están direccionadas y/o difieren respecto de la **TS2068**. Sin embargo no descartamos que algún experto programador amigo nos haga llegar su colaboración al respecto...

En este caso, el programa en **lenguaje de máquina** que presentamos **agrega nada menos que 7 (siete) nuevos comandos/funciones** a nuestra **Spectrum/TK90X** que son de suma importancia, a saber:

1. **AUTO**: *permite la numeración automática (no manual) de las nuevas líneas de programa.*
2. **ELSE**: *potencia la construcción* **IF/THEN** *permitiendo la bifurcación controlada cuando* **NO** *se cumple la condición* **IF**.
3. **MODE**: *agrega la posibilidad de trabajar a razón de 64 caracteres por línea*, duplicando así la capacidad de la pantalla e impresora.
4. **ON ERR**: *posibilita programar el tratamiento de los errores de operación, impidiendo así la interrupción del proceso.*
5. **ON**: *Posibilita los "saltos condicionados", según el valor de una variable dada.*
6. **RENUM**: *permite la renumeración automática de líneas del programa* (incluso **GOTO**'s y **GOSUB**'s, entre otras instrucciones).
7. **SWAP**: *realiza la transferencia simultánea de los contenidos de dos variables entre sí*, sin requerir una variable intermedia.

CARGA DEL PROGRAMA

Si bien el programa es bastante extenso, contamos para auxiliarnos en esta difícil pero muy importante y necesaria tarea, con la ayuda de un programa **monitor** que desarrollamos en este mismo número de nuestra revista **Programación Popular**: **Cargando programas en Lenguaje de Máquina (I)**. Debemos en consecuencia seguir las instrucciones allí explicadas y tipear con sumo cuidado el listado de este programa (**MEGA-BASIC**), para luego grabarlo de inmediato.

Recién entonces podremos hacer uso del mismo y asombrarnos de su potencialidad.

AUTO

Este comando lo utilizaremos cuando estemos ingresando un nuevo programa o parte de éste, eliminándonos la molestia de tener que numerar "manualmente" las nuevas líneas de programa incorporadas, las que se encargará de numerar automáticamente nuestra rutina.

Sólo deberemos hacer **AUTO l,i**, siendo l el número de línea a partir de la cual queremos

SINTESIS DE OPERACION

Comando	Teclas a usar (modo extendido)	Significado / variables
AUTO 1,i	CAPS SHIFT + E	1 = Número de línea a partir de la cual se hará la numeración i = Intervalo entre líneas
ELSE i	CAPS SHIFT + Q	Se agrega a continuación de **IF...THEN...** (es **opcional**) i = Instrucción/es a ejecutar si **NO** se cumple la condición **IF...**, o sea cuando **NO** se ejecutan las sentencias **THEN...**
MODE c	CAPS SHIFT + Y	c = 132 caracteres/línea c = 1 64 caracteres/línea
ON ERR e1, e2,...	CAPS SHIFT + W	Se intercala inmediatamente antes de la línea de programa a la que se desea ceder el control cuando se produzca alguno de los códigos de error especificados e = Código/s de error
ON x GOTO/GOSUB 11, 12,...	CAPS SHIFT + T	x = 1 GOTO/GOSUB 11 x = 2GOTO/GOSUB 12 etc.
RENUM l,i,p,u,	CAPS SHIFT + R	l = Número de la primera línea ya renumerada p = Primera línea del programa a renumerar i = Intervalo entre líneas requerido u = Ultima línea del programa a renumerar **p/u son opcionales**
SWAP x,y	CAPS SHIFT + U	Equivale a LET w=x: LET x=y: LET y=w

empezar a numerar en modo automático, e **i** representará el intervalo o incremento entre la numeración de dos líneas consecutivas.

Probemos esto haciendo **AUTO 1000,10** y veremos que en la parte inferior izquierda de nuestra pantalla aparece el número **1000**, y el cursor preparado para que ingresemos la línea correspondiente que será automáticamente reemplazada por el número **1010** cuando demos **ENTER** a la línea **1000**.

La forma de desactivar el modo **AUTO** es borrando el número de línea y haciendo **ENTER**.

ELSE

Esta instrucción, de **utilización optativa**, complementa la construcción **IF... THEN...**, agregándole la posibilidad **ELSE...**, cuando **NO** se cumple la condición prevista en **IF**, o sea, cuando **NO** se ejecutan las sentencias inmediatamente posteriores al **THEN**. En otras palabras, el **MEGA BASIC**, en este caso transfiere el control a los comandos inmediatos al **ELSE** cuando **no** se cumple la condición **IF**, en lugar de transferirlo a la línea del programa siguiente (tal como lo hace en la construcción **IF/THEN** básica).

Veamos un ejemplo:

```
  10 REM comprobaremos que la
probabilidad de salida de un nu
mero aleatorio entre 2 alternati
vas (0 y 1 en este ejemplo) tien
de a igualarse cuantas mas veces
repitamos la experiencia (50%)
  20 LET x0=0
  30 LET x1=0
  40 LET x=INT (2*RND)
  50 IF x=0 THEN LET x0=x0+1 ELS
E LET x1=x1+1
  60 PRINT AT 10,13;100*x0/(x0+x
1);AT 12,13;100*x1/(x0+x1)
  70 GO TO 40
```

Un caso especial se dá cuando existan en una misma línea varias sentencias **IF** y varios **ELSE**, correspondiendo cada **ELSE** al **IF** más cercano, salvo cuando existan entre ambas otras construcciones **ELSE**, conformándose, en consecuencia, una serie de estructuras **IF/THEN/ELSE** ''anidadas'' unas dentro de otras.

MODE

Este nuevo comando nos brinda la posibilidad de **seleccionar la cantidad de caracteres por línea** que podremos utilizar en la pantalla del TV o en la impresora.

Así, si hacemos **MODE 1**(y **ENTER**), veremos que todo lo que escribamos aparece en la pantalla mucho más ''finito'' y que en lugar de 32 caracteres por línea dispondremos ahora del doble de capacidad, o sea, nada menos que 64 caracteres/línea.

En este Modo de impresión ampliado a 64 caracteres, podremos utilizar todas las facilidades de las instruccionee **PRINT** y **LPRINT**, incluso los caracteres de control **8** y **9**, para desplazar el cursor una posición hacia la izquierda y derecha, respectivamente. Existe una diferencia con el uso de las **''comas''** como separadores, dado que ahora la pantalla queda dividida en cuatro zonas de 16 caracteres cada una. Así, mediante la ''coma'' trasladamos el *Print Position* al inicio del siguiente ''cuarto'' de la pantalla, en lugar de a la siguiente ''mitad'' como en el Modo normal.

Para retornar al Modo normal sólo deberemos tipear **MODE 0** (y **ENTER**), volviendo entonces el formato de presentación al que estábamos acostumbrados.

ON ERR

Tenemos ahora en la **CZSpectrum** y **TK90X** la posibilidad de programar el manejo de los errores de procesamiento, capacidad de la que carecía totalmente (en la **TS2068** se encuentra incorporado en el Basic de la ROM una instrucción equivalente.)

Spectrum

TK 90X

```
----------------------------------
   1 DATA "MEGA-BASIC",60159,653
15,5157,61254
   2 DATA "01EBF5E52A785C2322785
C7CB52003FD3440C5D5CD19EBC34D00C
D8E02C02100",58134
   3 DATA "5CCB7E200723352B20023
6FF7D21045CBD20EECD1E03D02100SCB
ECA1003EB21",50304
   4 DATA "045CBECA1003CB7E2004E
8CB7EC85F7723605233A095C7723FD4
E07FD5601E5",52994
   5 DATA "CD65EBC306037BFE3ADAA
4EB0DFA4F032803C64FC921EB0104280
D210502F578",47573
   6 DATA "FE192803214 9E8F1C34A0
37EDCDA5C1A7BECEDBFAEAAABDDDEDF7
F181B7C1C1D",69407
   7 DATA "DB19D91ED7FE30D80DFA3
D03C2890321540 2CB68CA4A03FE38D28
203E60732 0D",53839
   8 DATA "5C3E1004C83CC90000000
0000000000202020200020000050500
0000000000",10207
   9 DATA "5070505070500000207040
701070200040401020401010002050 2
060B0F00000",31609
  10 DATA "2040000000000000404040
4040200004020202040000000502
07020500000",18250
  11 DATA "0020207020200000000000
0002020400000000700000000000000
00060500000",10699
  12 DATA "1010202040400000705050
5050700000206020202070000020501
0204070000",23324
  13 DATA "6010601010600000103050
7070100000704060101060000030406
0505020000",23917
  14 DATA "7010202040400000705020
5050700000205050301060000000002
0000200000",15550
  15 DATA "0020000020204000001020
4020100000000070007000000000402
0102040000",13103
  16 DATA "2050102000200000F0F0D
0A0807000002050507050500000650506
0505060000",34976
  17 DATA "2050404050200000650505
0505050000070406040407000007040 7
04040400000",30161
  18 DATA "2050407050200000050507
0505050000070202020207000001010 1
0505020000",20946
  19 DATA "5050606050500000404040
4040407000005070707050500000705050
05050500000",32083
  20 DATA "2050505050200000605050
4040400007050505070701000070505
0606050000",33348
  21 DATA "3040201010600000702020
2020200000505050505070000050505
0505020000",25765
  22 DATA "5070707070200000505020
2050500000505050202020000070102
0204070000",22806
  23 DATA "7040404040700000004050
2030100000701010101070000020702
0202020000",17607
  24 DATA "0000000000F00020504
0F040F0000006010705070000040405
0505060000",32280
  25 DATA "0030404040300000101030
5050303000000205060403000003040 6
04040400000",22121
  26 DATA "0035050503010600004040 5
0505050000200602020700000100010
1010502000",18362
  27 DATA "4050606050500000404040
4040403000000050707070500000006050
5050500000",27531
  28 DATA "0020505050200000000605
0506040400000305050301010000030 4
04040400000",21404
  29 DATA "0030402010600000202702
0202010000000505050507000000005050
6050200000",22349
  30 DATA "0050707070200000005050
0205050000000505050301060000070 1
0204070000",23534
  31 DATA "3020402020300000202020
2020200000602010202060000050A00
0000000060",18287
  32 DATA "80B0A0B0809060F0909090
9090900F0303030300000000C0C0C0C0
00000000F0",49952
  33 DATA "F0F0F0000000000000000
0303030303030303030303030C0C0C0C
0303030F0",44481
  34 DATA "F0F0F0303030300000000
0C0C0C0C0303030C0C0C0C0C0C0C0C0
0C0C0C0C0F0",82690
  35 DATA "F0F0F0C0C0C0C000000000
0F0F0F0F030303030F0F0F0F0C0C0C0C0
0F0F0F0F0F0",98243
  36 DATA "F0F0F0F0F0F0F03EEAED4
7ED5EED7B3D5CF1C335FCAF32F8F0FD3
63102CDF0F1",86393
  37 DATA "CDB0163AF8F0A728193EF
FCD01162AF1F0ED5BF3F019CB7420082
2F1F0444DCD",70577
  38 DATA "1B1A3E00CD0116CD73F3C
D89F4FDCB007E2012FDCB306628772A5
95CCDA711FD",65620
  39 DATA "3600FF18DD2A595C225D5
CCD75F478B1C2F9F0DFFE0D289DFDCB3
046C4B7F1CD",85119
  40 DATA "6E0DCDDAF13E19FD954FF
D7752FDCB01FEFD3600FFFD360A01E52
A8A5C22B05C",64825
  41 DATA "E1CD52F53A3A5C3C28254
FC52A535C1E19CD51F1EBE3EB3816C5C
D67F4CD90F9",78106
  42 DATA "C1CAACF0EBE3EBCD68F1E
BE3EB30EAC176FDCB01AEFDCB304EC4D
C0EFD7E003C",78810
  43 DATA "F5210000FD7437FD74262
2055C2101002216 5CCDB016FDCB37AEC
D6E0DCDDAF1",65531
  44 DATA "FDCB02EEF147FE0A3802C
607CDEF153E20D778119113CD0A0CAF1
13615CD0A0C",44980
  45 DATA "ED4B455CCD1B1A3E3AD7F
D4E0D0600CD1B1ACD9710FD7E003CC48
6F0FD360AFF",63799
  46 DATA "FDCB019EC381EFFE09280
4FE152003FD340D0103001170SC21445
CFDCB0A7E28",45070
  47 DATA "0109EDB8C93E10010000C
31FF02A5D5CE5D5CDB01621EDF0225D5
CCD1F1CD17B",63289
  48 DATA "FE15200BAFDBFEE61FFE1
F20F73E15CD86F0CD282DCDFF2AE1225
D5CD13E0192",65350
  49 DATA "3E445C21E3EFE5FD3600F
FC33FF5455522A00000A0045525200E
D43495C2A5D",44461
  50 DATA "5CEB21A4F0E52A615C37E
D52E56069CD8E192006CDB819CDE8190
1793DB02828",60670
  51 DATA "C0303030328ED5B535CD
5CD5516E122535CC1C5132A615C2B2BE
DB82A495CEB",58697
  52 DATA "C1702B712B732B72F1C35
8EF7EFE3A2818237EE6C037C046234EE
D43425C234E",56630
  53 DATA "2346E5D9444DE11600C5C
D72F1C1D0606918DE225D5C0E0015C8C
D67F4BB2004",58952
  54 DATA "A7C9237ECDB618225D5CF
E2220010DFE3A281AFECB2816FE19201
6CB412012CD",50160
  55 DATA "67F4D5CD90F9D12A5D5C2
B18C6CB4128C7FE0D20CD1537C921000
0227D5CFDCB",57866
  56 DATA "3086CDDAF13EFECD0116C
D4D0D0618CD440E2A515C11CDFCC3CF0
```

Con esta instrucción evitaremos en consecuencia las "paradas" por error, controlando así según lo hayamos definido en nuestro programa, el **punto de "reinicio" del proceso cuando se produzca un determinado tipo de error**.

¿Cómo se utiliza? Muy fácil: poniendo este comando **ON ERR**, seguido de los códigos de error que hayamos previsto tratar, separados por "comas" (ver los Códigos de error en el Manual de nuestra máquina) e inmediatamente antes de la instrucción a la que deseamos ceder el control al producirse algunos de dichos códigos de error. La variable *"err"* contendrá el código del último error producido.

No hay limitaciones en cuanto al número de veces que puede utilizarse la instrucción **ON ERR** en un programa, ni tampoco en lo referente a la cantidad de errores a prever en cada comando **ON ERR** de que se trate.

Veamos un ejemplo de este poderoso comando:

```
10 PRINT AT 10,0;"Intenta dete
nerme con BREAK !!.."
20 ON ERR L CONTINUE
30 GO TO 10
```

Hay una gran variedad de aplicaciones posibles de este comando, que sugerimos utilizar con sumo cuidado, en especial cuando se trate de una serie de errores combinados. A practicar, entonces...

ON

El nuevo comando **ON** nos brinda una útil herramienta para efectuar saltos "condicionales", o sea, sujetos al valor de la variable de que se trate.

Así, si tecleamos, por ejemplo:

ON variable GO TO línea A, línea B,..., etc.

De esta forma, si

variable = 1... **GO TO línea A**

variable = 2... GO TO línea B

En otras palabras, si el valor de la expresión que sigue al ON es 1, se producirá un salto a la primera línea después del **GO TO/GOSUB**, si la expresión vale 2, el salto se producirá a la segunda línea, y así sucesivamente.

Veamos un ejemplo sencillo:

```
10 INPUT "Ingresa 2 numeros cu
alquiera  x=";x;"  y=";y
20 INPUT "Elige operador  1=su
mar 2=restar3=multiplicar  4=div
idir  ?";operador
30 ON operador GO SUB 100,200,
300,400
40 GO TO 10
100 PRINT "x;" + ";y;" = ";x+y:
RETURN
200 PRINT "x;" - ";y;" = ";x-y:
RETURN
300 PRINT "x;" * ";y;" = ";x*y:
RETURN
400 PRINT "x;" / ";y;" = ";x/y:
RETURN
```

En caso que el valor de la expresión que sigue al **ON** supere a la cantidad de líneas indicadas después del **GO TO**, el salto se producirá a la siguiente línea del programa. Las ventajas con respecto al **GO TO** que conocíamos son bastante obvias, principalmente con respecto a su comprensión (legibilidad) y simple modificación.

Hay un pequeño inconveniente: cuando querramos ingresar el **GOTO** o el **GOSUB** de una línea **ON** veremos que el cursor no se encuentra en **"K"**. Para obviar el problema debemos presionar las mismas teclas (**H** y **G**, respectivamente) simultáneamente con **CAPS SHIFT** y pasando previamente al Modo Extendido (**SYMBOL SHIFT** + **CAPS SHIFT**.)

RENUM

Es éste un comando que seguramente será sumamente apreciado por nuestros amigos lectores de **Programación Popular**, y al cual venimos aproximándonos desde los primeros números de nuestra revista; ¿se acuerdan???

Pues bien, ahora podemos decir que contamos con una valiosa herramienta parae renumerar, en forma totalmente automática, nuestros "desprolijos" programas en Basic y, ahora sí, **incluyendo la reasignació automática de GOTO's y GOSUB's**!!!

```
D3EFDCD0116",57616
  57 DATA "2A515C11CDFCA77323722
31109F3C3A70DED733F5CFD360210CD8
7F1C3A017FD",69794
  58 DATA "CB01C6D5EBFDCB309623
B5CCB96FDCB378E2802CBD62A5F5CA7E
D52200523E3F",57707
  59 DATA "CDEFF2CDB6F2EB7ECDB51
823FE0D2806EBCDD37F218E0D1C9CD882
C3FD28C19FE",74311
  60 DATA "2028F9FE2C28F5FE1AC33
E19AF3210FDCD4D0DFDCB029EFDCB02A
E2A8A5CE52A",66968
  61 DATA "3D5CE5219FF2E5ED733D5
C2A8525CE537CD9511EBCDFEF1EBCDB6F
22A8A5CE3EB",68477
  62 DATA "CD4D0D3A8B5C923811200
678FD9650300923E20D5CDCDFCD118E93
E20D7C37C11",64246
  63 DATA "1600FD5EFE21901ACDB50
3FD3600FFED5B8A5C1CC37E112A5B5CA
7ED52C03A41",61360
  64 DATA "5CCB072804C6431821213
B5CCB9E1B1A13FE2C2802FE193E4B280
4CB56280BCB",44126
  65 DATA "DE3CFDCB305E28023E43D
5CDEFF2D1C9D921915C56D53604CDCDF
CE1FD7457D9",82680
  66 DATA "C9CDB7F1CD6E0DC3DAF1F
DCB0025EC44BF2A7FDCB016EC83A085CF
DCB01AEF5FD",81066
  67 DATA "CB026EC403F3F1FE18304
7FE103030FE06300A47E6014F781FC61
2181F200921",39169
  68 DATA "6A5C3E08AE77180EFE0ED
8D60D21415CBE7720023600FDCB02DEEB
FC9FD71D311",65152
  69 DATA "63F318063A0D5C1109F32
A4F5C23237323723723 7C92A3D5CE52146F
4E5ED733D5C",58221
  70 DATA "CDD415F51600FD5EFF21C
800CDB503F1217FF3E5FE0A285EFE0B2
83DFE082814",57984
  71 DATA "FE182816FE092821FE0CC
24C0FCD04F4EBC3E519CD04F4C3111103
E3ACD810F3E",58374
  72 DATA "06CD810F3E18C3810F2A5
B5C7EFE0DC8545D23CD32F4C31110FDC
B376EC02A49",60802
  73 DATA "5CCDE619EBCD9516214A5
CCD1C19CDF0F13600C30116FDCB376EC
20110214950",53238
  74 DATA "CD0F1918E92A5B5C37CD9
511ED521923C1D8C5444D626B231ACD3
2F4E6F8FE10",68826
  75 DATA "2009231AD617CE0020012
3A7ED4209EB38E3C9FE3AC07EFE08200
A237EFE1620",57268
  76 DATA "032318012B1AC9FDCB306
6CA2510FD3600FF1600FD5EFE21901AC
DB503C377F3",66561
  77 DATA "2A5D5C7ECD6CF4D0CD740
018F7FE203FC8FE18C37F002A595C2B2
25D5CCD67F4",62534
  78 DATA "C3031ADFB9C28A1C18DEF
DCB01BECD75F4AF32475C3D323A5C180
3CD67F4CDBF",66805
  79 DATA "16FD340DFA8A1CCD5FF40
600FE0DCA7BF5FE3A28E6213FF5E54FC
D67F479FECE",81466
  80 DATA "3011D6183804FE083804C
38A1C7E21F1F5180F21CCF54FCDDC163
8F17921481A",57718
  81 DATA "D6CE4F094E0918032A745
C7E2322745C01E7F4C54FFE20D282F4F
E03200DDFCD",66922
  82 DATA "DE1CBFC1CCB6F5EBC3161
CA728F4FE0520030518EEFE022008C1C
D561CCDB6F5",65703
  83 DATA "C9FE07C26318FDCB017EF
DCB0286C44D0DF13A745CD613CDFC21C
DB6F5C3AD1C",75184
  84 DATA "CD541FD27B1BFDCB0A7E2
0712A425CCB7C2814218FEFF22455C2A6
15C2BED5B59",54581
  85 DATA "5C1B3A445C1833CD8E193
A445C2819A72043477EE6C078280FCFF
FC1CD3025C8",62146
  86 DATA "2A555C3EC0A6C0AFFE01C
E005623SEED53455C35E2356EB19232
2555CEB225D",50938
  87 DATA "5C571E00FD360AFF15FD7
20DCA9AF414CD72F12808CF16CD3025C
0C1C1DFFE0D",71424
  88 DATA "28BAFE3ACA9AF479A728F
9CF0BF708FB09FD0AEA0BFA0CE60DEB0
EF50FEE10E9",71271
  89 DATA "11F012E313E414FC15E01
6E117D818CE1900615D57A6062646A1
21416181A1E",39758
  90 DATA "2027292B2D2F313336383
A3D0365FE0000F30370F6057AF506CB0
58DF60035FC",56297
  91 DATA "043D06CC0605F2F6050BF
7051DF7054DF60567F805CEF80596F80
905A8F80507",57803
  92 DATA "F70563F80905BDF805E0F
8080040F9080586FA05BCF9053AFA05C
4F9060062F9",62992
  93 DATA "0581F900C8FCCD671E010
000CD451E1803CD991E78B12004ED4BB
25CC5ED5B4B",58429
  94 DATA "5C2A595C2BCDE519CD00F
3C3C61EC1CD3025280AEF0238EBCDE93
4DAA3F6CD5F",74806
  95 DATA "F4C39DF4FD460D0E01CD5
FF4FE1820030D2811FEFA20010C04CDC
CF6CD740030",55658
  96 DATA "E8C37BF5FD700DCD67F4C
39DF47ECDE3F6CDB61822SD5CFE3AC8F
ECBC8FE0D37",82331
  97 DATA "C82318E9FE22C0237EFE2
2C8FE0D20F7F137C9DFFECD200AE7CD8
21CCDB6F5C3",83658
  98 DATA "161DCDB6F5C3131D3E031
8023E02CD3025C40116CD4D0DCDDF1FC
DB6F5C9CD30",63706
  99 DATA "25280B3E01CD0116CD03F
3FD360201CD55F7CDB6F5ED4B685C3A6
B5CB838030E",59119
  100 DATA "2147ED43885C3E1990328
C5CFDCB0286CDD90DC303F3CD4E2028F
BFE28200EE7",65074
  101 DATA "CDDF1FDFFE29C28A1CE7C
345F8FECA2011E7CD1F1CFDCB37FEFDC
B0176C28A1C",75542
  102 DATA "180DCDB2CD243F8CD1F1
CFDCB37BECD3025CA46F8CDBF1621715
CCBB6CBEE01",72370
  103 DATA "0100CB7E200BFD7E01E64
020020E03B677F7360D79 0F0F30053E2
2122B77225B",34863
  104 DATA "5CFDCB377E202C2A5D5CE
52A3D5CE521CEF7E5FDCB30662804ED7
33D5C2A615C",60759
  105 DATA "CDA711FD3600FFCD73F3F
DCB01BECDB9211803CD73F3FD362200C
DD621200ACD",63984
  106 DATA "4BF2ED4B825CCDD90D217
15CC8AECB7ECBBE201CE1E1223D5CE12
25F5CFDCB01",67250
  107 DATA "FECDB9212A5F5CFD36260
0225D5C18172A635CED5B615C37ED524
44DCDB22ACD",57066
  108 DATA "FF2A1803CDFC1FCD4E20C
A55F7C9CDB228C28A1CCD30252008CBB
1CD9629CDB6",69227
  109 DATA "F5C3152C3E0318023E02F
D360200CD3025C40116DFCD702038 14D
FFE3B2804FE",52742
  110 DATA "2C2006E7CD821C1808CDE
61C1803CDDE1CCDB6F5C32518DFCD302
5C2371ECDFB",68345
  111 DATA "24FE2CC4B6F5E718F5DFF
E2C2806CDB6F5C37724E7CD821CCDB6F
5C39423DFFE",86102
  112 DATA "2CC28A1CE7CD821CCDB6F
5C32D23E7CD1F1CCD3025C2F51DDFFE2
C28F1CDB6F5",79479
  113 DATA "C9DFCD3025C2651FFDCB0
1F6CD8D2C3016E7FE242005FDCB01B6E
7FE28203CE7",68237
  114 DATA "FE292820CD8D2CD28A1CE
BE7FE242002EBE7EB010600CD5516232
3360EFE2C20",49604
  115 DATA "03E718E0FE292013E7FE3
D200EE73A385CF5CDFB24F1FDAE01E64
```

(además de **SAVE**, **LIST**, **LLIST**, **RESTORE**, **RUN** y **ON**).

Para su utilización sólo debemos ingresar **RENUM** (**CAPS SHIFT** + **R** en Modo Extendido) y agregarle el número de línea a partir del cual queremos que comience el programa ya renumerado y a continuación (''coma'' mediante), el ''intervalo'' deseado entre línea y línea.

Pero no termina aquí la capacidad de este importante comando. También puede aplicarse para renumerar parcialmente bloques del programa (sub-programas y subrutinas). Cómo??? Muy simple, definiendo previamente los ''rangos'' de numeración deseados para cada bloque del programa y haciendo, para cada uno de ellos:

RENUM l,i,p,u (y **ENTER**)

Donde **l** será el primer número de línea del bloque previsto e **i** el intervalo entre línea y línea, tal cual vimos anteriormente. La variante en este caso es que **p** nos indicará el primer número de línea del programa ''viejo'' (el desprolijo) y **u** el último número de línea de dicho programa. El **MEGA-BASIC** sólo se ocupará, en este caso, de renumerar las líneas comprendidas entre ambos números (**p** y **u**), dejando intactos los números de líneas de las restantes sentencias del programa, pero por supuesto modificando (reasignando números de líneas) todas aquellas instrucciones que se refieran al rango renumerado.

Repitiendo varias veces este proceso con los distintos bloques del programa, obtendremos una prolija y por demás segura **numeración estructurada** de nuestro, ahora, ordenado programa, la cual podremos a su vez variar cuantas veces querramos.

Veamos ahora un ejemplo:

a. Programa anterior:

```
  9 REM rutina 1
 10 PRINT " 1";
 11 GO SUB 50: GO SUB 1000
 16 REM 1.1
 17 REM 1.2
 25 REM 1.3
 26 GO TO 9
 50 REM rutina 2
 51 PRINT " 2";
 60 REM 2.1
 65 REM 2.2
 70 REM 2.3
 71 REM 2.4
100 REM 2.5
111 RETURN
1000 REM rutina 3
1001 PRINT " 3";
1002 REM 3.1
1010 REM 3.2
1111 REM 3.3
1112 RETURN
```

b. Ahora tipeamos:

```
RENUM 2000,10,1000,1500   <ENTER>
RENUM 1000,10,50,200      <ENTER>
RENUM 10,10,9,100         <ENTER>
```

c. Y, como por arte de magia, veremos:

```
10 REM rutina 1
20 PRINT " 1";
30 GO SUB 1000  GO SUB 2000
40 REM 1.1
50 REM 1.2
  60 REM 1.3
  70 GO TO 10
1000 REM rutina 2
1010 PRINT " 2";
1020 REM 2.1
1030 REM 2.2
1040 REM 2.3
1050 REM 2.4
1060 REM 2.5
1070 RETURN
2000 REM rutina 3
2010 PRINT " 3";
2020 REM 3.1
2030 REM 3.2
2040 REM 3.3
2050 RETURN
```

Está RE-bueno, ¿no?...

SWAP

Este comando, como su nombre lo sugiere (swap en inglés significa cambiar, permutar, y aseguramos que no tiene nada que ver con la popular serie de televisión Swat), podremos utilizarlo para **intercambiar** simultáneamente los contenidos de dos variables cualesquiera, sin necesidad de recurrir a una variable intermedia, y con ahorro de tiempo, memoria y, lo que es aún mejor, simplificándonos la programación.

O sea que si hacemos:

SWAP x,y (y **ENTER**)

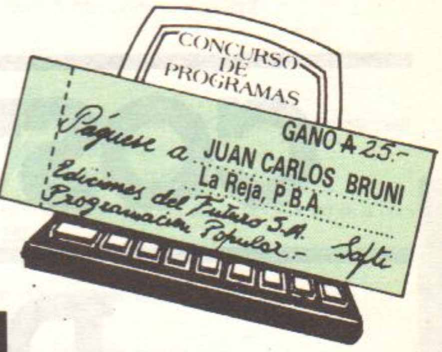
estaríamos transfiriendo el contenido de ambas variables entre sí, tal como lo haríamos (sin **MEGA-BASIC**) con:

LET a=x: LET x=y: LET y=3

Sólo resta decir al respecto que esta nueva instrucción puede ser utilizada tanto para variables **numéricas** como **alfanuméricas**.

```
0C28A1CCDB6",75472
 116 DATA "F5CD991EC5CD991EE1505
9EBA7ED5222F1F0ED53F3F021F8F0360
1ED7B3D5CF1",80465
 117 DATA "C35FEFCDD52D3804FE023
8022CF0A874F0600217DF9095E2356ED5
30CFDC9F409",56630
 118 DATA "35FDCD3025C27AF5CD90F
9CD5FF4C39DF4CD5FF4CD882C30D3F50
8F1FE3A3804",77782
 119 DATA "E6DFD607D630BB2001080
D67F4FE2C200ACD67F4CD882C38E3CF0
B08C9CDD3025",62447
 120 DATA "C27AF518C6CD302520190
D1F1C0E2CCD82F43A3B5CF5CD1F1CF1F
D5601AAE640",64083
 121 DATA "C8CF0B2A5D5CE5CDFB24F
DCB0176CC0AFACD67F4CDFB24E1225D5
CCD1F1CCDFF",76497
 122 DATA "EACD67F4CD1F1CCDFF2AC
9CDF12B78B1CAB22AE1D5F7D1C52A615
CEBED80CD67",86477
 123 DATA "F4CDFB24C1E1C5225D5CC
D1F1CCDFF2AED5B615CC1CDB22AC300F
ACD5FF4CD82",78725
 124 DATA "1C47E6FEFEEC20990E00C
5CD302537C4D52D30023EFFF5CD67F4C
D881CF1CD30",72867
 125 DATA "2528083D281CF5EF0238F
147CD5FF4FE2C7828E2F1CD3025C8D0F
EEDCAED1EC3",84096
 126 DATA "671EC10DC518E4CD30252
008DFFE2CC0E7C37A1CDFFE2C2809211
027E5010000",48556
 127 DATA "180BE7CD7A1CCD991EC5C
D991EE1A7ED4209380E222DFCED432FF
CCD991E78B1",67269
 128 DATA "2002CF0AED4333FCCD991
EED4331FCED4B2DFCCD02FCDDE5CCB81
958235EE1A7",76722
 129 DATA "E05230DE2A5D5C225F5CC
D2CFB3EFECD01162A5F5C225D5CFD362
6002A2DFC23",49418
 130 DATA "CD6E197EF53E80772A2FF
CCD6E19ED5B31FCD57EFE802811D1722
3732BE52A33",62269
 131 DATA "FC19E3CDB819EB18EAD1F
177C92A535C7EE6C0C023234E2346E50
9224D5CE122",58898
 132 DATA "28FC225D5C180EE11602C
D72F13006A4D5C2318DBCD67F4E5216
AFB4FCDDC16",65945
 133 DATA "30E516005E19CD2C1618D
CF8101C25EC3AED38E136F034F732E53
00001CAFF2A",62069
 134 DATA "5D5C237ECDB618FE0DC8A
0B920F4225D5C181701ECFECD7EFBE7C
D821CF5EF02",75091
 135 DATA "38F1FE2CC0CDA9FB18EF2
A5D5C237EFE0E2806CD182DD818F4E5C
DB618FE3A28",65168
 136 DATA "09FE0D2805FE2CD1C0D52
B2B462B4ECD02FCE1D8DDE5C1E523232
3712370E1ED",72797
 137 DATA "5B5D5C13225B5CC5CDE51
93EFFCD0116C1CD1B1A2A4D5CED5B2BF
CA7ED52EB72",68931
 138 DATA "2B73C92A2DFCA7ED42D82
A2FFCCD1CFF0009DD2A31FCC5CD6E19C
1CD8019D0ED",74480
 139 DATA "5B33FCDD19C5CDB819EB1
8EE1261F40101000A000A002AB25C363
E28F92B2B22",40318
 140 DATA "3D5CFD213A5C21B65C224
F5C11AF15011500EBED50EB2B22575C2
322535C2248",45266
 141 DATA "5C3680232259C360D233
6602322615C22635C22655C3E0F328D5
C328F5C3248",39380
 142 DATA "5C3E01D3FEFD35C6FD35C
ACD12FF11105C010E00ED50FDCB01CEC
DDF0EFD3631",67538
 143 DATA "02CD00F3AF11B5FCCD0A0
CFD360100FDCB02EEC35FEF807F50726
F6772616D61",64362
 144 DATA "63696F6E20506F70756C6
1F2CF0BCDF0FCE5F57E32A5FEF1FE203
025FE103821",71442
 145 DATA "FE18380AD61811EBFECD1
1FDE1C9E11845210EFDFDCB014EC023F
DCB0246C823",67395
 146 DATA "C92A0CFDCD2C18E13AA5F
E77C9F409000000F5CD410C3E20FDCB0
146CC3B0C1A",52338
 147 DATA "E67FCD3B0C1A138730F5D
1FE482803FE523E20C33B0CCD030BFE2
0D21BFEFE10",61281
 148 DATA "386AFE183062FE1630531
14FFDC37D0A11CDFCCD800A2A0E5C573
A0DFDFE09CA",59776
 149 DATA "910A7DFE16DA1122201E4
44A3E3F91DAAC0AC6044FAFCB3938023
E1532A5FECD",56129
 150 DATA "A30ACDF0FCC307FD7CCD0
30BF53AA5FEFE01CB11F181D603E63FC
3CA0A11A1FD",73900
 151 DATA "18AB114FFDC3700A3E3F1
870FE06200BCB213E4491E630C61018C
EFE08202E3A",49737
 152 DATA "A5FEA72805AF32A5FEC93
E1532A5FE0C3E22B92015FDCB014E200
9040E023E19",39269
 153 DATA "B82007050E21AF32A5FEC
3D90DFE092013FD7E57F5FD77573E20C
D41FEF1FD77",76012
 154 DATA "57C3DC0AFE0D20A0FDCB0
14EC20AFFAF32A5FE0E21CD550C05C3D
90DCD21FEC3",67563
 155 DATA "DC0AFE90381CD6A5D25F0
BC615F53AA5FEA728053E20CD41FEF1C
5ED4B7B5CC3",73542
 156 DATA "6A0BC501C6EAEB2138B5CC
B86FE202002CBC626006F29292909C1E
B3AA5FEA728",59707
 157 DATA "06AF32A5FE181F793D3E2
1200E054FFDCB014E2806D5CD0AFFD17
9B9D5CC550C",62413
 158 DATA "D13E1532A5FEC5E5FD7E5
706FF1F3801041F1F9F4F3E08A7FDCB0
14E2805FDCB",56558
 159 DATA "30CE37E80818003E0FA14
F3EF0B047AFED671AA0AEA912AFED6F1
80D3EF0A14F",63114
 160 DATA "3E0FB0471AA0AEA912083
81814233D20D3EB25FDCB014ECCDB0BE
1C13AA5FEA7",69692
 161 DATA "C00D23C9083E20835F081
8E1BF454C53C54F4E204552D241555 4C
F52454E55CD",55054
 162 DATA "4FCE535741D04D4F44C5C
021A5FE3600C3CD0E21CDFC22C55C21C
615C9ED42C8",69362
 163 DATA "3FD1D8D5C9",3149
```

ATAQUE ESPACIAL

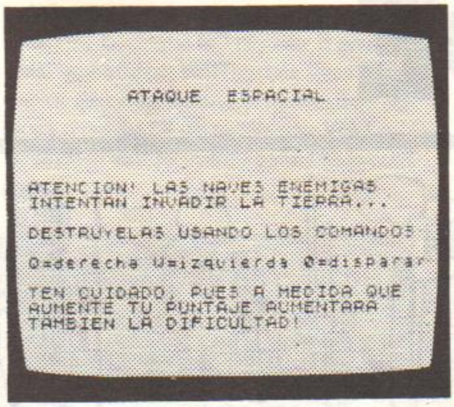


El programa de **Juan Carlos** nos recuerda al **"Space Invaders"** pero en una versión simplificada, muy loable teniendo en cuenta su corta extensión.

Veámoslo…

ESTRUCTURA DEL PROGRAMA

10/140	Inicialización y gráficos de pantalla
150/230	Rutina principal
250/280	Subrutina planeta invadido. Fin del juego
400/440	Subrutina disparo del misil
1000/1060	Instrucciones



VARIABLES

hi	Máximo puntaje
xi	Coordenada provisoria de la base
sc	Puntaje
niv	Nivel de dificultad
t	Puntaje provisorio
x	Coordenada de la base
a(16i	matriz de las naves espaciales

```
  10 REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@       ATAQUE ESPACIAL        @
@   Juan C Bruni-La Reja,PBA   @
@ Rev.Programacion Popular-N°8 @
@    CZSpectrum-TK90X-TS2068   @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  20 REM © Juan C. Bruni - 1985
  30 LET hi=0: LET x1=16
  40 FOR i=USR "a" TO USR "c"+7:
 READ j: POKE i,j: NEXT i
  50 DATA 60,126,255,66,255,255,
255,90
  60 DATA 0,8,8,8,8,28,42,0
  70 DATA 0,0,24,126,255,60,0,0
  80 GO SUB 1000
 100 LET sc=0: LET niv=17: LET t
=0
 110 BORDER 4: PAPER 5: BRIGHT 0
: CLS : INK 0
 115 PRINT AT 1,9;"ATAQUE ESPACI
AL"
 120 PRINT AT 10,1; PAPER 1; INK
 9;"PUNTOS: ";AT 10,24;"RECORD: ";
AT 12,26;hi: PAPER 5: BRIGHT 1
 130 FOR i=4 TO niv: PRINT AT i,
8;TAB 24: NEXT i
 140 LET x=16: DIM a(16): PRINT
AT niv,16; PAPER 5; INK 0;"♠"
 150 LET x$=INKEY$
 160 IF x$="q" THEN LET x1=x-1:
IF x1<8 THEN LET x1=23: PRINT AT
 niv,x;" "
 170 IF x$="w" THEN LET x1=x+1:
IF x1>23 THEN LET x1=8: PRINT AT
 niv,x;" "
 180 IF x$="0" THEN GO SUB 400
 190 PRINT AT niv,x; PAPER 5;" "
;AT niv,x1; INK 0;"♠": LET x=x1
 200 IF sc>=t+1000 THEN LET t=sc
: PRINT AT 20,10;"NUEVO NIVEL":
LET niv=niv-1: BEEP .05,10: BEEP
 1,0: PAUSE 50: GO TO 110
 210 LET a=INT (RND*16)+1
 220 PRINT INK 5; PAPER 5;AT a(a
)+6,a+7;" "; INK 2;AT a(a)+7,a+7
;"◆": LET a(a)=a(a)+1: IF a(a)=n
iv-7 THEN GO TO 250
 230 GO TO 150
 240 REM muerto
 250 IF sc>hi THEN FLASH 1: LET
hi=sc: PRINT AT 12,26;hi
 255 FOR i=30 TO 0 STEP -1: BEEP
 .1,i: NEXT i: FLASH 0
 260 PRINT #0;"Pulse una tecla p
ara otro juego"
 270 PAUSE 0
 280 GO TO 100
 390 REM disparo
 400 FOR i=niv-2 TO 6 STEP -1: P
RINT AT i,x; PAPER 5; INK 1;"ı";
AT i+1,x;" "
 410 BEEP .01,30-i
 420 IF ATTR (i-1,x)=106 THEN PR
INT PAPER 5;AT i,x;" ";AT i-1,x;
" ": LET sc=sc+50: PRINT AT 12,2
;sc: LET a(x-7)=0: RETURN
 430 NEXT i: PRINT AT 6,x;" "
 440 RETURN
1000 BORDER 2: PAPER 4: INK 0: C
LS
1010 PRINT AT 1,8;"ATAQUE  ESPAC
IAL";AT 5,0;
1020 LET a$="ATENCION! LAS NAVES
 ENEMIGAS     INTENTAN INVADIR LA
 TIERRA...  "
1025 GO SUB 2000
1030 LET a$="DESTRUYELAS USANDO
LOS COMANDOS:
            Q=derecha W=izquier
da 0=disparar"
1035 GO SUB 2000
1040 LET a$="TEN CUIDADO, PUES A
 MEDIDA QUE  AUMENTE TU PUNTAJE
AUMENTARA     TAMBIEN LA DIFICULT
AD!"
1045 GO SUB 2000
1060 PRINT #0; FLASH 1;"Pulsa un
a tecla para comenzar": PAUSE 0:
 FLASH 0: RETURN
2000 PRINT
2010 FOR f=1 TO LEN a$: PRINT a$
(f);: IF a$(f)<>" " THEN BEEP .0
5,30
2020 NEXT f
2030 RETURN
3000 SAVE "ataque" LINE 10
```

CGDU	(GRAPHICS) + TECLA
♠	a
ı	b
◆	c

Tenemos un plan bien programado.

Consiste en cuatro pagos sin interés, con la calidad y el diseño que usted merece



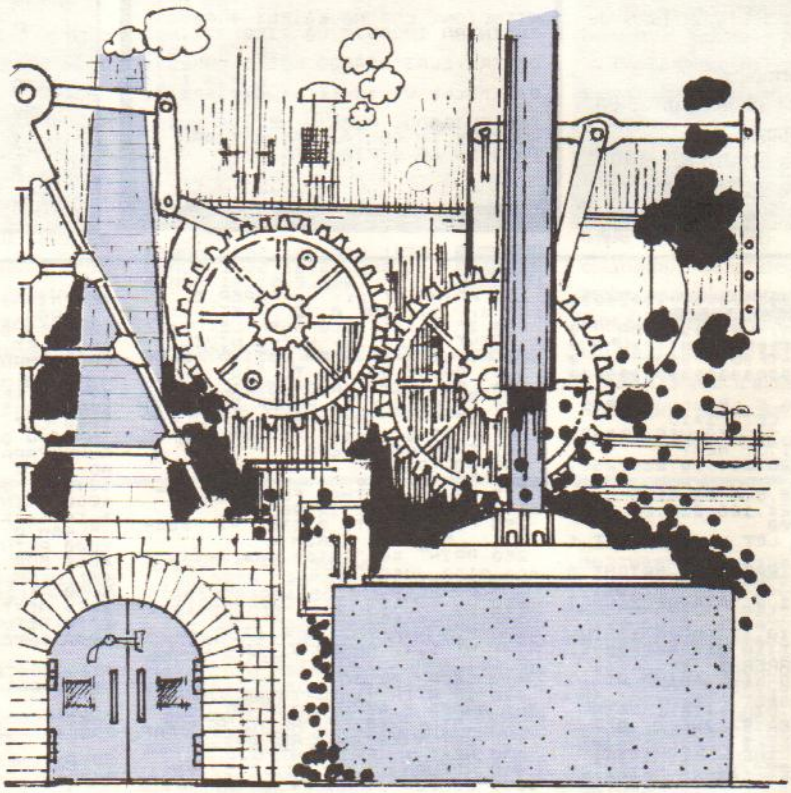
aguilar equipamientos s.a.

Está en las mejores empresas.

Fábrica: G. de Laferrere 1946 - Cap.
tel. 631-1742 8909 - 632-2201
Ventas: Luis Saenz Peña 443 - Cap **Est. Propio**
• Directorio 403 - Cap.
• Suipacha 944 - Cap. • Cabildo 2869 - Cap
Dispo: Av. Maipu 2263 - Olivos

COSTOS DE FABRICACION





Es éste un muy interesante y práctico programa de aplicació que nos envía el amigo **Raúl** desde **Santiago del Estero** (Madre de ciudades'', como él mismo la define), acompañado de una muy elaborada, cuidada y didáctica documentación que nos posibilita así ponerlo al alcance de todos nuestros amigos lectores.

Nos dice **Raúl** que el título elegido tiene mucho que ver precisamente con la idea original con que fue diseñado, pero posteriormente lo fue dotando con detalles de generalidad tales que pudo ser finalmente usado en una enorme variedad de aplicaciones no necesariamente relacionadas con ''fabricación''. Las ideas fundamentales que han guiado el diseño de este programa son: Eficacia, generalidad, uso fácil, mínima memoria ocupada.

Dada la extensión de la nota, publicaremos en este número el programa y la explicación para su utilización, dejando para números siguientes los bien realizados fluxogramas y las tablas de variables.

Antes de pasar al trabajo (dedicado con todo cariño a sus hijos **Fernando Raúl** y **Ana Cecilia**), resta decir que la ilustración que publicamos fue realizada por **Gabriela Gómez Filas**, discípula de **Raúl** y estudiante de abogacía y computación. Muchas gracias, **Raúl**...

OBJETIVOS

Este programa ha sido hecho con la mente puesta y dedicado al pequeño industrial o artesano, cocinero, capintero, herrero, constructor, mecánico, pintor, electricista, etc. etc., quien frecuentemente necesita conocer con rapidez y exactitud el aspecto económico de su oficio. Este programa le facilitará la tarea en la elaboración de **presupuestos**, conocer sus **costos de fabricación**, los **precios de venta** de sus productos, las **comisiones** que deberá abonar a sus empleados y vendedores si los tuviese. Todo este cálculo realizado en contados segundos, con todas las variantes que se le ocurra, sin matemáticas difíciles y de una manera cómoda, sencilla y muy bien guiada, mediante un sistema de **menús**.

De ahora en adelante ya no habrá excusa para una estimación de costos demasiado burda, o de quedarse "cortos" en el presupuesto. En definitiva, este programa viene a solucionar elegantemente esta barrera frecuentemente infranqueable entre profesional y cliente, arrojando luz en un tema que es causa principal de la incomunicación, esto es el aspecto económico de toda realización.

RESULTADOS OBTENIDOS

Detallaremos a continuación cuáles son los resultados que entrega este programa a partir de los datos que se le ingresen, y también se darán los detalles de los cálculos que efectúa en cada caso.

a - Costo de materiales

Para cada tipo o modelo del producto a fabricar el programa efectúa el cómputo del total de dinero necesario en concepto de **costo de materiales** o sea de **materia prima**. Esta columna se identifica mediante su abreviatura **CMAT**. Este resultado y todos los demás son entregados **únicamente con cuatro dígitos**, incluyendo la coma decimal. Esto es así debido al formato de las tablas utilizadas que aceptan números de solamente cuatro caracteres. Será necesario por lo tanto tomar los recaudos cuando esta cantidad se espera que supere los 9999. Otra advertencia es que debido al poco número de columnas que presenta esta máquina, las cantidades **no contengan el formato financiero** (con los dígitos finales que representan los centavos) ni tampoco se ha previsto el encolumnamiento de los puntos decimales por la misma razón expuesta. Si el honorable futuro usuario fuese por ejemplo un vendedor de automóviles o se maneja en su profesión en los negocios inmobiliarios, seguramente las cantidades involucradas en el cálculo sobrepasarán el número 9999. Para estos casos sugiero que las cantidades ingresadas sean expresadas en **miles de australes** en lugar de australes simplemente, de esta manera se reducen notablemente (a tres menos) los dígitos necesarios para cálculos correctos.

Para la obtención del costo de materiales, es decir **CMAT**, la máquina tiene implementada una rutina que efectúa la suma de la columna correspondiente al tipo considerado. Los valores que contiene cada elemento de esta columna son cada uno (de cada tipo) resultado del producto del **costo unitario** (o específico) de cada material por la **cantidad de este material**. Este cálculo repetido por supuesto para cada **material** y también para cada modelo o **tipo** (se usó **tipo** y no modelo debido a su menor longitud a la hora de su escritura en la tabla).

Los datos y los resultados obtenidos son todos ubicados en una matriz bidimensional llamada "matriz de costos" (vease tabla de dimensiones).

b - Costo bruto

Es el resultado de afectar a la cantidad anterior, **CMAT**, de los índices de aumento en porciento correspondientes por **mano de obra** (variable numérica llamada **MO**) y por **comisión de venta** (variable numérica llamada V). La cantidad **MO** (%) sobre el **precio de venta** (variable llamada **VENT**) es la que deberá pagarse al obrero que ejecuta el trabajo de fabricación del producto y la cantidad **CV** (%) también sobre **VENT** es la que deberá abonarse por unidad vendida al vendedor del producto.



Por lo tanto, el **costo bruto**, en adelante **CBRU**, queda expresado matemáticamente así:

CBRU=CMAT+VENT (MO/100)+VENT. (CV/100)

Sacando factor común **VENT** se tiene:

CBRU=CMAT+VENT (MO/100+CV/100)

Por otra parte definimos como precio de venta a:

VENT=CBRU (1+UT/100) (1)

o sea que el costo bruto incrementado en la **utilidad (UT)** o ganancia que se desee. Reemplazando este valor en la expresión anterior:

CBRU–CMAT+CBRU (1+UT/100) (MO/100+CV/100)

despejando **CBRU** arribamos finalmente a:

CBRU–CMAT/(1-(1+UT/100) (MO/100+CV/100))

que es la fórmula básica que se calcula en la línea **396** donde por supuesto está expresada en BASIC.

c - Precio de venta

Mediante la expresión (1) se determina el **precio de venta** que como ya sabemos se identifica mediante **VENT** en la columna de costos finales que la máquina muestra. Su cálculo en línea **397**

d - Costo de mano de obra

Según la definición ya dada anteriormente esta cantidad se determina así:

MOBR–VENT (MO/100)

cuyo cálculo en el programa se lleva a cabo en la línea **398**.

e - Comisión de venta

Por definición aquí también:

CVEN–VENT (CV/100)

y su cálculo se verifica en la línea **399**.

f - Presupuesto general

Para una determinada cantidad de unidades a fabricar por cada tipo, es deseable conocer cuál es la *cantidad total de cada material con su costo correspondiente* como requisito previo que ayudaría económicamente en la decisión para encarar o no el proyecto. Este cálculo lo denominamos *presupuesto general de materiales*.

Para obtener la cantidad total de cada material para el conjunto de modelos elegidos, se deberá sumar las cantidades individuales del material considerado de cada tipo elegido, o sea:

CANT = Sumatoria de las cantidades de material empleada en cada tipo

Si ocurre que se elige determinada cantidad de unidades para cada uno de estos tipos para los cuales se desea el presupuesto, la definición anterior se dará de una manera más general así:

CANT=Sumatoria de las cantidades de cada material empleada en cada tipo, multiplicada por las cantidad de unidades a fabricar de cada tipo.

Por otra parte, es igualmente útil el conocimiento de lo que cuesta cada una de estas cantidades, lo que nos obliga a definir otra columna llamada precisamente **COSTO**:

COSTO=Costo del total de cada material a utilizar en el lote de fabricación considerado.

Y como una consecuencia inevitable sería también conveniente conocer el total de desembolso necesario con el que habrá que pagarse a los proveedores para la obtención de la totalidad de las materias primas o materiales para encarar con éxito la fabricación del lote de unidades propuesto.

Este *total de costos* se definió como **TL** (como un acronimo de la palabra total):

TL=Sumatoria de los costos de cada material.

Todo este cálculo se lleva a cabo entre las líneas **520** y **625** donde está precisamente la subrrutina **PRESUPUESTO**.

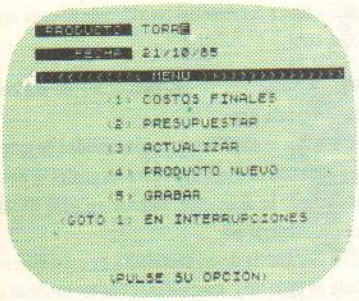
Una vez tipeado el programa, el que recomendamos ir saveando parcialmente "por las dudas", haremos **GOTO 4015** (no **RUN**). Si el tipeo anduvo bien se tendra ante la vista la **"primera página"** del programa donde aparece una advertencia o prohibición de la ejecución de la sentencia **RUN** (debido a su higiénica componente **CLEAR**) que como ya se sabe borra o "limpia"

(CLEAR) el contenido de las variables del programa, lo cual sería desastroso para este programa en particular. También se muestra un cartel que nos advierte de la cantidad de memoria en Bytes ocupada por el programa (línea **4009**) pero sin considerar la memoria que es ocupada por las **DIM**ensiones.



En la línea **4013** del programa podemos advertir que existe allí una estructura de control que nos permite continuar con sólo digitar cualquier tecla (distinta de **BREAK** o **SHIFT** obviamente). Y como esto es ya una costumbre trivial, no es necesario a mi entender escribir también la advertencia "pulse una tecla para continuar".

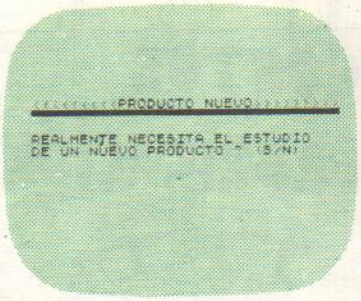
Hecho lo anterior, la pantalla ahora es ocupada por el **MENU PRINCIPAL** donde un parpadeo de la opción



4 es lo que se destaca primero. En la parte superior está escrito "sin llenar" PRODUCTO:? debajo FECHA: DD/MM/AA (nos indica el formato en que se debe ingresar más adelante el dato "FECHA") presentando un aspecto como el siguiente:

El parpadeo de la opción **4** nos invita a que sea ésta precisamente la opción elegida puesto que todas las demás salvo **5** están bloqueadas. Si se elige **5** tenemos facilitada la tarea para la realización inmediata de una **copia de seguridad**, la que se ha de grabar con un nombre constituido por un caracter **?** , indicado como **SAVE "?"**). Una vez hecha esta copia (habiendo digitado **G** para "grabar" optamos ahora por la opción **4** luego de pasar nuevamente por la primera página.

Digitamos **4** PRODUCTO NUEVO y tenemos ante nosotros la **"tercera página"**.



De inmediato el programa realiza una pregunta de protección o advertencia para evitar la pérdida de un cálculo contenido en ese momento en la **RAM**. Para una contestación negativa queda protegido el cálculo y el control retorna al menú principal, mientras que para una afirmativa, el control ingresa a la zona de **DIM**ensionamientos, posicionando a cero todas las matrices, *limpiando* todo vestigio de datos del cálculo anterior.

El control ingresa de esta manera a la subrutina del cuestionario previo, donde se solicita al operador el ingreso de los siguientes datos:



Nombre del producto: (con un máximo de 15 caracteres).

Fecha actual: solicitada en el formato DD/MM/AA (día, mes, año). Luego solicita el **número de tipos** (o modelos) de este producto que se desan fabricar mostrándonos el máximo admitido (para evitar un sobredimensionamiento y un seguro error 4 posterior). Luego deben ingresarse los **nombres dados a cada tipo** con un máximo de 6 (seis) caracteres; una entrada ilegal en este sentido queda automáticamente filtrada mediante la atribución procusteana de la matriz **T$(TP,6)** (**TP**=número máximo de tipos admitidos, **6**=número de caracteres máximos en cada modelo) que es donde se "guardan" estos nombres. Se completa así la **"cuarta página"** con una estructura de control para pulsar una tecla y continuar.

Pulsando una tecla ingresamos a la **"quinta página"** desarrollada por la subrrutina **Datos del material**. Esta página se desarrolla totalmente "**SCROLL**'ada" en virtud de la necesidad de independizarse de la cantidad de materiales a ingresar.

En primer término aparece el título **DATOS DEL MATERIAL**, seguidamente tres observaciones o advertencias, identificadas mediante un asterisco.

```
  1 REM
 COSTOS DE FABRICACION.
 ING.RAUL D VERA-SGO D/ESTERO
 REV.PROGRAMACION POPULAR Nº 8
 CZ1000/1500-TK83/85   9,3 KB
 10 GOTO VAL "4015"
 11 REM PRODUCTO NUEVO
 12 CLS
 13 LET B=0
 14 LET L=0
 15 LET x$="PRODUCTO NUEVO"
 16 GOSUB G3
 18 PRINT ,,"REALMENTE NECESITA
 EL ESTUDIO  DE UN NUEVO PRODUC
TO ? (S/N)"
 26 GOTO VAL "26"+(INKEY$="S")+
VAL "74"+(INKEY$="N")+VAL "3989"
 102 CLS
 103 REM CUESTIONARIO PREVIO
 105 LET x$="CUESTIONARIO PREVIO
"
 110 GOSUB G3
 112 PRINT AT 3,0;"INGRESE LOS S
IGUIENTES DATOS:"
 115 PRINT AT CI,0;"NOMBRE DEL P
RODUCTO:   (MAX.15) "
 120 INPUT P$
 121 IF LEN P$>15 THEN GOTO VAL
"120"
 125 PRINT AT CI,0;"PRODUCTO = "
;P$;"
 130 PRINT AT 6,0,"FECHA ACTUAL?
 (FORMATO DD/MM/AA)"
 135 INPUT F$
 140 PRINT AT 6,0;"   FECHA = "
F$;"
 145 PRINT AT 7,0;"Nº TIPOS DE "
P$+"S (MAX.";TP;")"
 150 INPUT T
 152 IF T>TP THEN GOTO VAL "150"
 155 PRINT AT 7,0;"T = ";T;" TIP
OS A FABRICAR                "
 165 DIM M$(D,VE)
 170 DIM A$(D,DO)
 175 DIM T$(TP,6)
 180 DIM C(D+CI,DO+TP+CI)
 190 LET C$(U)="COSTO COMERCIO"
 195 LET C$(DO)="TOTAL UNIDADES"
 200 LET C$(3)="ENTRAN EN CADA'
 220 FOR P=U TO T
 225 PRINT AT P+8,0;"ENTRE NOMBR
E DEL TIPO ";P;" (MAX 6)"
 230 INPUT T$(P)
 235 PRINT AT P+8,0;"TIPO ";P;"
= ";T$(P);"
"
 240 NEXT P
 245 PRINT "(PULSE UNA TECLA PAR
A CONTINUAR)"
 247 IF INKEY$="" THEN GOTO VAL
"247"
 254 REM DATOS MATERIAL
 255 CLS
 256 LET C=U
 257 LET L=VU
 259 SCROLL
 260 LET x$="DATOS DEL MATERIAL'
 261 SCROLL
 262 GOSUB G3
 264 SCROLL
 265 PRINT "*INGRESE <£>, DESPLE
S DEL ULTIMO"
 266 SCROLL
 267 PRINT "*MAXIMO: ";D-U;" MAT
ERIALES CON 16 KB"
 268 SCROLL
 269 PRINT "*REPITE CON "" "" AN
TES DEL NUMERO"
 270 SCROLL
 271 FOR J=U TO D-U
 272 SCROLL
 273 LET L=0
 275 PRINT "NOMBRE DEL MATERIAL
Nº";J;" ?"
 280 INPUT M$(J)
 281 IF M$(J,U)="£" THEN GOTO VA
L "360+840*(B=U)"
 282 SCROLL
 285 PRINT J;"-";M$(J);
 288 REM PERFORATURAS
 290 FOR P=U TO 18
 295 IF M$(J,P)=" " AND M$(J,P+U
)<>" " THEN GOTO VAL "305"
 300 NEXT P
 305 LET A$(J)=M$(J,U)+M$(J,P+U)
 310 PRINT TAB 28;A$(J)
 312 LET U$="0"
 315 FOR I=U TO T+DO
 320 LET N=(I<=DO)*I+(I>DO)*3
 322 SCROLL
 325 PRINT TAB U;C$(N);" ";
 330 IF I>DO THEN PRINT T$(INT (
(DO+I-DU)/DO));"? ";
 333 IF U$(U)<>" " THEN INPUT U$
 334 IF U$="0" AND I=DO THEN GOT
O VAL "333"
 335 LET W=VAL U$
 336 IF I<=DO THEN LET C(J,I)=W
 337 IF I>DO THEN LET C(J,DO+I-D
O)=W
 340 IF I=DO AND W<>0 THEN LET C
(J,3)=C(J,U)/C(J,DO)
 345 IF I>DO THEN LET C(J,DO+I-U
)=C(J,3)*C(J,DO+I-DO)
 353 PRINT (STR$ W+"    ")[ TO C
I)
 355 NEXT I
 359 NEXT J
 360 REM PORCENTAJES
 361 CLS
 362 LET L=0
 363 LET x$="PORCENTAJES"
 364 GOSUB G3
 365 PRINT ,,"PORCENTAJE DE MAYO
R OBRA SOBREEL PRECIO DE VENTA
?";
 366 INPUT MO
```

```
(<<<  <DATOS DEL MATERIAL>>>)
+INGRESE (f)  DESPUES DEL ULTIMO
+MAXIMO  20 MATERIALES CON 16 KB
+REPITE CON "  ANTES DEL NUMERO

NOMBRE DEL MATERIAL Nº1 ?
1=CANO 1                    C1
 COSTO COMERCIO  2.7
 TOTAL UNIDADES  0
 ENTRAN EN CADA  TT3    ? 3
 ENTRAN EN CADA  TT6    ? 3
 ENTRAN EN CADA  TT9    ? 3
 ENTRAN EN CADA  TT12   ? 3
NOMBRE DEL MATERIAL Nº2 ?
2=CANO 2                    C2
 COSTO COMERCIO  3.23
 TOTAL UNIDADES  0
 ENTRAN EN CADA  TT3    ? 0
 ENTRAN EN CADA  TT6    ?
```

La primera nos invita a digitar el caracter **"libra" (shift SPACE)** después de haber ingresado el último material componente del producto. Este caracter es identificado por el programa en la línea **281** para "romper" el bucle **J** de ingreso de datos del material.

El segundo asterisco nos habla de la capacidad máxima **DIM**ensionada para contener estos materiales (para 16 KB o sea máquinas con esta capacidad nominal de memoria RAM). Esta cantidad máxima de materiales admitida dependerá del problema o producto en particular en estudio, y en general podrá definirse una función que denominamos **Dm** (cantidad máxima de materiales) de la siguiente forma:

Dm=f(Tp,MN) donde

TP=máxima cantidad de tipos o modelos admitidos.

Dm=máxima cantidad de materiales admitidos.

MN=cantidad de memoria nominal RAM de la máquina.

Por lo tanto es fácil deducir que si la máquina tiene expansión a 64 KB la cantidad **Dm** puede llegar a cuadruplicarse, y de igual manera con un razonamiento inverso, a mayor cantidad de **TP** menor será la cifra máxima de materiales admitidos **Dm** (pero esta vez la relación no será lineal). *(Ver dimensionamiento.)*

El tercer asterisco nos avisa de la posibilidad de no necesitar en cada tipo la digitación repetitiva de un mismo número. Para esto, ingresando un espacio **" "** previo al número que deberá repetirse **en todos los tipos**, este mismo número es ingresado como cantidad idéntica para todos los modelos. Según en qué punto se digite el espacio previo, es a partir de allí que las cantidades se repetirán automáticamente en todos los próximos tipos y hasta el final de ellos en el material considerado.

De inmediato la máquina solicita el ingreso del nombre del material Nº 1 (uno), el que admite mediante atribución procusteana un máximo de 20 caracteres **(DIM M$(D,VE)** en línea **165** donde **VE=20)**.

Una vez ingresado este nombre, el mismo ingresa a una subrrutina de la línea **288** hasta la **310** llamada **Abreviatura**, y es la encargada de tomar las primeras dos siglas (sigla: primera letra de una palabra usada para abreviarla) del nombre del material, el resultado (una cadena bicaractérica) se guarda en la matriz **A$(D,DO) (DO=2)** dimensionada en línea **170** y luego finalmente impresa en la pantalla a continuación del nombre del material (a manera de control). Estas abreviaturas serán de gran utilidad posteriormente para acortar o abreviar los nombres de los materiales a la hora de la presentación en pantalla.

A continuación la máquina nos solicita el **costo comercio** de este material. Este dato generalmente se toma de una factura de compra antérior cantidad que expresa el precio de una determinada cantidad de unidades (verbigracia: 500 tornillos ~~=~~ 6,86 el **costo comercio** es aquí ~~=~~ 6,86). Entrado este dato se pide a continuación el **total unidades** a que corresponde este costo (en nuestro ejemplo **total unidades** es igual a 500). Seguidamente la máquina solicita la cantidad de unidades de este material que se emplean para la fabricación de cada tipo o modelo mediante la pregunta reiterativa: **entran en cada (tipo)?** Y es aquí donde puede digitarse un espacio previo **" "** al número que expresa esta cantidad cuando éste debe repetirse en todos los modelos subsiguientes. Cabe acotar aquí que en todos los casos de ingresos de números, puede ingresarse en su lugar también una expresión numérica cuyo resultado sea precisamente la cantidad solicitada.

Después de solicitar el ingreso de la cantidad específica del último tipo, la máquina pedirá el nombre del próximo material y este ciclo comentado volverá a repetirse hasta que ya no tengamos material para ingresar, y resulta que ante la pregunta **nombre del material J?** debe contestarse con el caracter "libra" **shift SPACE** y la máquina comienza a "desenrrollar" la pantalla para ingresar de inmediato a la **"sexta página"** caracterizada con la subrutina **porcentajes** donde se

```
(<<<  <PORCENTAJES>>>)

PORCENTAJE DE MANO DE OBRA SOBRE
EL PRECIO DE VENTA

        MO=12 *

PORCENTAJE DE COMISION DE VENTA
SOBRE EL PRECIO DE VENTA

        CV=5 *

PORCENTAJE DE UTILIDAD ESTIMADA
SOBRE EL PRECIO DE COSTO BRUTO?

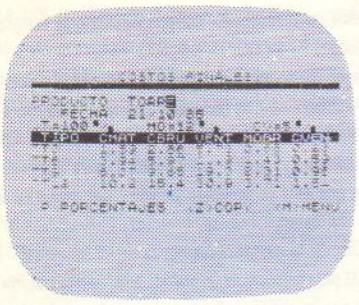
        UT=30 *

(M)MODIFICA  (C)CONTINUA
```

solicitará el ingreso de los porcentajes correspondientes por mano de obra MO, comisión de venta CV y finalmente la utilidad deseada UT todo rematado al pie de la página con un submenú de control que nos invita a digitar **"M"** si nuestro deseo es modificar los valores entrados o si pulsamos **"C"** continuar con el programa.

Si la tecla pulsada es **S** el submenú de pie de página es tapado por la leyenda **(calculando opción (1) espéreme)** con el número 1 parpadeando sirviendo esto de monitor del trabajo de la máquina. Evidentemente el programa ingresó a la subrrutina **cálculo costos** (de línea **380** hasta línea **400**). El tiempo empleado por este cálculo depende obviamente de la cantidad de materiales y de los modelos ingresados, en caso de querer acelerarlo digitando **381 FAST** y **401 SLOW** lo lograremos.

```
 367 PRINT ,,TAB DI;" MO=";MO;"
 %";"
 368 PRINT ,,"PORCENTAJE DE COMI
SION DE VENTA SOBRE EL PRECIO DE
 VENTA ?",
 369 INPUT CV
 370 PRINT ,,TAB DI;" CV=";CV;"
 %";"
 371 PRINT ,,"PORCENTAJE DE UTIL
IDAD ESTIMADA SOBRE EL PRECIO DE
 COSTO BRUTO?"
 372 INPUT UT
 373 PRINT ,,TAB DI;" UT=";UT;"
 %";"
 374 PRINT AT VU,0;"   (M)MODIFI
CA   (C)CONTINUA"
 377 GOTO VAL "377"-(INKEY$="M")
+VAL "17"+(INKEY$="C")+U
 380 REM CALCULO COSTOS
 382 PRINT AT VU,0;"(CALCULANDO
OPCION (1) ESPEREME)"
 385 LET E=0
 386 FOR P=U TO T
 387 LET C(J,P)=0
 388 FOR Q=U TO J-U
 389 PRINT AT VU,VE;" "
 390 LET C(J,P)=C(J,P)+C(Q,DO*P+
3)
 394 PRINT AT VU,VE;"1"
 395 NEXT Q
 396 LET C(J+U,P)=C(J,P)/(U-(U+U
T/100)+(MO/100+CV/100))
 397 LET C(J+DO,P)=C(J+U,P)*(U+(
UT/100))
 398 LET C(J+3,P)=(MO/100)*C(J+D
O,P)
 399 LET C(J+CU,P)=(CV/100)*C(J+
DO,P)
 400 NEXT P
 410 REM COSTOS FINALES
 411 CLS
 412 LET X$="COSTOS FINALES"
 413 LET L=0
 414 GOSUB 63
 415 PRINT "PRODUCTO: ";P$
 416 PRINT "    FECHA: ";F$
 417 PRINT "UT=";UT;"%","  ";"MO
=";MO;"%","  ";"CV=";CV;"%"
 419 PRINT " TIPO  CMAT CBRU VEN
  MOBR QUEN "
 420 FOR P=U TO T
 421 PRINT T$(P);TAB 7;((STR$ C(
J,P)+"    ")( TO CR));
 434 PRINT TAB 12;(STR$ C(J+U,P)
+"    ")( TO CR);TAB 17;(STR$ C(J
+DO,P)+"    ")( TO CR);
 443 PRINT TAB 22;(STR$ C(J+3,P)
+"    ")( TO CR);TAB 27;(STR$ C(J
+CU,P)+"    ")( TO CR)
 455 NEXT P
 458 PRINT ,,"(P)PORCENTAJES  (Z
)COPY  (M)MENU"
 460 IF INKEY$="Z" THEN COPY
 464 GOTO VAL "460"-(INKEY$="P")
*VAL "100"+(INKEY$="M")*VAL "355
5"
 470 REM UNIDADES A FABRICAR
 472 CLS
 480 LET X$="UNIDADES A FABRICAR
"
 485 GOSUB 63
 488 PRINT ,,"CUANTAS UNIDADES S
E FABRICARAN  DEL TIPO.....(0)
PARA NINGUNA"
 489 PRINT
 490 FOR P=U TO T
 495 PRINT TAB CU;T$(P);" ? ";
 500 INPUT C(J+CI,P)
 505 PRINT C(J+CI,P)," UNIDADES"
 510 NEXT P
 520 REM PRESUPUESTO
 523 CLS
 524 LET X$="PRESUPUESTO GENERAL
"
 525 GOSUB 63
 527 PRINT "PARA: ";
 528 FOR P=U TO T
 529 IF C(J+CI,P)<>0 THEN PRINT
C(J+CI,P);"/";T$(P);" ";
 530 NEXT P
 533 LET TL=0
 536 PRINT TAB 0;" MATERIALES=>>
 >P=>>  CANT.  COSTO"
 540 FOR Y=U TO J-U
 541 LET C(Y,DO*T+CU)=0
 542 LET C(Y,DO*T+CI)=0
 544 IF M$(Y,U)="£" THEN GOTO VA
L "585"
 545 FOR X=CU TO DO*(T+U) STEP D
O
 550 LET C(Y,DO*T+CU)=C(Y,DO*T+C
U)+C(J+CI,(3*X-6)/6)*C(Y,X)
 555 LET C(Y,DO*T+CI)=C(Y,DO*T+C
I)+C(J+CI,(3*X-6)/6)*C(Y,X+U)
 558 NEXT X
 560 LET TL=TL+C(Y,DO*T+CI)
 565 IF C(Y,DO*T+CU)=0 THEN GOTO
 VAL "575"
 566 PRINT M$(Y);
 570 PRINT TAB VU;(STR$ C(Y,DO*T
+CU)+"    ")( TO CR);
 573 PRINT TAB 27;(STR$ C(Y,DO*T
+CI)+"    ")( TO CR)
 575 NEXT Y
 585 PRINT " TL=TOTAL GENERAL=>>
>>>>>>A ";
 600 PRINT TAB 27;(STR$ TL+"    "
)( TO CR);TAB 31;"■"
 610 PRINT ,,"(K)OTRAS CANTID.  (
Z)COPY  (M)MENU"
 615 IF INKEY$="Z" THEN COPY
 625 GOTO VAL "615"+(INKEY$="K")
*VAL "145"+(INKEY$="M")*VAL "340
0"
1200 REM TABLA ELECTRONICA
1201 LET B=U
1202 FAST
1203 DIM Q$(J-U,23)
1206 FOR S=U TO J-U
1207 LET Q$(S, TO CU)="  "+STR$ S
1208 LET Q$(S,CI TO 13)=M$(S, TO
 CU)+"  "+A$(S)
1209 LET Q$(S,14 TO 18)=(STR$ C(
S,U)+"    ")( TO CR)
1210 LET Q$(S,19 TO 23)=(STR$ C(
```



Hecho este cálculo ingresamos a la **"séptima página"** encabezada por la subrrutina **costos finales** donde quedan plasmados los resultados finales motivo del cálculo de costo efectuado anteriormente. Observando la pantalla notaremos que para cada tipo (primer columna) se da el costo de materiales **CMAT**, el costo bruto **CBRU**, el precio de venta **VENT**, el monto a pagar en concepto de mano de obra **MOBR**, y finalmente la cantidad a abonar al vendedor en concepto de comisión de venta **CVEN**. Todos estos datos encabezados con el nombre del producto y más abajo por la fecha actual, mostrada **DD/MM/AA**. Más abajo los porcientos con los cuales se efectuó el cálculo para una más cómoda referencia.

El submenú de pie de página nos permite optar entre: **P porcentajes** para obtener otra tabla similar con porcentajes diferentes, **Z copy** para obtener de inmediato una copia en impresora y finalmente si digitamos **M** retornaremos al **menú** principoal.

Estando nuevamente en el menú principal notamos de inmediato que el parpadeo del **4** ha cesado, habiéndose habilitado simultáneamente las demás opciones. Así por ejemplo si optamos por la **1** retornaremos de inmediato (sin pasar por el tedioso cálculo de costos) a la página 7 abandonada recientemente y ya analizada en toda su extensión.

Si nos decidimos por la opción **2 presupuesto** del menú principal, ingresamos a la **"octava página"** de nuestro programa.



o sea **UNID**, el costo unitario **CUNI=CCOM/UNID** y finalmente la cantidad de unidades de este material que entran en el tipo analizado **CTIP**. Más abajo una leyenda que da cuenta del total de materiales ingresado, como así también el total de tipos actualmente en estudio.

Siguiendo para abajo abordamos el menú de pie de página que se encuentra claramente dividido en dos: el de la izquierda es el empleado para localizar tanto el material como así tambien el tipo que se desea modificar o simplemente observar. De su observación se deduce que:

- **R** - Muestra el **material anterior**
- **T** - Muestra el **próximo material**
- **O** - Muestra el **tipo anterior**
- **P** - Muestra el **próximo tipo**

El submenú de la derecha que comenzaremos analizando con la opción **(K) corregir.** Ingreamos con esto a la subrutina (línea 2080). Corregir en donde se bifurcan tres posibilidades: **A alterar, F fechar, B buscar**. Digitando **B** el control retorna completamente a la página diez para tener la posibilidad de elegir otra combinación Material/Tipo y obrar allí. Optando **F** tenemos ante nosotros las herramientas necesarias para actualizar la fecha, digitando la fecha actual siempre en el formato conocido, si no se desea cambiar la fecha digitamos **N** y luego de esto el control retorna automáticamente a la **"Tabla electrónica"**.

Finalmente digitando la primera opción, o sea **A alterar**, se abre nuevamente otro abanico de posibilidades a saber: **¿qué desea modificar? C CCOM U UNID T CTIP**. Con la primera **C** puede cambiarse el costo comercio del material presente en pantalla, con el segundo **U** las unidades a que corresponde este costo (protegido contra división por cero en la línea **2230**) y por último con **T** podremos alterar las cantidades de este material que intervienen en el tipo analizado. En cualesquiera de estos últimos tres casos de alteración, el control retorna sin permiso después de cada corrección a la actualización a la matriz **Q$** (retorna a la línea **1200**) para permitir un monitoreo constante del estado de los datos a través de la tabla electrónica.

Si lo que deseamos agregar fuese modelos o tipos adicionales al cálculo, digitaremos **T** ingresando así a la rutina **agregar tipos** que comienza en la línea **2800**. Después de informársenos a modo de recordatorio muy útil la cantidad total de tipos ingresados ya y el último de ellos, y si todavía queda lugar (habrá de inmediato un mensaje si esto no es posible) se nos pedirá que ingresemos el próximo tipo rematado con la pregunta **más tipos ? (S/N)**. Con la respuesta **S** se pedirá otro tipo; con la respuesta **N** el control retornará a la tabla electrónica previa actualización de la matriz **Q$** a partir de la línea **1200**. Ingresados los tipos adicionales, es necesario completar para ellos los **CTIP** correspondientes (con opción **alterar**), lo mismo debe decirse después de agregar materiales.

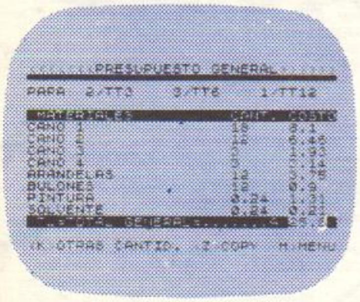
Si lo que se desea es eliminar materiales del cálculo es suficiente con hacer **CCOM=0** con lo cual este material

```
5,DO)+"    ") ( TO CR)
1211 LET Q$(5,24 TO )=(STR$ C(5,
3)+"    ") ( TO CR)
1213 NEXT 5
1215 SLOW
1216 CLS
1219 PRINT AT VE,0;"
                    "
1220 PRINT AT 19,0;" (A)MATERIA
L(T)  (K)CORREGIR";AT VE,0;"A
NTERIOR  PROXIMO  (M)MENU.";AT
VU,0;"  (O)  TIPO  (P)  ";AT
G)AGREGAR "
1225 IF K>J-U THEN LET K=U
1230 LET L=0
1500 LET X$="TABLA ELECTRONICA"
1505 GOSUB G3
1510 PRINT AT DO,0;"  PRODU
CTO: ";P$;AT CU,0;"  FEC
HA: ";F$
1512 PRINT AT 8,0;" N  MATERIAL
CCOM UNID CUNI CTIP"
1515 PRINT AT 12,0;"------------
--------------------";AT 14,6;"M
ATERIALES: ";J-U;AT 15,0;"TOTAL"
;AT 16,11;"TIPOS: ";T
1518 PRINT AT 6,0;" TIPO ANALIZA
DO: ";T$(Z)
1521 IF Z<>T THEN PRINT AT 6,23;
"(";Z;")    "
1522 IF Z=T THEN PRINT AT 6,23;"
(";Z;" FIN)"
1524 LET K=K+(INKEY$="T" AND K<J
-U)-(INKEY$="R" AND K>U)
1525 LET Z=Z+(INKEY$="P" AND Z<T
)-(INKEY$="O" AND Z>U)
1550 PRINT AT DI,0;Q$(K);AT DI,2
8;"  ";AT DI,28;(STR$ C(K,DO+Z
+DO)+"    ") ( TO CR)
1990 GOTO VAL "1518"+(INKEY$="K"
)*VAL "561"+(INKEY$="M")*VAL "24
97"+(INKEY$="G")*VAL "4482"
2000 REM FECHAR
2073 PRINT AT VE,0;"FECHA ACTUAL
 ?    (N) NO CAMBIA  "
2074 INPUT G$
2075 IF G$="N" THEN GOTO VAL "12
19"
2076 LET F$=G$
2077 PRINT AT CU,17;F$;"
"
2078 GOTO VAL "1219"
2080 REM CORREGIR
2081 PRINT AT 19,0;"
                    "
2082 PRINT AT VE,0;"(A)ALTERAR
(F)FECHAR  (V)VOLVER"
2090 GOTO VAL "2090"-(INKEY$="U"
)*VAL "871"+(INKEY$="A")*DI-(INK
EY$="F")*VAL "90"
2100 REM ALTERAR
2110 LET E=U
2150 PRINT AT VE,0;"QUE DESEA AL
TERAR ?          "
2155 PRINT AT VU,0;"(C) CCOM
(U) UNID    (T) CTIP"
2170 GOTO VAL "2170"+(INKEY$="C"
)*VAL "2"+(INKEY$="U")*VAL "130"
+(INKEY$="T")*VAL "430"
2172 REM NUEVO CCOM
2175 PRINT AT VE,0;"NUEVO COSTO
DEL MATERIAL ?    "
2176 PRINT AT VU,0;"
                    "
2180 INPUT C(K,U)
2200 GOSUB VAL "2500-320+(C(K,U)
<0)"
2220 GOTO VAL "1200"
2300 REM NUEVA UNID
2302 PRINT AT VE,0;"NUEVA CANTID
AD DE UNIDADES ?    "
2305 PRINT AT VU,0;"
                    "
2315 INPUT C(K,DO)
2330 GOSUB VAL "2500-185+(C(K,DO
)<=0)"
2350 GOTO VAL "1200"
2500 REM CALCULAR C(Y,3)
2505 LET C(K,3)=C(K,U)*C(K,DO)
2510 RETURN
2600 REM NUEVA CTIP
2605 PRINT AT VU,0;"
                    "
2610 PRINT AT VE,0;"NUEVA CANTID
AD EN TIPO ";T$(Z);" ?"
2615 INPUT C(K,DO*Z+DO)
2626 LET C(K,DO*Z+3)=C(K,DO*Z+DO
)*C(K,3)
2630 GOTO VAL "1219"
2700 REM AGREGAR MATERIALES
2702 LET L=VU
2704 SCROLL
2706 LET X$="AGREGAR MATERIALES"
2708 SCROLL
2710 GOSUB G3
2712 SCROLL
2713 PRINT "ULTIMO MATERIAL INGR
ESADO ERA:  "
2714 SCROLL
2716 SCROLL
2718 PRINT J-U;"-"+M$(J-U)
2720 SCROLL
2721 IF J>=D-U THEN PRINT "(MA
XIMA CAPACIDAD DE MATERIALES)"
2722 IF J>=D-U THEN GOTO VAL "12
18"
2723 PRINT "ENTRE A CONTINUACION
 EL PROXIMO,"
2724 SCROLL
2726 PRINT "Y NO OLVIDE, PULSE <
F> AL ULTIMO"
2728 GOTO VAL "272"
2800 REM AGREGAR TIPOS
2801 CLS
2802 LET L=0
2804 LET X$="AGREGAR TIPOS"
2806 GOSUB G3
2808 PRINT ,,"LA MAQUINA ESTA CA
RGADA CON UN TOTAL DE ";T;" TIP
OS DISTINTOS."
2810 PRINT ,,"EL ULTIMO TIPO ING
RESADO ERA:  "
2811 LET R=T
2812 PRINT ,,"TIPO ";T;" = ";T$(
T)
2813 IF T>=TP THEN GOTO VAL "700
0"
2814 IF T=R THEN PRINT ,,"INGRES
E AHORA EL SIGUIENTE:  "
```

no intervendrá en el mismo.

Por último si digitamos **M menú** el control retorna al menú principal, donde nos queda sólo la última opción que es **5 grabar**.

Pulsando **5** ingresamos a la rutina **grabar** en línea **5000**.



Ingresamos así a la subrutina **unidades a fabricar** donde se solicita las cantidades de cada tipo para las cuales se desea el **presupuesto**

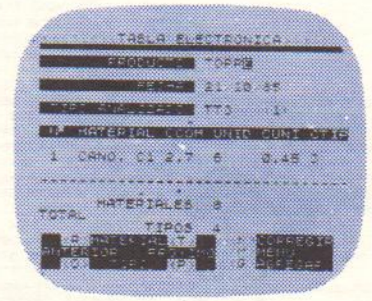
Terminado el ingreso de estas cantidades, inmediatamente la máquina nos muestra la **"página nueve"** encabezada por el detalle de referencia para el cual se efectuó el cálculo del presupuesto. Dice **para: cantidad/tipo** (cantidad de unidades/tipo considerado).

A continuación se forma una tabla con el nombre de cada material integrante escrito completo (con sus 20 caracteres) a la izquierda, la cantidad **CANT** de unidades necesarias del material considerado para la elaboración del presupuesto para encarar la fabricación del **"lote"** de modelos o tipos en estudio, y finalmente el costo total de cada una de estas cantidades en la columna **COSTO**.

Esto se repite para cada material integrante del lote de fabricación notándose que los materiales que intervienen con una cantidad **CANT** nula (cero) **no son escritos** lográndose ahorrar papel de impresión y aumentando la claridad de la presentación y su interpretación.

Esta lista es rematada por una banda final de cierre con la leyenda **total general** en australes y que resulta de la suma total de cada uno de los **costos** integrantes del presupuesto. En el submenú de pie de página podemos optar por: **K otras cantidades** o sea se podrá acceder a la configuración de otro lote de fabricación, o si se prefiere solicitar el presupuesto de tan sólo una unidad del tipo que se desee con sólo indicar **1 tipo deseado** y en el resto **O todos los demás tipos; Z copy** para obtener una copia en impresora; **M menú** con el que retornamos a nuestro "distribuidor" menú principal.

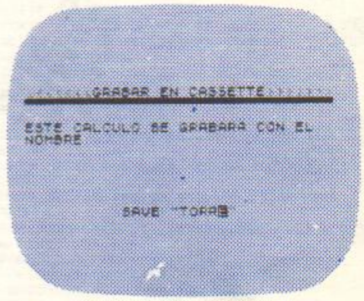
Optemos ahora por (3) **actualización** e ingresemos de esta manera a la subrutina más complicada de nuestro programa, dedicada a todo lo que concierne a correcciones, actualizaciones, alteraciones, etc.



Es precisamente esta opción la que le otorga una gran utilidad a este programa, puesto que las correcciones no implican ingresar nuevamente todos los datos y detalles no advertidos durante el proceso de carga normal. Ingresando a la subrutina **tabla electrónica** la máquina previamente (desde línea **1200** hasta **1215**) "carga" los resultados en una matriz **Q$** para obtener de esta manera las facilidades de una presentación ágil y veloz (debido a que no tiene que imprimir números) requisito éste indispensable para la rápida localización de los datos buscados en la pantalla a través de lo que denominamos **tabla electrónica** que describiremos a continuación.

La **tabla electrónica** constituye pues la **"página diez"**. De arriba para abajo aparece, nombre del producto, la fecha (del cálculo realizado), el tipo que se analiza en ese momento, luego una tabla que tiene de izquierda a derecha: el número del material, el nombre del material con su abreviatura, el costo comercio **CCOM**, el número de unidades de ese costo definiendo así la **"página once"** de nuestro programa.

Aquí sencillamente se informa al operador el nuevo nombre que adopta este programa para ser preservado en cinta de grabación. Este nombre es precisamente el valor asignado a la variable **P$** o sea el **nombre del producto**. Se podrá disponer de una grabación para cada producto analizado totalmente por separado cada uno de ellos.



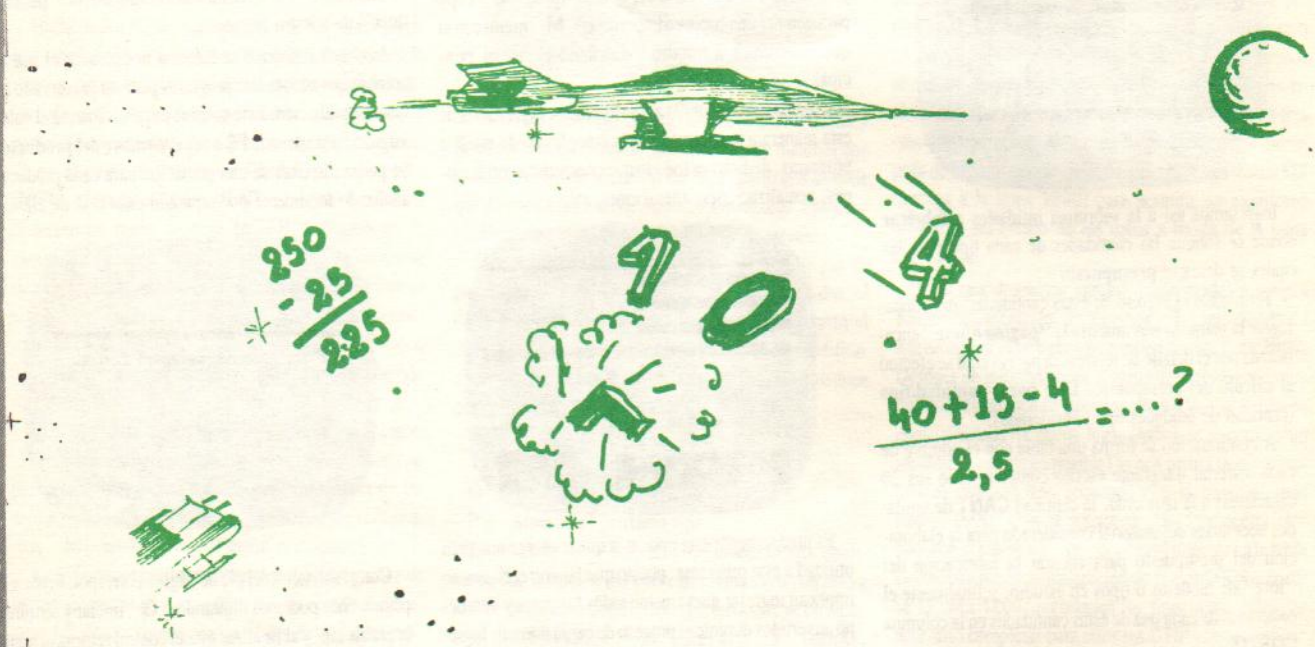
Con el submenú del pie de página correspondiente a la página once podemos digitando **G** iniciar y terminar la grabación, y al finalizar ésta el control retorna al menú principal, dando por terminado el ciclo completo del programa en su análisis operativo.

En próximos números continuaremos con el diagrama de flujo de este interesante programa y sus tablas de variables. Hasta entonces...

```
2818 LET T=T+U
2819 PRINT "TIPO ";T;" = ";
2820 INPUT T$(T)
2822 PRINT T$(T);" MAS TIPOS?(S/
N)"
2830 GOTO VAL "2830"-(INKEY$="S"
)*VAL "17"-(INKEY$="N")+VAL "161
2"
3000 REM AUTOCENTRADO
3005 PRINT AT L,0,"<<<<<<<<<<<<<<
<<<>>>>>>>>>>>>>>>>>"
3010 PRINT AT L,16-LEN X$/DO;X$
3011 IF L=VU THEN SCROLL
3012 PRINT "
"
3015 RETURN
4000 REM PRIMERA PAGINA
4002 CLS
4003 LET L=0
4005 LET X$="COSTOS DE FABRICACI
ON"
4007 GOSUB G3
4008 PRINT AT CI,0;"-HOJA ELECTR
ONICA ESPECIALIZADA-"
4009 PRINT AT DI,DO;"MEMORIA OCU
PADA=";PEEK 16396+256*PEEK 16397
-16509;" BYTES"
4010 PRINT AT 14,7;" * SOFTSECRE
T * ";AT 16,7;"(SETIEMBRE 1985)"
4012 PRINT AT VU,0;"PROHIBIDO DI
GITAR SENTENCIA  RUN"
4013 IF INKEY$="" THEN GOTO VAL
"4013"
4014 GOTO U
4015 REM MENU
4016 CLS
4017 PRINT " PRODUCTO  ";P$
4018 PRINT ,,"        FECHA   ";F$
4019 PRINT ,,"             MENU
"
4021 PRINT ,,"           <1> COSTOS
FINALES"
4025 PRINT ,,"           <2> PRESUPU
ESTAR"
4030 PRINT ,,"           <3> ACTUALI
ZAR"
4033 PRINT ,,"           <4> PRODUCT
O NUEVO"
4036 PRINT ,,"           <5> GRABAR"
4040 PRINT ,,"   <GOTO 1> EN INTE
RRUPCIONES"
4045 PRINT AT VU,0;"       (PULS
E SU OPCION)      "
4046 POKE 16510,0
4049 IF C=0 THEN PRINT AT 12,8;"
■"
4050 IF INKEY$="1" AND C=U THEN
GOTO VAL "380*(E=U)+410*(E=0)"
4060 IF INKEY$="2" AND C=U THEN
GOTO VAL "470"
4065 IF INKEY$="3" AND C=U THEN
GOTO VAL "1200+18*(B=U)"
4070 IF INKEY$="4" THEN GOTO VAL
 "11"
4072 IF INKEY$="5" THEN GOTO VAL
 "5000"
4073 IF C=0 THEN PRINT AT 12,8;"
"
4075 GOTO VAL "4049"
5000 REM GRABAR
5002 CLS
5010 LET L=0
5020 LET X$="GRABAR EN CASSETTE"
5030 GOSUB G3
5040 PRINT ,,"ESTE CALCULO SE GR
ABARA CON EL  NOMBRE "
5055 PRINT AT DI,12-LEN P$/DO;"S
AVE """;P$;""""
5060 PRINT AT VU,0;" <G> GRABA
 <M> VUELVE AL MENU "
5078 GOTO VAL "5078"+(INKEY$="G"
)+DO-(INKEY$="M")*VAL "1063"
5079 REM INICIALIZACIONES
5080 SAVE P$
5081 LET D=30
5082 LET K=PI/PI
5083 LET U=K
5084 LET VU=21
5085 LET VE=20
5086 LET Z=K
5087 LET B=PI-PI
5088 LET O=B
5089 LET C=U*(C(U,U)<>0)
5090 LET CI=5
5091 LET DI=10
5092 LET G3=3000
5096 LET CR=4
5097 LET TP=8
5100 LET CU=4
5102 LET DO=2
5104 LET E=0
5190 GOTO VAL "4000"
6000 REM AGREGAR
6001 LET B=U
6002 LET E=U
6004 PRINT AT 19,0;"
                    "
6008 PRINT AT VE,6;"QUE DESEA AG
REGAR ?";AT VU,0;"<M>MATERIALES
<T>TIPOS  <V>VOLVER"
6014 GOTO VAL "6014"-(INKEY$="M"
)*VAL "3314"-(INKEY$="T")*VAL "3
214"-(INKEY$="V")*VAL "4795"
7005 PRINT ,,"<<<ARCHIVO DE TIPO
S COMPLETO>>>"
7010 FOR P=1 TO DI*CI
7015 NEXT P
7020 GOTO VAL "1218"
```

FIN

ARITMETICA ESPACIAL



...Atención, atención: *Softi* desde nave *Columbia* llamando a base
...Me ataca una nave enemiga... ¿Qué hago?
...Aquí base para *Softi*: tranquilo... la mente vence a la fuerza...
responda la pregunta y la detendrá... Suerte...

Los tripulantes de la nave espacial son atacados por fuerzas invasoras que desprecian y destruyen a los seres que consideran inferiores. Para ello los someten a toda clase de ''torturas'' obligándolos a resolver cálculos aritméticos, en grado creciente de complejidad. Si no los pueden resolver, los aniquilan sin piedad...

Es éste un interesante ''pasatiempo'' para la familia que permitirá enfrentar a los niños (desde 8 a 99 años) con una guerra espacial en medio de cálculos matemáticos, permitiéndoles aprender juando y evaluar sus conocimientos en el tema y su agilidad mental.

INSTRUCCIONES

Al cargar el programa notaremos que se indican, en muchos casos, la/s tecla/s a presionar para obtener los símbolos programados entre corchetes [ ]. Asímismo, en caso que debamos presionar una tecla más de una vez, veremos un número al principio que indica la cantidad de veces (*ver Nota 1*).

Podremos anular la música presionando **ENTER**.

Tipo de problema a resolver:
1 = suma
2 = resta
3 = multiplicación
4 = división

Nivel de dificultad:
1 = fácil
2 = difícil
3 = más difícil
4 = muy difícil

Podremos contener al enemigo contestando correctamente y dentro del tiempo asignado, los problemas matemáticos exhibidos en el centro de la nave, con lo cual efectuaremos un disparo de rayos láser sobre el atacante.

Si damos una respuesta errónea o tardamos más tiempo del previsto, el enemigo se aproximará más a nuestra nave. Si no contestamos positivamente las preguntas suficientes, finalmente nuestra nave será destruida.

Para ganar nosotros esta guerra en el espacio de las matemáticas, deberemos contestar con éxito nada menos que las 48 preguntas de cada una de las cuatro ''vueltas'' (o sea, un total de 192 preguntas... Fácil, ¿no?), teniendo en cuenta que, a medida que vamos pasando de una ''vuelta'' a la otra, los problemas son cada vez más difíciles de resolver pues tendremos cada vez menos tiempo de respuesta.

Y ahora, ¿se animan a ser el **capitán** de la nave?

```
100 REM @@@@@@@@@@@@@@@@@@@@@@@@
110 REM @ ARITMETICA-LASER @
120 REM @ PROG.POPULAR N:8 @
135 REM @...COMMODORE 64...@
140 REM @@@@@@@@@@@@@@@@@@@@@@@@
210 POKE53281,1:POKE53280,2
220 PRINT"[CLR][4 ABAJO][5 DER.]{BLK}L[DE
R.]{ABAJO}[RED]A[DER.]{ABAJO}[CYN]S[DER.]
[ABAJO][PUR]E[DER.]{ABAJO}[GRN]R[DER.]{AB
AJO}[GRN]-[DER.]{ABAJO}[BLU]A[DER.]{ABAJO
}[YEL]R[DER.]{ABAJO}[BLK]I[DER.]{ABAJO}[R
ED]T[DER.]{ABAJO}[RED]M[DER.]{ABAJO}[CYN]
E[DER.]{ABAJO}[BLU]T[DER.]{ABAJO}[GRN]I[D
ER.]{ABAJO}[BLU]C[DER.]{ABAJO}
230 DIMM(60)
240 FORX=1TO57:READZ:M(X)=Z:NEXTX
250 FORX=0TO62:READQ:POKE832+X,Q:NEXTX
260 FORX=0TO62:READQ:POKE896+X,Q:NEXTX
270 POKE53292,6:POKE53293,7:POKE2045,13:P
OKE2046,14
280 SP=32:FL$="[COMD F]":CL$=" ":MA=254
290 PRINT"[3 ABAJO][8 DER.]PRESIONE RETUR
N PARA EMPEZAR"
300 GOSUB1290
310 PRINT"[CLR][3 ABAJO][BLU]QUE TIPO DE
PROBLEMAS DESEA?":PRINT"[3 ABAJO]1)SUMA[A
BAJO]":PRINT"2)RESTA[ABAJO]"
320 PRINT"3)MULTIPLICACION[ABAJO]":PRINT"
4)DIVISION[2 ABAJO]"
330 PRINT"[5 ABAJO]SELECCIONE UNA DE LAS
OPCIONES"
340 GETT$:IFT$=""THEN340
350 IFASC(T$)<49ORASC(T$)>52GOTO340
360 TYPE=ASC(T$)-48
370 PRINT"[CLR][3 ABAJO][3 DER.]CUAL ES S
U NIVEL?":PRINT"[3 ABAJO]1)MUY FLOJO[ABAJ
O]":PRINT"2)FLOJO[ABAJO]"
380 PRINT"3)BUENO[ABAJO]":PRINT"4)MUY BUE
NO"
390 PRINT"[5 ABAJO]INGRESE OPCION:"
400 GETS$:IFS$=""THEN400
410 IFASC(S$)<49ORASC(S$)>52THEN410
420 SKILL=ASC(S$)-48
430 POKE53281,14:POKE53280,14
440 PRINT:PRINT"[CLR][3 ABAJO][WHT]CAPITA
N:[3 ABAJO]"
460 PRINT"USTED DEBE DESTRUIR LA BASE COL
UMBIA[ABAJO] "
470 PRINT"UTILIZANDO SU MARAVILLOSO LASER
,SOLO[ABAJO] "
480 PRINT "LO PODRA HACER SI RESPONDE LOS
 SIGUIEN-";PRINT"[ABAJO]TES PROBLEMAS."
490 PRINT"[ABAJO][6 DER.]SUERTE.";PRINT"[
ABAJO][8 DER.]SOFTI.";PRINT"[2 ABAJO][DER
.]PRESIONE CUALQUIER TECLA"
500 GETX$:IFX$=""THEN500
510 SC=0:FORL=1TO4:J=1:MA=254:SP=32:
520 PRINT"[CLR]";POKE53269,PEEK(53269)AND
NOT32:POKE53258,MA:POKE53259,125
530 PRINT"[CLR][10 ABAJO][15 DER.]RONDA "
L
540 PRINT"[2 ABAJO][6 DER.]PRESIONE RETUR
N PARA CONTINUAR"
550 GOSUB1290
560 GOSUB1110:GOSUB1200
570 FORF=1TO12:FORI=1TO4:E=F
580 D=INT(RND(1)*SK*4)
590 IFD<>ETHEN610
600 T=E:E=D:D=T
610 ON TYPE GOTO620,630,640,650
620 AN=E+D:SG$="+":GOTO670
630 AN=E-D:SG$="-":GOTO670
640 AN=E*D:SG$="*":GOTO670
650 IFD=0THEN580
660 AN=E:E=E*D:SG$="/"
670 PRINT"[HOME][7 ABAJO][WHT][4 DER.]"E;
"[IZQ.]"SG$"D"="
680 M$="[3 SPC]"+STR$(E)+SG$+STR$(D)+"="
690 LM=LEN(M$):AN$=STR$(AN)
700 GETX$:IFX$<>""THEN700
710 GETX$:FORW=1TO10:NEXTW
720 IFX$=CHR$(140)THENPRINT"[CLR]":GOTO10
50
730 ENT=CNT+1
740 IF(CNT>60-(SK$4))ANDB$=""THEN880
750 IFX$=""THEN710
760 IFX$<"0"ORX$>"9"THEN700
770 X=VAL(X$):X$=STR$(X)
780 B$=B$+X$:B=VAL(B$):B$=STR$(B)
790 PRINT"[HOME][7 ABAJO]"TAB(LM+2)"[WHT]
"B$:FORW=1TO50:NEXTW
800 IFLEN(B$)=LEN(AN$)THEN870
810 FORU=1TOLEN(B$)
820 IFMID$(B$,U,1)=MID$(AN$,U,1)THEN850
830 GOSUB1210:MA=MA-8:SP=SP-1:GOSUB1410:G
OSUB1130:CNT=0
840 PRINT"[HOME][7 ABAJO][WHT][4 DER.]"E;
"[IZQ.]"SG$"D"="B$="":GOTO700
850 NEXTU
860 GOTO710
870 IFB$=AN$THEN940
880 GOSUB1210
890 SP=SP-1:MA=MA-8:GOSUB1410
900 B$="":X$="":J=J+1:CNT=0
910 IFJ>20THEN990
920 IFSP<12THEN990
930 GOSUB1130:PRINT"[HOME][2 ABAJO][WHT][
4 DER.]"E;"[IZQ.]"SG$"D"="B$="";GOTO700
940 GOSUB1230
950 B$="":SC=SC+1:PRINT"[HOME]"TAB(20)"PU
NTAJE= "SC:CNT=0:GOSUB1130:NEXTZ,F,L
960 POKE53269,PEEK(53269)ANDNOT32
970 PRINT"[CLR][6 ABAJO][5 DER.]BUEN TRAB
AJO,CAPITAN":PRINT"[3 ABAJO]SU PUNTAJE ES
 "SC
980 FORW=1TO2000:NEXTW:GOTO1040
990 PRINT"[CLR]":POKE53269,PEEK(53269)AND
NOT32
1000 FORW=1TO5:POKE53281,1:POKE53280,2:PO
KE53281,2:POKE53280,1:NEXTW
1010 FORX=1TO350:NEXTX:POKE53281,1:POKE53
280,0
1020 PRINT"[6 ABAJO][7 DER.]DESTACION
 ESPACIAL DESTRUIDA"
1030 PRINT"[3 ABAJO][12 DER.]SU PUNTAJE="
SC:FORW=1TO2000:NEXTW
1040 POKE53269,PEEK(53269)ANDNOT32
1050 PRINT"[3 ABAJO][3 DER.]DESEA UD. JUG
AR OTRA VEZ (S/N)"
1060 GETX$:IFX$<>""THEN1060
1070 GETX$:IFX$=""THEN1070
1080 IFX$="S"THENSP=32:MA=254:GOTO290
1090 IFX$<>"N"THEN1060
1100 POKE53281,6:POKE53280,14:PRINT"[SHFT
 Z][CLR]":END
1110 PRINT"[CLR]":POKE53281,0:POKE53280,0
1120 FORX=1TO25:N=INT(RND(1)*(999)+1024):
POKEN,46:POKEN+54272,14:NEXTX
1130 PRINT"[BLK][HOME][4 ABAJO][4 DER.][R
VS ON][COMD V][9 SPC][COMD C][RVS OFF]"
1140 PRINT"[RED][3 DER.][RVS ON][COMD V][
9 SPC][COMD C][RVS OFF]"
1150 PRINT"[SHFT v][2 DER.][RVS ON] [SHFT
 W] [SHFT W] [SHFT W][2 SPC][RVS OFF]"
1160 PRINT"[RED][3 DER.][RVS ON][COMD F][
11 SPC][COMD D][RVS OFF]"
1170 PRINT"[SHFT a][4 DER.][RVS ON][COMD
F][9 SPC][COMD D][RVS OFF]"
1180 PRINT"[8 DER.][PUR][COMD L]"
1190 PRINT"[9 DER.][PUR][COMD L](":RETURN
1200 POKE53269,PEEK(53269)OR32:POKE53258,
MA:POKE53259,125:RETURN
1210 FORA=1TO20:POKE54296,15:FORT=1TO3:NE
XT:POKE54296,0:FORT=1TO3:NEXT
1220 NEXT:RETURN
1230 LZ$="":FORK=1TOSP-11:LZ$=LZ$+FL$:NEX
TK
1240 BL$="":FORK=1TOSP-11:BL$=BL$+CL$:NEX
TK
1250 POKE53296,15:POKE54283,129:POKE54284
,16:POKE54285,240
1260 POKE54280,200:POKE54279,243
1270 FORW=1TO20:PRINT"[HOME][10 ABAJO][WH
T]"TAB(10)LZ$:NEXTW
1280 PRINT"[HOME][10 ABAJO]"TAB(10)BL$:PO
KE54283,0:POKE54284,0:GOSUB1430:RETURN
1290 FORLL=54272TO54296:POKELL,0:NEXT:POK
E54296,15
1300 POKE54284,124:POKE54285,248:FORA=1TO
57STEP3
1310 POKE54280,M(A):POKE54279,M(A+1):POKE
54283,17:FORT=1TOM(A+2):NEXT
1320 POKE54283,16:FORT=1TO25:NEXT
1330 GETW$:IFW$=CHR$(13)THENRETURN
1340 NEXTA:GOTO1300
1350 DATA 18,209,100,18,209,50,18,209,100
,25,30,400
1360 DATA 37,162,450,33,135,50,31,165,50,
28,89,50
1370 DATA 50,60,400,37,162,450,33,135,100
,31,165,50
1380 DATA 28,49,50,50,60,400,37,162,450,3
3,135,50
1390 DATA 31,165,100,33,135,100,28,49,550
1400 REM
1410 POKE53269,PEEK(53269)ANDNOT32:POKE53
258,MA:POKE53259,125
1420 POKE53269,PEEK(53269)OR32:RETURN
1430 POKE53269,PEEK(53269)ANDNOT32
1440 POKE53260,MA:POKE53261,125
1450 FORWW=1TO5:POKE53269,PEEK(53269)OR64
:FORNN=1TO10:NEXTNN
1460 POKE53269,PEEK(53269)ANDNOT64
1470 FORNN=1TO10:NEXT:NEXT:GOSUB1200:RETU
RN
1480 DATA 0,0,0,0,0,0,0,0
1490 DATA 0,0,0,7,255,224,15,255,240
1500 DATA 14,60,112,14,60,112,14,60,112
1510 DATA 15,255,240,15,255,240,15,129,24
0
1520 DATA 15,129,240,15,129,240,15,129,24
0
1530 DATA 15,129,240,0,0,0,0,0,0
1531 DATA 0,0,0,0,0,0,0,0,0
1540 DATA 0,0,6,33,12,4,49,0,96
1550 DATA 0,0,6,33,12,4,49,0,96
1560 DATA 0,0,64,0,0,0,0,8,0
1570 DATA 16,24,0,56,0,4,8,0,8
1580 DATA 0,2,16,0,6,0,0,2,0
1590 DATA 0,0,0,48,16,32,16,112,48
1600 DATA 16,32,0,0,0,0,4,2,8
1610 DATA 24,12,8,1,128,12,0,0,0
```

Nota (1): Esta novedad la incorporamos a partir del presente número de nuestra revista ***Programación Popular*** *para facilitar la carga de los programas, debido a la gran cantidad de símbolos de la* ***C-64****, y a que su impresión directa resulta a veces confusa y la difícil interpretación. Esperamos responder así a la inquietud de muchos amigos lectores y facilitarles el tipeo de nuestros programas, a los que informaos que seguimos trabajando para mejorar más aún la presentación en la revista de los programas para la* ***C-64*** *y la seguridad en la carga. Próximamente habrá sorpresas al respecto...*

ESTRUCTURA DEL PROGRAMA

100/300	Encabezamiento, inicialización y títulos en pantalla
310/550	Opciones e instrucciones
560/690	Determinar problema y mostrarlo en pantalla
700/950	Ingresar respuesta, evaluar y seguir curso de acción en consecuencia
960/1100	Fin de juego. Opción de recomenzar
1110/1190	Pantalla del espacio y nave
1200/1280	S/R de sonido y acción
1290/1390	S/R de música
1400/1470	S/R de movimiento nave enemiga
1480/1610	Datos sprite nave enemiga

Drean commodore
CZERWENY Spectrum
Joysticks - Interfaces - Impresoras - Cassettes - Diskettes - Lápiz óptico
PAPELSHOP
COMPUTACION
PAPELERA - LIBRERA
TARJETAS DE CREDITO
LAPIZ OPTICO
ENVIOS AL INTERIOR
AV. PUEYRREDON 658 (1032)
CAP. FED. T.E. 88-8582
ref. 20

VENENO



Este programa ganador del amigo **Rodolfo** bien podría llamarse ''Aerosol'' (para no hacer marcas), en lugar de ''Veneno'', porque ése es el tema de este divertido y original entretenimiento: el aerosol es el único elemento con que contamos para defendernos del encarnizado y molesto ataque de moscas y otros bichos mortíferos (abejas, avispas, pulgas, pulpos negros y otros animalejos por el estilo).

Una vez cargado y **SAVE**'ado el programa (por si las moscas...) lo haremos correr y utilizaremos las teclas:

[E] para dirigir hacia arriba el ''veneno'' del aerosol

[X] para dirigirlo hacia abajo

[.] o **[Q]** para lanzar el veneno del aerosol

[P] para parar el programa (los insectos se seguirán moviendo, pero sin picarnos) y luego cualquier tecla para continuar.

Cuidado!!!... Si te pica algún bicho perdés un aerosol (hay 5 en total) y si perdés todos los aerosoles... te comen los bichos!!!

Al matar a todos los insectos de la pantalla, aparecerán otros, y luego otros... y cada vez con mayor velocidad!!!

Por último, Rodolfo nos advierte que puede suceder que al aumentar la velocidad de los insectos el ''disparo' del aerosol pierda precisión, en cuyo caso habrá que repetirlo. Esto se debe a que se produce una descoordinación entre el movimiento de los ''sprites'' de los insectos y el **COINC** de la línea **750**.

¿Y qué esperamos para ''morirnos'' de risa???...

```
100 CALL PRESENTACION
110 RESTORE 280
120 CALL CLEAR :: CALL MAGNI
FY(3)
130 CALL CHAR(36,"001818183C
7E7E7E7E7E7E7E7E7E7E00"&RPT$
("0",16*2))
140 CALL CHAR(40,"0B20952A05
000000"&RPT$("0",16*3))
150 CALL CHAR(44,"0000000000
00FFFF")
160 CALL CHAR(92,"0003056B7E
FE7E24"&RPT$("0",16*3))
170 CALL CHAR(96,"0305C9FEAB
7E1400"&RPT$("0",16*3))
180 CALL CHAR(100,"0006CEFC3
F100000"&RPT$("0",16*3))
190 CALL CHAR(104,"003E083DF
F143E00"&RPT$("0",16*3))
200 CALL CHAR(108,"000E1FFF1
F0E0000"&RPT$("0",16*3))
210 CALL CHAR(112,"060A126A1
20A0600"&RPT$("0",16*3))
220 CALL CHAR(116,"0000FBA80
FA8F800"&RPT$("0",16*3))
230 CALL CHAR(120,"001C3E252
5538102"&RPT$("0",16*3))
240 MALAS=6 :: PUNT=0 :: VID
A=5 :: VEL=10
250 CALL HCHAR(3,1,44,32)::
CALL HCHAR(20,1,44,32)
260 DISPLAY AT(2,3):"VIDAS:"
 :: CALL HCHAR(1,11,36,VIDA)
:: CALL HCHAR(2,11,37,VI
DA):: DISPLAY AT(2,16):"PUNT
OS:"
270 CALL SCREEN(4):: RANDOMI
ZE
280 DATA 92,MOSCA,96,ABEJA,1
00,AVISPA,104,ATTAKING,108,P
ULGA,112,TRINGULIN,116,D
ENTUDO,120,PULPO NEGRO,9,RO
290 VEL=VEL+3 :: CALL DELSPR
ITE(ALL):: CALL HCHAR(23,32,
41):: READ NU,VI$ :: IF
NU=9 THEN RESTORE 280 :: GOT
O 290
300 DISPLAY AT(23,13):VI$
310 CALL SPRITE(#1,NU,9,28,2
00,0,-(INT(RND*VEL)+20))
320 CALL SPRITE(#2,NU,9,45,2
00,0,-(INT(RND*VEL)+20))
330 CALL SPRITE(#3,NU,9,72,2
00,0,-(INT(RND*VEL)+20))
340 CALL SPRITE(#4,NU,9,99,2
00,0,-(INT(RND*VEL)+20))
350 CALL SPRITE(#5,NU,9,126,
200,0,-(INT(RND*VEL)+20))
360 CALL SPRITE(#6,NU,9,150,
200,0,-(INT(RND*VEL)+20))
370 CALL SPRITE(#7,36,2,90,5
)
380 CALL COINC(ALL,C):: IF C
=-1 THEN CALL MATADO :: GOTO
 310
390 X=0 :: CALL MOTION(#7,0,
0):: CALL KEY(0,K,S):: IF S=
0 THEN 380
400 IF K=ASC(".")OR K=ASC("Q
")THEN CALL DISPARO :: CALL
GCHAR(23,32,LL):: IF LL=
39 THEN 290
410 IF K=ASC("E")THEN X=-30
420 IF K=ASC("X")THEN X=30
430 IF K=ASC("P")THEN CALL P
AUSA
440 CALL MOTION(#7,X,0):: CA
LL POSITION(#7,W,Q):: IF W>=
140 THEN CALL LOCATE(#7,
140,5)
450 IF W<=28 THEN CALL LOCAT
E(#7,28,5):: GOTO 380
460 GOTO 380
470 SUB PAUSA :: DISPLAY AT(
24,13):"PAUSA"
480 CALL KEY(0,K,S):: IF S=0
 THEN 480
490 DISPLAY AT(24,7):" " ::
SUBEND
500 SUB MATADO
510 FOR I=1 TO 7 :: CALL MOT
ION(#I,0,0):: NEXT I
520 CALL SOUND(800,201,2)::
CALL DELSPRITE(#7)
530 VIDA=VIDA-1 :: FOR I=1 T
O 100 :: NEXT I
540 CALL HCHAR(2,15+VIDA+1,9
3):: CALL HCHAR(1,15+VIDA+1,
93):: CALL HCHAR(23,1,39
)
550 FOR UYT=1 TO 50 :: NEXT
UYT :: CALL SOUND(34,712,3)
560 CALL DELSPRITE(ALL)
570 IF VIDA>-5 THEN 640 ELSE
 580
580 REM
590 DISPLAY AT(11,11):"OTRO
JUEGO S O N)"
600 CALL KEY(0,K,S):: IF S=0
 THEN 600
610 IF K=ASC("S")THEN RUN
620 IF K=ASC("N")THEN END
630 GOTO 600
640 SUBEND
650 SUB DISPARO
660 CALL GCHAR(23,1,HY):: CA
LL HCHAR(23,1,42):: IF HY=39
 THEN MI=0 :: CALL MOTIO
N(#7,0,0)
670 CALL POSITION(#7,W,Q)
680 IF W>20 AND W<36 THEN MU
N=1
690 IF W>37 AND W<52 THEN MU
```

```
N=2
700 IF W>53 AND W<80 THEN MU
N=3
710 IF W>81 AND W<109 THEN M
UN=4
720 IF W>110 AND W<134 THEN
MUN=5
730 IF W>135 AND W<160 THEN
MUN=6
740 CALL SPRITE(#8,40,2,W-3,
8,0,50)
750 CALL COINC(#8,#MUN,12,C)
:: IF C=-1 THEN 780
760 CALL POSITION(#8,W1,Q1):
: IF Q1>=200 THEN CALL DELSP
RITE(#8):: GOTO 840
770 GOTO 750
780 CALL DELSPRITE(#MUN,#8):
: CALL SOUND(120,110,2)
790 PUNTOS=PUNTOS+50
800 MI=MI-1 :: IF MI=-6 THEN
 JK=1 ELSE JK=0
810 DISPLAY AT(2,22)SIZE(-7)
:PUNTOS
820 IF JK=1 THEN 830 ELSE 84
0
830 CALL HCHAR(23,32,39):: M
I=0
840 SUBEND
850 SUB PRESENTACION
860 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL COLOR(13,2,
16,14,2,16):: CALL SCREE
N(16)
870 CALL CHAR(128,"0103070F1
F3F7FFF",129,"FFFEFCF8F0E0C0
80",130,"80C0E0F0F8FCFEF
F",131,"FF7F3F1F0F070301")
880 DISPLAY AT(3,1)SIZE(10):
"PRODUCCION" :: DISPLAY AT(3
,18):"MIRANDA"
890 CALL CHAR(136,"FFFFFFFFF
FFFFFFF")
900 DISPLAY AT(12,11):"VENEN
O"
910 CALL VCHAR(3,15,136,4)::
 CALL VCHAR(3,16,136,4):: CA
LL HCHAR(6,14,136,4):: C
ALL HCHAR(7,13,136,6):: CALL
 HCHAR(8,12,136,8)
920 CALL HCHAR(9,12,136,8)::
 CALL HCHAR(10,12,136,8):: C
ALL HCHAR(11,12,136,8)::
 CALL VCHAR(12,12,136,7):: C
ALL VDHAR(13,13,136,6)
930 CALL VCHAR(13,14,136,6):
: CALL VCHAR(13,15,136,6)::
CALL VCHAR(13,16,136,6):
: CALL VCHAR(13,17,136,6)::
CALL VCHAR(13,18,136,6)
940 CALL VCHAR(12,19,136,7):
: CALL HCHAR(5,14,128):: CAL
L HCHAR(6,13,128):: CALL
 HCHAR(7,12,128):: CALL HCHA
R(5,17,130)
950 CALL HCHAR(6,18,130):: C
ALL HCHAR(7,19,130):: CALL V
CHAR(20,5,136,3):: CALL
VCHAR(20,9,136,3):: CALL HCH
AR(24,7,136)
960 CALL HCHAR(23,5,131):: C
ALL HCHAR(23,6,130):: CALL H
CHAR(24,6,131):: CALL HC
HAR(23,8,128):: CALL HCHAR(2
3,9,129):: CALL HCHAR(24,8,1
29)
970 CALL VCHAR(20,11,136,5):
: CALL HCHAR(20,12,136,2)::
CALL HCHAR(22,12,136)::
CALL HCHAR(24,12,136,2):: CA
LL VCHAR(20,15,136,5)
980 CALL VCHAR(20,18,136,5):
: CALL HCHAR(20,16,130):: CA
LL HCHAR(21,16,131):: CA
LL HCHAR(21,17,130):: CALL H
CHAR(22,17,131)
990 CALL VCHAR(20,20,136,5):
: CALL HCHAR(20,21,136,2)::
CALL HCHAR(22,21,136)::
CALL HCHAR(24,21,136,2):: CA
LL VCHAR(20,24,136,5)
1000 CALL VCHAR(20,27,136,5)
:: CALL HCHAR(20,25,130):: C
ALL HCHAR(21,25,131):: C
ALL HCHAR(21,26,130):: CALL
HCHAR(22,26,131)
1010 CALL VCHAR(20,29,136,5)
:: CALL VCHAR(20,31,136,5)::
 CALL HCHAR(20,30,136)::
 CALL HCHAR(24,30,136)
1020 DATA 1,330,1,392,1,392,
1,392,2,349,1,330,1,392,1,39
2,1,20000,1,196
1030 DATA 1,20000,1,392,1,44
0,1,349,1,440,2,523,1,440,2,
392,1,196,1,20000
1040 DATA 1,392,1,440,1,349,
1,440,1,523,1,494,1,440,1,39
2
1050 DATA 1,440,1,392,1,330,
1,262,1,339,1,392,1,392,1,39
2
1060 DATA 2,440,1,494,2,523,
1,196,1,196,1,0
1070 RESTORE 1020
1080 D=150 :: READ GU,F :: I
F F=0 THEN 1110
1090 CALL SOUND(D*GU,F,5)
1100 GOTO 1080
1110 SUBEND
```


unicomp s.r.l.

SISTEMAS DE COMPUTACION

Distribuidores
Microdigital
Latindata
Pelikan (Cintas para impresoras)
VISICOMP (Monitores B y N, Verde)
Drean Commodore
Czerweny

Diskettes
Maxell
Datalife
FUJI
SKC

Bibliografía
Textos
Revistas (Nacionales e Importadas)

Cassettes
(Programas)
TK 85
Microsoft TK 90
TK 2000
Spectrum
Commodore
Sinclair 2068

Impresoras
OKIDATA
Alphacom 32
Seikosha G-100

Accesorios
Joysticks
Interface de grabador para Commodore 64.
Amplificador de sonido para Spectrum
Reset para Commodore
Interface de joysticks para Spectrum
Fundas para computadoras
Expansiones de memoria

MONROE 4502 esq. LUGONES Tel. 51-2754/2659

ref. 13

Spectrum TK 90X TS 2068

MONITOR - LM - 1

Softi nos brinda una invalorable ayuda para simplificarnos y asegurarnos una correcta carga de los programas y rutinas en *LM* (*lenguaje de máquina*), y listados en nuestra revista Programación Popular.

Uno de los principales problemas con que nos encontramos al *querer cargar desde la revista un programa o rutina en lenguaje de máquina*, es que resulta extremadamente dificultoso el tipeo debido a que sólo podemos limitarnos a "copiar" caracteres que no tienen ningún significado aparente.

Además, en tanto más largo sea el programa, más tediosa resulta su carga, con lo cual el riesgo de errores de tipeo se incrementa notablemente.

Como si esto fuera poco, y dado que no tenemos ninguna forma de validar la corrección del programa una vez incorporado en nuestra máquina, a no ser un "punteo" con otra persona o con nosotros mismos, los cuales no ofrecen demasiadas garantías, finalmente tenemos que correr el riesgo de transferirle el control al LM a través del respectivo **RANDOMIZE USR**... y, con dedos temblorosos por el miedo, apretar la fatídica tecla **ENTER**.

Si todo fue bien: fantástico, maravilloso, etc. y el programa en LM hará lo suyo!!!

¿Pero si no fuera así???.. ¿Si hemos deslizado un solo error en toda la (a veces larga) lista de números ingresados? Sabemos que entonces, lo más probable es que el programa *se "cuelgue"*, no pudiendo retomar nosotros nunca más el control del teclado a través del Intérprete Basic, o bien, lo cual no es menos grave, que termine *"autoborrándose"* o *"autodestruyéndose"*, mientras nosotros vemos con desesperación cómo van apareciendo en la maldita pantalla una serie de extraños símbolos y colores que nos indican que, lamentablemente, lo temido se produjo...

Por todo esto, y sonando aún en nuestros oídos la gruesa serie de maldiciones lanzadas por muchos de nuestros "amigos" lectores, es que nuestro equipo de programación, con **Softi** a la CPU (quiero decir, a la "cabeza") nos hemos ocupado de brindar una efectiva solución a este problema.

Fruto de este trabajo es la rutina que presentamos a continuación, que nuestros lectores podrán utilizar para cargar los **programas en LM** para **Spectrum-TK00X-TS2068** que, desde aquí en más, salgan listados en nuestra revista **Programación Popular**. Les adelantamos que estamos trabajando para hacer lo propio con las otras máquinas, cuyos respectivos **Monitores** iremos publicando oportunamente...

```
1000 RESTORE : READ P$,DI,DF,BYT
ES,INICIO: CLEAR DI-1: LET T$="
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                              @
@  M O N I T O R - L M - 1     @
@ CARGADOR DE PROGRAMAS EN LM  @
@ REV.PROGRAMACION POPULAR-N° 8 @
@     SPECTRUM-TK90X-TS2068    @
@                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
"

1010 BORDER 0: PAPER 0: INK 9: C
LS : RESTORE : READ P$,DI,DF,BYT
ES,INICIO
1020 LET B$="
                                "
1030 LET G$="--------------------
--------------"
1040 LET T$=T$(2 TO )
1050 LET CL=0
1060 PRINT AT 3,0; PAPER 1;T$
1070 LET L=1: GO SUB 4000
1080 IF DF-DI+1<>BYTES THEN LET
E$="VERIFICAR DIRECCIONES/BYTES"
: GO TO 5020
1090 LET DA=DI
1100 LET LINEAS=1+INT (BYTES/32)
+((BYTES/32)<>INT (BYTES/32))
2000 REM Control de lineas y car
gs LM
2010 FOR L=2 TO LINEAS
2020 LET DANT=DA
2030 GO SUB 4000
2040 LET CL=L
2050 READ D$,CONTROL
2060 LET CARACT=LEN D$: IF DANT-
DI+CARACT/2>BYTES THEN LET E$="
BYTES="+STR$ (DANT-DI+CARACT/2)+
"   PREVISTOS="+STR$ BYTES: GO T
O 5000
2070 IF CARACT>64 OR CARACT<2 OR
 L<LINEAS AND CARACT<64 THEN LET
 E$="CARACTERES LINEA="+STR$ CAR
ACT+"     (OK=64)": GO TO 5000
2080 IF CONTROL<0 THEN LET E$="F
ALTA TOTAL DE CONTROL": GO TO 50
00
2090 FOR B=1 TO CARACT-1 STEP 2
2100 IF D$(B)<"0" OR D$(B)>"F" T
HEN LET E$="  """+D$(B)+"""  = CAR
ACTER INVALIDO": GO TO 5000
2110 IF D$(B+1)<"0" OR D$(B+1)>"
F" THEN LET E$="  """+D$(B+1)+"""
 = CARACTER INVALIDO": GO TO 500
0
2120 LET CODIGO1=16*(CODE D$(B)-
48+(D$(B)="9")-55*(D$(B)>"9"))
2130 LET CODIGO2=CODE D$(B+1)-48
+(D$(B+1)="9")-55*(D$(B+1)>"9")
2140 LET CODIGO=CODIGO1+CODIGO2
2150 POKE DA,CODIGO
2160 LET P=CODIGO*(B+1)/2
2170 LET CL=CL+P
2180 LET DA=DA+1
2190 NEXT B
2200 GO SUB 3000
2210 NEXT L
2220 IF BYTES<>DA-DI THEN STOP
2230 GO TO 6010
3000 REM Control de totales por
 linea
3010 IF CL<>CONTROL THEN LET E$=
"DIFIERE TOTAL CONTROL C/DATOS"
 GO TO 5000
3020 RETURN
4000 REM S/r etiqueta verificaci
on  linea
4010 PRINT PAPER 1;AT 3,0;T$; FL
ASH 1;AT 16,5; PAPER 6;"VERIFICA
NDO LINEA "; FLASH 0;" ";L;" "
4020 RETURN
5000 REM Listar-tipo error y lin
ea errones
5010 LET DA=DANT
5020 PRINT AT 14,0; PAPER 1;B$;B
$;B$;B$;B$
5030 PRINT AT 15,1; PAPER 7;"ERR
OR EN DATA EN LA LINEA     ";AT
15,27;L
5040 PRINT AT 16,1; PAPER 7;B$(3
 TO );AT 16,1;E$
5050 PRINT AT 18,1; INK 6;"BREAK
+EDIT+CORREGIR"; PAPER 6; INK 9;
"(GOTO 8000)".
5060 BEEP 1,-10
5070 RESTORE L
5080 PAPER 1: LIST L
5090 STOP
6000 REM Fin proceso OK/grabacio
n programa LM
6010 BORDER 0: PAPER 0: INK 9: C
LS : PRINT INK 6;T$
6020 PRINT AT 10,0; PAPER 1; FLA
SH 1;"CARGA LM OK!! - GRABAR PRO
GRAMAS"
6030 FOR A=1 TO 20: BEEP .03,A:
NEXT A
6050 INPUT INK 6;" CONECTAR GRAB
ADOR PARA GRABAR    """;(P$);"""
Y BYTES (CODE) "; LINE M$
6060 PRINT AT 10,0;B$;AT 10,1; P
APER 1;" GRABANDO PROGRAMA: "; F
LASH 1;P$
6070 SAVE P$ LINE 7000: PRINT AT
 10,0;B$;AT 10,1;"GRABANDO BYTES
: "; FLASH 1;P$;" (CODE)": SAVE
P$CODE DI,BYTES
6080 PRINT AT 10,0;B$
6090 INPUT FLASH 1; INK 6;"REBOB
INAR PARA VERIFICAR"; LINE M$
6100 PRINT AT 10,1; PAPER 1;"VER
IFICANDO PROGRAMA "; FLASH 1;P$:
 VERIFY P$
6110 PRINT AT 10,0;B$;AT 10,0;"V
ERIFICANDO BYTES "; FLASH 1;P$:
VERIFY P$CODE DI,BYTES
6120 PRINT AT 10,0; FLASH 1;"
   FIN DEL PROCESO OK !!!      "
6130 STOP
7000 REM lectura automatica prog
ramas
7010 BORDER 0: PAPER 0: INK 9: C
LEAR DI-1
7020 RESTORE : READ P$,DI,DF,BYT
ES,INICIO: PRINT AT 1,1; PAPER 1
;" Revista Programacion Popular
";AT 3,9;" presenta: ";AT 11,10
; PAPER 6;P$
7030 PLOT 72,103: DRAW 96,0: DRA
W 0,-40: DRAW -96,0: DRAW 0,40
7040 PRINT AT 18,4; INK 4; FLASH
 1;"un momento por favor!"
7050 LOAD P$CODE
7060 RANDOMIZE USR INICIO
8000 REM reinicio del proceso po
r errores
8010 PAPER 0: CLS : LET DA=DANT:
 IF L>1 THEN GO TO 2030
```

INSTRUCCIONES A SEGUIR

1. Cargar el presente programa **MonitorLM1** y **SAVE'ARLO** en casete.

2. Ingresar los datos del programa/rutina en **LM** que haya aparecido en nuestra revista **P.P.** y que querramos incorporar a nuestra biblioteca de programas.

3. Los programas en **LM** que se publiquen a partir de este número en nuestra revista **Programación Popular** para estas máquinas (**Spectrum, TK90X** y **TS2068**) se listarán a través de **DATA**'s.

4. El **DATA** de la línea **1** contendrá el encabezamiento del **LM** con la siguiente información, separada por "comas":

- *Nombre* del programa en **LM**
- *Dirección inicial* en la memoria
- *Dirección final* en la memoria
- Cantidad de *bytes* del programa en **LM**
- *Dirección* de memoria en la cual se *iniciará la ejecución* del programa (para direccionar el **RAND USR**).

5. Las líneas número **2** y subsiguientes contendrán, cada una de ellas, la siguiente información, separada por una *"coma"*:

- *32 bytes*, a través de 64 caracteres **hexadecimales** (alfanuméricos). La última línea contendrá la cantidad de caracteres necesarios para completar los bytes totales del programa
- *Total de control* de la línea (numérico), formado por la suma de los productos de cada caracter hexa por su posición relativa dentro de la línea, más el número de la línea (**DATA**) respectivo.

6. Lo primero que debemos hacer, con el **Monitor** cargado en nuestra computadora, es ingresar cuidadosamente la línea **1** del programa en **LM** de que se trate y ya podremos hacer **RUN**. De esta forma, el **Monitor** tendrá incorporados la totalidad de los parámetros para poder validar la carga posterior que realicemos del **LM** Veremos que el programa se detiene con **OUT OF DATA**, significando que fue a leer (**READ**) el **DATA** siguiente y no lo encontró.

7. De aquí en más, es todo muy sencillo, se trata de ir incorporado la totalidad de los **DATA**'s que compongan el **LM**, admitiendo nuestro **Monitor** dos variantes básicas:

a. Cargar la *totalidad* de los **DATA**'s, y luego hacer **RUN**, tras lo cual se encargará de chequearnos cada línea y permitirnos su corrección si detectara errores (retomando el chequeo con **GOTO 8000**).

b. Cargar *grupos* de **DATA**'s y hacer **GOTO 8000** con cada uno de ellos, para irlos validando parcialmente (y corrigiéndolos de ser necesario).

8. El **Monitor** controlará la *validez* de los caracteres hexa y la *consistencia* general de cada línea, para finalmente comparar el *total de control* ingresado con el calculado en base a los caracteres tipeados.

9. En general, el **Monitor** nos informará el *tipo de error* detectado y la *línea errónea* (salvo cuando falten líneas de **DATA**'s) que parará con **OUT OF DATA**, según vimos antes). También nos facilitará el trabajo de corrección de la línea errónea haciéndonos un **LIST** de la misma, la que veremos en pantalla.

10. Para hacer las **correcciones**, sólo deberemos *detener el programa* (con **BREAK**), *editar* la línea errónea (con **CAPS SHIFT + EDIT**), *corregir* el/los errores de la línea, *reingresar* la línea corregida (con **ENTER**, simplemente) y, finalmente, hacer **GOTO 8000**, para *retomar el proceso*. Una advertencia, debemos acostumbrarnos a **GOTO 8000** en lugar de **RUN**, porque sino comenzaría todo el proceso de chequeo nuevamente a partir de la línea **1**. De todos modos, si nos equivocamos, lo único que perderemos es... tiempo.

11. Aclaramos que *no es necesario hacer toda la carga de una sola vez*; podremos tomarnos todos los descansos que querramos y hacer todas las interrupciones que necesitemos, con tal que nos acordemos de grabar el **Monitor** con los **DATA**'s tipeados hasta allí, el cual leeremos nuevamente cuando dispongamos continuar la carga.

12. Una vez que hayamos cargado correctamente todo el listado de **DATA**'s del **LM**, nuestro **Monitor** nos felicitará con el ansiado **"Carga LM OK!! - Grabar programas"**, tras lo cual simplemente deberemos disponer del grabador conectado para **SAVE**'ar el programa **Monitor** con más el archivo en **Bytes** del flamante programa en **LM**, de lo cual se encargará automáticamente el **Monitor**. Luego de verificar el programa cargado podremos (por fin!!!) utilizar el programa **LM**, ya sea haciendo **LOAD " "** (o el nombre del programa en **LM**), o bien, si la ansiedad nos domina, haciendo **GOTO 7060**, puesto que todavía tendremos en la memoria de nuestra máquina el programa en **LM** intacto.

13. En lo sucesivo, cada vez que querramos disponer del programa en **LM** sólo deberemos leer el programa antedicho grabado en el casete, el cual se encargará automáticamente de incorporarnos el código máquina en la memoria y transferirá el control a dicho **LM**.

14. Por último, en caso que usemos frecuentemente este programa en **LM** y querramos acortar el tiempo de lectura del programa cargador en Basic (que no es otro que nuestro bienamado **Monitor**), podremos hacer lo siguiente:

a. Cargar el programa **LM** en memoria.

b. Hacer **NEW** y **ENTER**.

c. Retipear sólo la línea **1** de **DATA**'s y las líneas **1000** y **7020** a **8010** (ambas inclusive).

d. Hacer **RUN** y grabar en casete la **versión definitiva** del programa en **LM**.

EXPRIMIENDO LA C-64

Continuamos viendo nueva ideas para explotar al máximo la programación de nuestra C-64

PROTECCION DE PROGRAMAS

1. Para dificultar el listado de un programa podremos hacer:

```
10 REM [SHIFT L]
```

siendo la línea **10** la primera de nuestro programa.

Al querer listarlo algún "*extraño curioso*" se verá imposibilitado de hacerlo.

2. La trampita anterior falla cuando listemos **NO** desde la primera línea del programa. Pero esto tiene fácil solución:

```
POKE 808,225 (No permite listar ni interrumpir el proceso)
POKE 775,200 (No permite listar)
```

LOCACION DE PROGRAMAS

Veremos ahora cómo determinar la ubicación en memoria de un programa cargado desde **Disco**:

1. **Comienzo del programa**

```
10 OPEN 8,8,8,"[nombre],P,R"
20 GET  8,A$,B$
30 PRINT "COMIENZA EN" ASC(A$+CHR$(0))+256 ASC(B$+CHR$(0))
40 CLOSE 8: END
```

2. **Fin del programa**

```
VERIFY "[nombre]"8,1
PRINT PEEK (175) 256+PEEK(174)
```

COLORES AL AZAR

Cuando necesitemos definir aleatoriamente colores en un programa, podremos hacer:

```
CHR$(149+RND(8) i1)
```

UTILIZANDO LA IMPRESORA

En caso que dispongamos de impresora para nuestra C-64, damos aquí un par de consejos útiles.

1. **Listando programas "a doble espacio".**

Podremos hacer:

```
OPEN128,2:CMD 128:LIST
```

2. **Imprimiendo la pantalla**

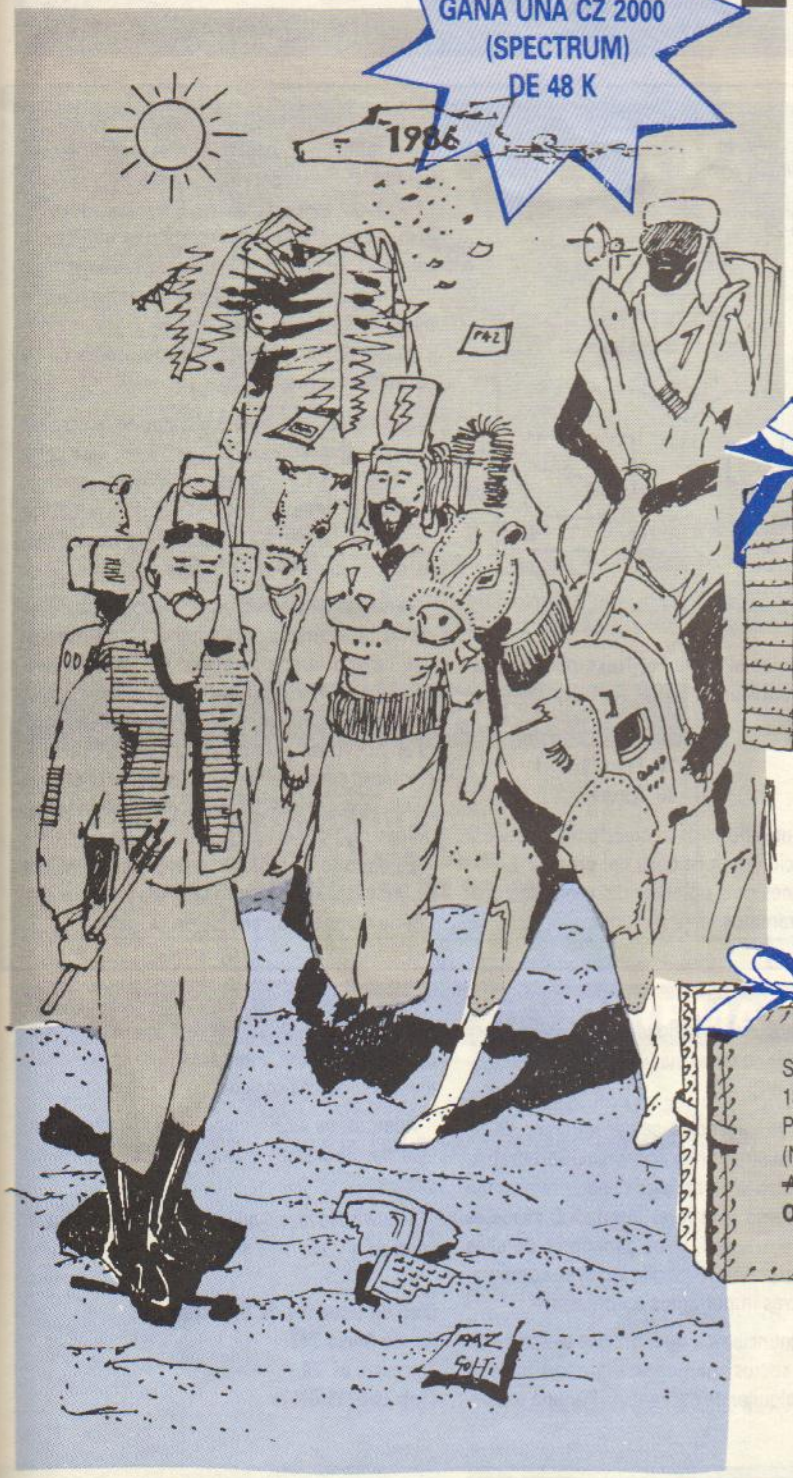
Si disponemos del lenguaje **Basic-Simmons**, podremos copiar en la impresora la pantalla de nuestro IV, utilizando la función

```
HRDCPY
```

Para estas fiestas:

LO MEJOR... LES DESEA

GANA UNA CZ 2000 (SPECTRUM) DE 48 K



programación popular

Tomo I P.P.
(revistas I al 6 encuadernadas)
₳ 12,60
Oferta ₳ 9,90

Casete Tomo I P.P.
₳ 5,80
Oferta:
₳ 4,50 c/u

Suscripción 13 Ejemplares P.P.
(N° 9 al 21)
₳ 23,40
Oferta ₳ 19,8

Los 3 productos
+ Suscripción 13 Ejemplares P.P.
+ Tomo I P.P.
+ 1 casete Tomo I P.P.
₳ 41,80
Oferta: ₳ 30,50

Entre todos los suscriptores a nuestra revista P.P. vigentes al 10/1/86 sortearemos el día 15/1/86 ante Escribano Público una computadora CZ 2000 (Spectrum) de 48K donada por CZERWENY Electrónica. El ganador lo informaremos en el N° 9 de P.P.

Vos podes ser el ganador. Suscribite YA con el talón

CLUB de USUARIOS de Spectrum

Sede: Av. de Mayo 749 Piso 3 Of. 20. (1084) Capital Tel. 34-8357/8135

El Club de Usuarios de Spectrum (C.U.S.) hace su presentación a través de esta Sección fija de Programación Popular, para acercarse a los actuales y futuros usuarios de Computadoras Spectrum (en todas las versiones y modelos), y ofrecerles participar en sus diversas actividades.

Temas de soft:

Bucles automáticos

Particularidades de la utilización de los ciclos **For-Next.**

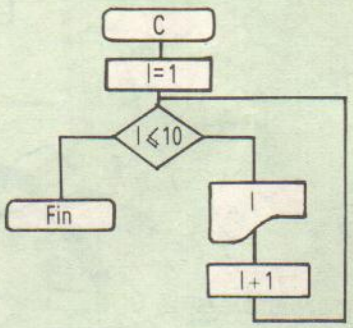
La posibilidad de realizar ciclos repetitivos contando con un contador que se "testea" automáticamente al final de c/u de ellos, le da al Basic una potencia considerable. La instrucción **For-Next** no presenta dificultades de aprendizaje pero, por ser sencilla su utilización, en general no se aprovecha al 100% de sus posibilidades, o lo que es peor, no se maneja su lógica operacional completa. Por ejemplo, una clásica interpretación errónea hace suponer que, finalizado el bucle, el valor de la variable es el del límite de ciclo. Es decir, para un ciclo **For I=1 TO N** hay quien cree que el valor final con que I sale del mismo es N. Aclarémoslo a través del análisis de los pasos lógicos que cumple esta instrucción.

Sabemos que ella contiene las funciones de un contador, cuyo incremento (positivo o negativo) es definible al programarlo y una bifurcación condicional (que bifurca según el valor del contador).

Un pequeño programa que utiliza dichas funciones, prescindiendo del ciclo automático, es:

Programa 1

```
10 Let I=1
20 If I =10 Then Go To 40
30 Stop
40 Print "I="; I
50 Let I=I+1
60 Go To 20
```



Programa 2

Si apelamos al bucle **For-Next**, nuestro programa se transforma en:

```
10 For I=1 To 10
20 Print "I="; I
30 Next I
```

El evidente ahorro de instrucciones, implica la condensación de funciones del programa 1 en instrucciones más potentes del programa 2, al que analizaremos línea por línea:

La línea 10 inicializa la variable (contador), es decir, le asigna el valor inicial 1, tal como lo hace la línea 10 del programa 1. Además, realiza la bifurcación condicional: va a ejecutar todas las instrucciones que encuentre antes del **Next I**, siempre que el valor de la variable sea menor o igual a 10. Por lo tanto, ejerce la misma función que la línea 20 del programa 1.

La línea 20 es idéntica a la 40 del programa 1 y no merece comentarios.

La línea 30 incrementa el valor de la variable según lo indicado en la línea 10 (en este caso, incremento = 1 ya que se obvió **Step**), tal como lo hace la línea 50 del programa 1, y bifurca incondicionalmente a la línea 10, del mismo modo que la línea 60 del programa 1.

Tanto en uno como en otro programa, el incremento se realiza previamente a la bifurcación que lleva a la comparación. Por este motivo, cuando la variable asuma el último valor permitido para correr internamente por el bucle, sufrirá un nuevo incremento (**Next I**) antes de que se efectúe la comparación final (en el **For**) y el flujo, por lo tanto, saldrá del bucle, con I valiendo una unidad más que el límite fijado (en este caso 11).

Es importante tener en cuenta este detalle si se intenta utilizar en otro sitio del programa, el último valor de la variable en el bucle.

Explosivo: Así podemos definir el nacimiento de nuestro Club de Usuarios de Spectrum.

Queremos, ante todo, agradecer a la gran cantidad de usuarios (y aspirantes a serlo) que se acercaron a nosotros por carta, por teléfono y personalmente.

Esto nos alienta y da fuerzas para lograr un CUS grande y provechoso para todos.

Una aclaración: **El CUS está abierto a todo el que se interese por sus actividades.** No excluimos a nadie (ni por no tener máquina ni por tener modelos A, B o C).

Los que ya son socios, se han enterado de las bases de nuestro primer concurso de programación. Sus etapas mensuales tienen premios tan valiosos como joysticks, interfaces, cassetes con programas, etc., y los ganadores de ellos competirán por mayores que incluyen **computadoras** y otras importantes sorpresas.

Y hoy anunciamos que, gracias al aporte de nuestros socios, hemos inaugurado nuestro sector de alquiler de periféricos. Por una pequeña cuota podemos llevar a casa y utilizar interfaces para joysticks, interfaces para impresoras, joysticks, lápices ópticos.

Además, ya está funcionando nuestra biblioteca de programas y la biblioteca de publicaciones.

Si aún no lo hizo, ¿qué espera? Llene el cupón ya mismo y emvíelo a:

Club de Usuarios de Spectrum
Av. de Mayo 749
3er. piso, of. 20
(Tel. 34-8135/8357)

CORTAR Y REMITIR AL C.U.S.

- Nombre y Apellido .....
- Domicilio .....
- Código Postal .....
- Localidad .....
- Computadora (marca y modelo) ..... Edad .....

REMITIR A: Club Usuarios de Spectrum
Av. de Mayo 749 Piso 3 Of. 20
(1084) Capital. Tel. 34-8357/8135

Temas de hard

A raíz de consultas referentes a ciertas publicaciones sobre la ampliación de la memoria de la Spectrum, tenemos que confirmar que, efectivamente, en Europa es fácilmente accesible dicha transformación, especialmente gracias al simple acceso a los componentes.

Se trata de reemplazar un chip de la memoria de 32 K por otro de 64 K, modificando el circuito para permitir el direccionamiento.

A esto se le debe agregar la necesidad de la complementación con el soft adecuado para ello.

En próximos números de Programación Popular trataremos de brindar detalles concernientes a la realización de esta ampliación en nuestro país.

Los servicios de Epi

CURSOS:

Sólo Epi da un computador para Ud. solo

- Introducción a la microinformática
- Basic elemental
- Basic avanzado
- Logo para Spectrum
- Grupo hasta 8 personas
- Niños adolescentes y adultos
- Turnos mañana y noche, inclusive sábados.
- Cursos especiales para colegios

Suipacha 946 - 1er. Piso - Capital Tel. 311-8618

EN VERANO 15% DE DESCUENTO

CASSETTES PARA

TI-99/4A
MICRODIGITAL
COMMODORE 64
COMPILADOR para TI-99/4A

LOGO PARA SPECTRUM Y COMPATIBLES

~~A~~ 22,65

SISTEMAS

Las microcomputadoras con equipos aptos para procesar sistemas comerciales, verifíquelo!, EPI se lo asegura.

DISPONEMOS:

- STOCK
- FACTURACION
- CLIENTES
- CUENTAS CORRIENTES
- CONTABILIDAD
- LISTAS DE PRECIOS

...y también sistemas a su medida.

SE ACEPTAN TARJETAS DE CREDITO

Epi

EMPRESA PARA INFORMATICA

INSTITUTO: SUIPACHA 946 1er. PISO (1008) CAPITAL.
TELEFONOS: 311-8618

FERROCARRIL OESTE
CUCHA CUCHA 350,
CAPITAL

CAMARA DE INDUSTRIALES DE ARTEFACTOS PARA EL HOGAR
PARAGUAY 1855

ref. 5

Lenguaje de máquina (III)

Codificación hexadecimal - Desensamble

1. Repaso

En un encuentro anterior vimos algunos ejemplos concretos de operaciones que puede ejecutar el **Z-80**, junto con programas sencillos sin otro propósito que el de ejemplificar el concepto de **"programa en LM"**.

Habíamos destacado, y queremos hacerlo nuevamente, que la idea de "programa", y la práctica de la programación, cambian bastante en el paso del **Basic** al ensamblador o **assembler**.

Las diferencias fundamentales radican en que **Basic** es un lenguaje evolucionado, preparado para facilitar la programación, remedando el uso de la gramática normal en un lenguaje humano como el inglés. Es un lenguaje **cerrado** en el sentido de que el control del ordenador se mantiene en manos del usuario, aun bajo condiciones de error de diversos tipos. Un programa en **LM** que nosotros escribamos, para permanecer bajo nuestro control, debe llevar un curso perfectamente calculado, libre de circunstancias imprevistas que lo hagan "colgarse" en lazos eternos, porque aquí el control del teclado ya no nos pertenece. Finalizada la tarea, debe volver a "tierra firme", al **Basic**, donde nuevamente dominamos la situación.

Lo contrario –por ejemplo, la ausencia de una instrucción **RET** (de retorno)– hará que perdamos definitivamente el control sobre la máquina, y no habrá más remedio que reinicialar el **Basic**, apagando y encendiendo la computadora... y perdiendo así la totalidad de los contenidos de la RAM.

(De todos modos, esto es lo peor que puede suceder; no hay riesgo de dañar la computadora mientras no nos pongamos a introducir sofisticaciones a nivel de hardware, es decir, conectando periféricos de nuestra invención.)

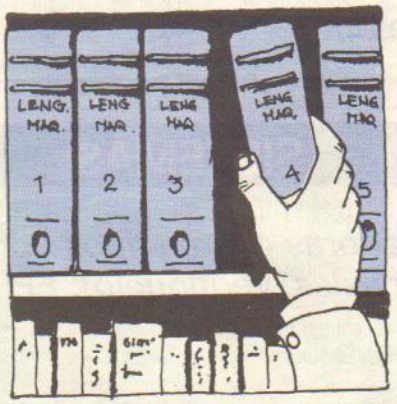
Vimos también que los datos y las instrucciones constan en memoria bajo el mismo formato: siempre están representados por uno o más números a un byte, siendo tarea del **Z-80** discernir cuáles números representan códigos operacionales de instrucciones y cuáles datos para que éstas operen.

...Y quedando a cargo nuestro proveer a cada instrucción de la cantidad **exacta** de datos que necesite.

2. Codificación hexadecimal

Vamos ahora a ampliar el juego de instrucciones ya conocidas, pero vistiéndolas con un ropaje más técnico. Vamos a hablar de **códigos hexadecimales**, con los que se suele representarlas.

Todos sabemos que el sistema numérico que habitualmente utilizamos es el **decimal o de base 10**, así llamado porque está compuesto por **diez dígitos** 1,2,3,...9, y que es **posicional**, porque al escribir números de más de un dígito el valor de cada uno se ve modificado por su posición relativa en la escritura del número. Concretamente, a medida que nos desplazamos lugares a la izquierda en una tal escritura, los valores **relativos** de los dígitos se obtienen multiplicando sus valores **absolutos** por potencias crecientes de diez. (Verbigracia, si escribimos el número 222 en sistema decimal, el primer "dos" a la izquierda representa 2 por diez al cuadrado (200), el segundo, dos por diez a la uno (20), y el tercero, dos por diez a la cero (2).)



La característica del sistema habitual de tener diez dígitos no es para nada esencial; podemos contar y hacer aritmética en sistema con distinta cantidad de dígitos con la misma facilidad (salvo por la falta de costumbre). Como ejemplos vayan los dos siguientes: todos sabemos contar **con palotes**, lo cual es en realidad usar un sistema posicional de base 1 –que consta de un solo dígito–. Además, muchos de nosotros conocemos, con seguridad, el sistema binario, de dos dígitos 0 y 1.

Lo que sí es fundamental es que el sistema es **posicional**, lo que permite que, agregando dígitos a la izquierda, podamos escribir números tan grandes como necesitemos.

Un sistema numérico posicional de gran importancia es el **hexadecimal**, constituido por 16 dígitos. Los diez primeros se representan por los usuales **0,1,2...9,** y tienen el mismo significado.

Los otros seis son algo desconcertantes, y cuesta al principio reconocerles su bien ganado título de números; se llaman **A,B,C, D,E y F**.

A **hexadecimal**	corresp. al N°	10 **decimal**
B	"	11
C	"	12
D	"	13
E	"	14
F	"	15

(Por favor, **no confundirlos con los registros A,B,C,D,E, y F** *nombrados en la nota anterior; toda semejanza entre sus nombres es pura coincidencia.)*

La serie completa, pues, de los 16 dígitos suena algo extraña, a mitad de camino entre los números y el alfabeto: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Para indicar que un número está escrito en **base 16** agregamos una letra **H** (que no es un número, sino una abreviatura de **hexadecimal**) al final del mismo. Por ejemplo, **DH** o **Dh** significan lo que en buen cristiano es **"13"**. Esta "convención" no es obligatoria y, cuando se sabe que estamos trabajando con caracteres hexadecimales, suele prescindirse.

Para representar números mayores que 15 (o sea, FH) usamos el mismo truco que en cualquier otro sistema posicional: multiplicamos los dígitos por potencias crecientes de la base –aquí 16– a medida que los agregamos a la izquierda.

Así

En hexa	vale en decimal
AH	10
AAh	$10 \times 16 + 10 = 170$
AAAH	$10 \times 16^2 + 10 \times 16 + 10 = 2730$
21h	$2 \times 16 + 1 = 33$

La idea de usar **notación hexadecimal** al trabajar en **assembler** se debe a una múltiple conveniencia:

1. La escritura de todos los **números a un byte** queda estandarizada a **dos cifras hexadecimales**.

2. Del mismo modo, todo **número a dos bytes** queda expresado con **cuatro cifras hexadecimales**.

3. Resulta sumamente fácil pasar de **"hexa" a binario** y viceversa.

2.1. Codificación de números decimales a hexadecimal

Para ver de qué modo se escriben los números a uno y dos bytes en **hexa**, basta con aprender la forma de pasar números hasta 255 y entre 256 y 65535 –respectivamente, los rangos de 8 y 16 bytes–, escritos en decimal, a hexadecimal.

a) Si el número es **menor que 256**, *se lo divide por 16 y se expresan cociente y resto en hexa; y se escriben uno a continuación del otro, en ese orden.*

```
201 | 16
 9  | 12
```

12 en hexa C
9 en hexa 9

201 = C9H

(Que, como recordaremos, es el código operacional de la instrucción **RET**.)

b) Si el número está **entre 256 y 65535**, *lo expresamos como cociente y resto de la división por 256:*

```
324 | 256
 69 |  1
```

Y ahora, resto y cociente son números menores que 256 y podemos expresarlos en hexa según lo visto en a.)

1 en hexa 1
69 en hexa 45

El número **325** queda expresado por las cuatro cifras hexadecimales **0145H**.

c) Si el número es **mayor que 65535**... hacemos de cuenta que no existe, por lo menos por ahora. (Va a ser muy difícil que tengamos que vérnoslas con algo así.)

Esto aclara los puntos 1 y 2 de la "lista de conveniencias" de más arriba.

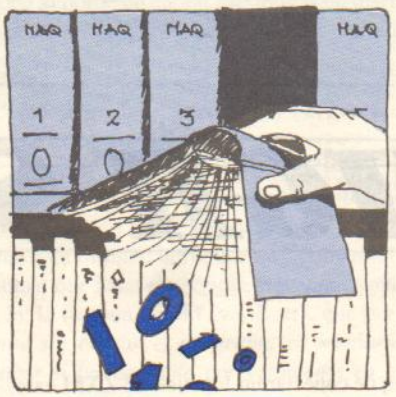
2.2. Pasaje de hexadecimal a binario

Para mostrar que es realmente fácil pasar **de hexa a binario y viceversa**, bastará con un ejemplo. El número en hexa **FBH** –dos cifras: es un **número a un byte**– está constituido por las cifras **F** y **B**. Representémoslas en sistema binario:

HEXA	BINARIO
F	1111
B	1101

Pues bien, *el número* **binario** *correspondiente a* **FBH** *es la reunión de ambas representaciones*; es el **número 11111011 en base 2**, que es de 8 dígitos binarios y por lo tanto cabe en un byte, como predijéramos.

Según nuestro artículo anterior, la instrucción de carga **LD BC,325** se escribía mediante los tres números a un byte **1,69,1**. Según lo visto recién, la codificación en hexa de los mismos viene a ser



01 45 01		
01H = 1	**45H = 69**	**01H = 1**
Código operacional de direccionamiento inmediato al registro **BC**.	Byte **"de orden bajo"**; contiene el resto de dividir 325 por 256.	**"Byte de orden alto"**; contiene el cociente de la división 325/256.

Ejercicio: *repasar la pequeña tabla de instrucciones referidas a los registros* **B**, **C** y **BC** dada en el artículo anterior y expresar sus códigos operacionales en sistema hexadecimal. Confrontar dichas expresiones con la tabla de neumónimos que se encuentra al final del manual de la **Sinclair**.

3. Desensamblar un programa

Otro ejercicio: volver a la nota **N° 1 de Lenguaje de Máquina (Programación Popular N° 4)** y examinar la línea 33 del programa **RENUMFLASH** allí presentado. Contiene un literal A$ compuesto por los códigos hexadecimales del programa resencuenciador en **LM**.

Escribir ese literal A$ separando los caracteres de a dos, es decir

11
00
00
21
A7
40... etc. y escribir sus equivalencias en decimal, usando la mencionada tabla del manual. Escribir al lado los nemónimos que correspondan.

Muchas de las instrucciones así encontradas serán interpretables a la luz de lo que ya sabemos. El acto de tomar un programa en **LM** y reconstituir el listado original de nemónimos assembler se llama **desensamblar** un programa, y es de lo más instructivo.

Al desensamblar **RENUMFLASH**, prestemos atención a las instrucciones que aparezcan y que tengan que ver con el par **HL**. Nos encontraremos con él próximamente.

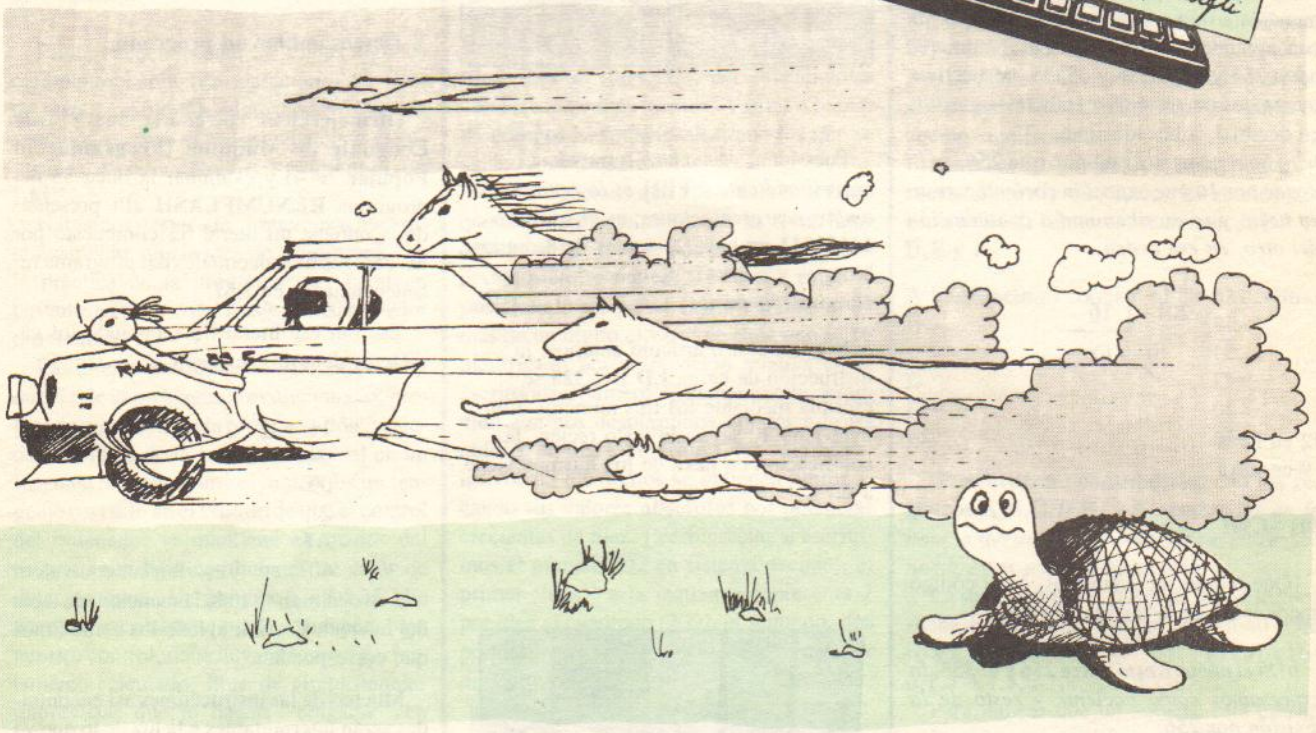
HL es especialista en recorrer grandes distancias llevando datos, ya que hay en el juego del **Z-80** útiles instrucciones que lo involucran, llamadas **de direccionamiento indirecto**, idea fundamental en programación en **LM**. Estas instrucciones, practicables también con otros registros, nos permiten depositar datos en posiciones de memoria, no haciendo referencia a las posiciones en forma explícita sino diciendo *"la posición adonde apunta* **HL***"*. Cuando **HL** cumple estas funciones, se dice que actúa como **puntero**, ya que señala direcciones cambiando de posición con suma facilidad.

Veremos éstas y otras dificultades de **HL, el mensajero de los dioses**, en un próximo encuentro.

FIN

Programas en





HIGH SPEED

Presentamos seguidamente un interesante artículo escrito por nuestro amigo **Mario**, que ya nos lo anticipara en el **SoftiClub** del Nº 6 de nuestra revista **Programación Popular**, referido a cómo grabar/leer programas en "alta velocidad" en la **TK85**

. Esperamos comentarios de nuestros amigos lectores...

Síntesis de mi odisea...

Tiempo atrás entregué en parte de pago mi **ZX81** (con módulo ampliador de memoria) por una **TK85**. Lo que más me sedujo para realizar el cambio no fue su teclado (muy superior por cierto al del **ZX81**), ni la idea de conectar un Joystick (el **ZX81** carece de tal adminículo), sino y en honor a la verdad, la posibilidad que ofrecía el **TK85** de cargar/grabar programas en **alta velocidad**... Ya no más el lento y errático cassette!

En el momento de concretar la operación le pregunté al vendedor qué tal prestación daba este computador en **HS** (High Speed para usar notación internacional) contestándome casi con una sonrisa burlona, que se había intentado infinidades de veces cargar programas en este modo pero nunca habían logrado pescar tan sólo uno. Demás está decir la sensación de inseguridad y sabor amargo que produjo en mí tal respuesta. Pero como en esta vida **NUNCA** hay que darse por vencido decidí seguir en el tren y llevar a mi hogar esta nueva maravilla electrónica y estudiarla como año atrás había hecho con la querida **ZX81**.

En realidad (y estoy seguro que el lector también lo confirmó), la mayoría de los comerciantes en esta área, saben menos del producto que venden que el más desinteresado de los principiantes; espero para bien de todos, que esta situación se esté revirtiendo.

Después de leer ávidamente los primeros capítulos del manual del **TK85** y de llegar a la conclusión que era muy semejante al del **ZX81**, busqué el capítulo específico de la carga/grabación en **HS** (Capítulo 29). Demás está decir que para el principiante no es lectura fácil.

El **TK85** es casi compatible totalmente con el **ZX81** y digo casi porque este computador tiene funciones **I/O** que sólo se en-

cuentran en las máquinas semiprofesionales, para nombrar algunas en nuestro mercado, como **Spectrum** y **Commodore 64**. Teniendo la ventaja de usar estas funciones **I** (**In**=en, dentro) y **O** (**Out**=fuera, salida) en **HS** le dan al **TK85** configuración, al menos en esta área, de **"máquina mayor"**.

Para el lector interesado aclaro que la **función I/O** permite *almacenar y recuperar (en medios magnéticos (cinta y diskette) información específica e independiente del programa*, lo que brinda un *gran ahorro de memoria* ya que sólo se usa la parte de la información entrada requiriendo de un pequeño programa cargador o grabador según sea el caso. Por ejemplo, trabajando con un archivo de datos del personal de una empresa, en vez de cargar todo el programa y sus variables se puede obtener la información requerida para determinado personal, según se desee.

Después de innumerables intentos y de haber peregrinado por cuanto comercio vendía el **TK85**, preguntando por el método de carga en **HS** sin obtener respuesta decidí cambiar mi tercer grabador de cassette por un viejo grabador **Sony** semiprofesional de dos velocidades de cinta y por supuesto a rollo descubierto. Al tercer intento y después de haber regulado varias veces el volumen, quedando aproximadamente a tres cuartos de su recorrido total, **el milagro se produjo. Apareció en el borde inferior de la pantalla el tan ansiado 0/0**. ¿Sería posible un milagro? ¿Y por qué no? Con mano temblorosa de la emoción apreté **RUN NEWLINE** y el Backgammon apareció en pantalla... Por primera vez después de haber luchado casi cuatro meses con los RAND USR 8405... 8539... 8630!!!

¿Se imaginan mi alegría? El monstruo, al fin, había sido conquistado!!!

Y ahora un poco de conceptos técnicos

Mi pregunta fue la siguiente: *Habiendo probado con tres grabadores de cassette*, todos en excelentes condiciones, usando cinta de primera calidad, ajustando los cabezales para obtener la lectura de sonido más aguda posible y sin embargo no logré nada, *¿por qué repentinamente alcanzo lo ansiado usando un viejo grabador del cual ya ni se obtiene cinta?...* (La cinta magnética en rollo es más ancha que la de cassette y es sumamente difícil de conseguir ya que fue reemplazada en honor a la practicidad de este último).

La respuesta en realidad es sencilla: *Usando el doble de velocidad de los grabadores de cassette deposito la misma información (cantidad de bits) en mayor cantidad de cinta*, lo que posibilita que esta información sea menos *errática*, además mejoró la *relación señal/ruido*, obteniendo mayor volumen audible y mayor fidelidad de la reproducción, factores importantísimos para que el **TK85** lea la cinta correctamente con un volumen disponible mayor que usando la velocidad del cassette.

El capítulo 29 del manual de la máquina dice que este computador puede trabajar en transferencia de información a la cinta magnética en dos formatos: Formato I: **300 bauds por seg.** (el **baud** es una unidad de medida usada en la electrónica digital que permite el cálculo de la velocidad a la cual se transfiere el flujo de datos binarios a través de un determinado canal). Y Formato II: **4000 baudios por seg.** *¡Se dan cuenta, qué maravilla!...*

4000 bauds o bits por segundo no son fáciles de manejar, por ello llevando la velocidad de la cinta de 4,75 cm/seg a 9,75 cm/seg, incrementando la anchura de la pista en la cinta (obteniendo así mayores frecuencias) ya que estos grabadores de rollos descubiertos utilizan cabezales más anchos que los de cassette, y elevando el volumen disponible de la señal, el milagro se produjo, llevando al **TK85** a cargar programas extensos en muy poco tiempo, lo que lo pone al borde de cargar programas en disco (por supuesto para otras máquinas ya que ésta carece de unidad de disco).

Cómo se graban, verifican y cargan los programas en HS

En este formato de operación *debemos olvidarnos* de **SAVE" "** y de **LOAD" "**. Además este formato introduce un "comando" nuevo que hace el servicio más interesante aún: la función **VERIFY** (verificar), que *permite sin borrar el programa en memoria saber si fue grabado correctamente en la cinta magnética*.

Para **grabar un programa** en HS debemos dar el siguiente comando compuesto de dos palabras **RAND USR 8405** , tratando de no confundir **RAND** (abreviatura de RANDOM=al azar, sin orden) con **RND** y entrando a través de la tecla de función la función **USR** (USER=usuario; esta función se usa para desviar el programa a una posición de memoria específica y en lenguaje de máquina).

Una vez escrito **RAND USR 8405**, poner el grabador en **RECORD** y dar **NEWLINE**. La pantalla titilará rápidamente, se verán líneas finitas y negras separadas por espacios blancos y luego el reporte **0/0**. **Pero aún no leamos ni borremos el programa; verifiquémoslo**, es decir veamos si se grabó correctamente *comparando el computador el programa depositado en la cinta con el que aún tiene en su memoria*. Para ello hagamos **RAND USR 8539** (sin equivocarnos), rebobinemos la cinta a su punto de origen, es decir donde comienza el programa (*es muy necesario un cuentavueltas en el grabador*) y apretemos **NEWLINE** poniendo en marcha el grabador en **PLAY**. La pantalla se llenará de líneas finitas moviéndose rápidamente y luego nos comunicará con un **OK** en su borde superior izquierdo si todo salió bien, si no nos dirá **ERRO**. Sin desanimarnos, intentémoslo nuevamente pero con el sonido un poquitito más bajo. Para ello debemos cargar nuevamente la orden **VERIFY**.

Si queremos **cargar un programa** cualquiera que esté en memoria usando HS, debemos grabarlo primero en HS, luego daremos al **TK-85** la siguiente orden (habiendo por supuesto, rebobinado la cinta al origen del programa) **RAND USR 8630**. Si todo salió bien la máquina nos contestará con el consabido **0/0**.

Y para finalizar...

Sé que la mayoría de los lectores me dirán: perfecto, maravilloso, obtuviste lo que querías, pero nosotros *de dónde sacamos un viejo grabador Sony de rollo descubierto, ambas velocidades*–, si ni siquiera se consigue la cinta...? Exacto, ¿de dónde?

Pero, y ya que grabadores de cassette se consiguen en cualquier parte, así como sus repuestos, ¿por qué no hacer el intento con un buen grabador, ajustando la cabeza grab./rep. a la transferencia sonora más aguda disponible? O ¿qué tal llevar la velocidad de la cinta a 9,75 cm/seg, cambiando para ello la polea sobre el eje del motorcito, llevándola a casi el doble de su diámetro, o busbando alguna solución intermedia?

Una sugerencia, y sin reír por favor: *el cabo de un plumero se presta maravillosamente bien para fabricar la poleíta*. Cortemos una rebanada de aproximadamente medio centímetro de alto y con una lima fina hagámosle a canaletita (no muy profunda) por donde irá la goma de tracción y luego con un compás encontremos su centro equidistante y perforemos con una pequeña mecha de acuerdo al diámetro del eje del motor. Obtendremos así una polea liviana e intercambiable con la original.

A probar pues... y no me pregunten sobre el tema porque el Sony acaba de largar un denso humo oscuro de un olor acre, lo que me hace pensar seriamente después de tantas penurias en un **Commodore 64** con su **Datassette**... (Las malas lenguas dicen que anda muy bien)... Ya veremos.....

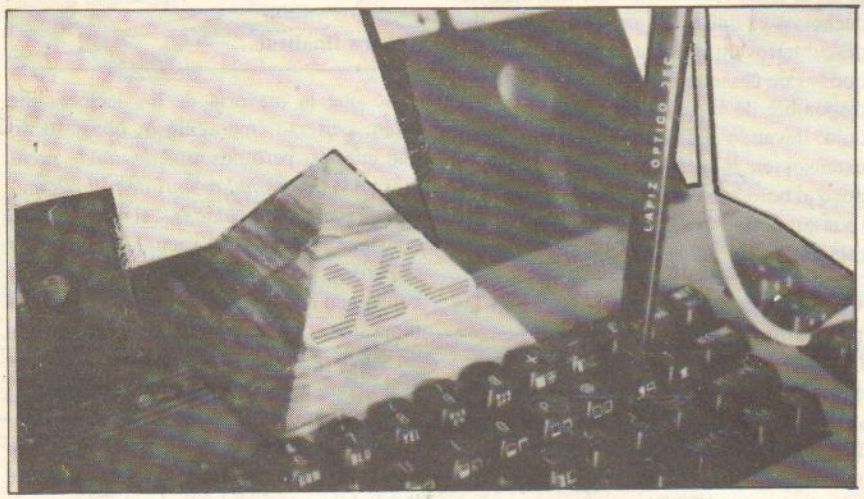
Novedad en el País: Curso de Basic en Casete para **Spectrum** y **TK90X** de 48 K

La empresa **Ciberne Software** lanza en diciembre un **Curso de Basic en casete,** inscripto dentro de la línea de Software Educativo de esta pujante empresa argentina.

Con este nuevo desarrollo, **Ciberne** pondrá al alcance del público usuario de estas populares máquinas, de un producto que cumpla con el objetivo de identificarlo con la informática y propenda a un aprendizaje de este lenguaje para acercarlo paulatinamente a una formación profesional en el tema.

El curso en casete estará compuesto por 10 lecciones que explican de una manera clara y rica en ejemplos prácticos los conceptos básicos de Estructura, Armado y Programación en Basic, posibilitándonos interactuar con la computadora y "ver" como responde ésta a las secuencias del programa.

Lápiz óptico argentino para **Commodore**



La empresa **DEC Electrónica** ha desarrollado el primer lápiz óptico argentino para computadoras **Commodore,** cuya función básica es la de escribir sobre la pantalla del TV o monitor.

Dentro de las posibilidades que ofrece, se encuentran las de dibujar con distintos tipos de trazos, realizar figuras geométricas (círculos, cuadrados, etc) y líneas rectas, rellenar cualquier parte del dibujo con distintos colores y utilizar la técnica "zoom" que, mediante la ampliación de la imagen, permite corregir cualquier imperfección "punto por punto". Es muy importante la característica que posee de combinar colores, obteniéndose asi una gran variedad de tonos y matices que alcanzan hasta doscientas variantes.

También se puede "borrar" el trabajo realizado a la última acción ejecutada dejando intacto el resto del mismo. Cuenta además con un archivo que le permite conservar el dibujo para utilizarlo en distintas oportunidades, disponiéndose de modelos de trabajos realiza con con el lápiz.

Otra opción muy impactante que posee es la técnica de "espejo", que permite la realización de figuras simétricas y efectos "caleidoscópicos"

El programa también contiene una rutina de impresión que permite imprimir normalmente con distintos tipos de impresoras, e incluso en colores (si disponemos de la máquina respectiva)

La distribución en la **Argentina** de este producto ha sido confiada a la firma **Papel-Shop,** Pueyrredón 658, Capital, con Tel. 88-8582.

Ya está en el aire el **Radio Computer Club!!!**

De lunes a viernes, de 17,30 a 18,00 hs. y por la onda de **FM** de **Radio Continental** (105.5 MH) ya se emite normalmente el **Radio Computer Club,** primer programa radial que transmite programas de microcomputación en nuestro país.

El primer mes de esta novedosa programación radial, contó con el auspicio de la firma **Arvoc SA,** para su línea de computadoras **Microdigital,** y con el apoyo de su cadena de distribuidores en nuestro país.

Nueva línea de productos **Sinclair** en nuestro país. También llega la **Spectrum Plus.**

Czerweny Electróncia SH lanza su nueva línea de productos **CZ,** entre los que se cuenta la **CZ1000, CZ1500, CZSpectrum** y la nueva **CZSpectrum Plus.**

Las primeras han sido remozadas totalmente en su aspecto exterior, ampliándose la integración de partes nacionales, especialmente de circuitos electrónicos. En tal sentido, y principalmente con la **CZSpectrum** la empresa nos informa haber mejorado notablemente, gracias al aporte de tecnología propia, el porcentaje de confiabilidad de estos nuevos equipos, respecto de sus predecesores importados (principalmente los procedentes de **Portugal),** logrando incluso reducir los márgenes de fallas por debajo de los stándares de origen.

En cuanto a la nueva **CZSpectrum Plus,** sabemos que sus principales innovaciones son el diseño ergonométrico del teclado y las mejoras en disipación de calor, tal cual adelantáramos oportunamente a nuestros lectores.

VIDEO JUEGOS

en castellano
para microcomputadoras



rel 11

CARGA GARANTIZADA

- Sistema concord de altísima velocidad (*)
- Instrucciones en castellano

COMPATIBLE PARA

- SPECTRUM
- TK 90

YA APARECIERON PROGRAMAS PARA COMMODORE 64



PIDALOS EN NEGOCIOS DE COMPUTACION Y DISQUERIAS

LEUCO SOFT Belgrano 3896 (1210) Capital Tel. 982-9645

TK-85 el microcomputador que Ud. puede usar

EL MICROCOMPUTADOR QUE LE BRINDA LAS MAXIMAS POSIBILIDADES DE LA COMPUTACION



• EXPANSOR DE MEMORIA de 64 Kbytes RAM **(opcional)**



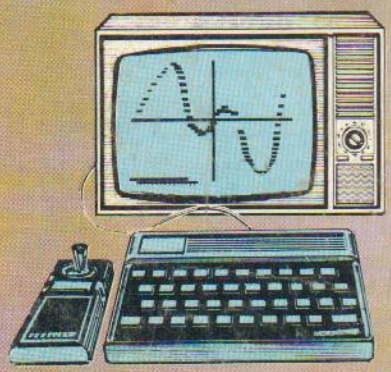
• JOYSTICK
(órgano de comando externo)
(opcional)

• TK85, fue especialmente diseñado y construido para que Ud. lo use con extrema simplicidad.

• Sólo basta consultar su ameno y completo manual de instrucciones, en Castellano por supuesto, y Ud. podrá aprender computación en forma fácil, rápida y práctica y en lenguaje BASIC.

• A partir de allí Ud. podrá preparar sus propios programas, o también utilizar centenares de programas que existen en el mercado para estudiar: matemáticas; física; química; biología; música; catalogar clientes; controlar stocks; programar compras y ventas; controlar su cuenta bancaria; poner en orden gastos e impuestos en su hogar; jugar ajedrez; backgamond; y ¿por qué no? con su exclusivo joystick disfrutar de los más fantásticos video juegos: guerra de las galaxias, invasores espaciales; etc. y todo lo que Ud. puede imaginar.

El único límite del microcomputador MICRODIGITAL TK85 es "SU IMAGINACION".



OPCIONALES:

• Joystick (órgano de comando externo), para juegos de video • Generador de sonido • Expansor de memoria de 64 Kbytes RAM • Impresora directa • Programador de EPROM • Interface para conexión de impresora profesional (paralela).

MICRODIGITAL TK-85

Importa y distribuye ARVOG S.A.C.I.

Tte. Gral. Juan D. Perón 1563 (ex Cangallo) C.P. 1037 - Capital. Tel. 35-2400 / 2511 / 8241.

GARANTIA: 6 MESES

En venta en comercios de microcomputadores, artículos del hogar, electronica, fotografia y librerías.

Ref 1