Año 1 Nº 4 Julio 1985

Precio: A 1,50 - Exterior: 3 U$S

programación popular

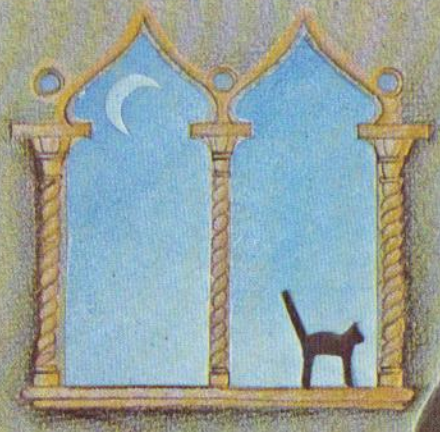
Curso Basic introducción

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

Ficha Técnica



MICRODRIVE SPECTRUM



CONCURSO DE PROGRAMAS

CONCESION Nº 1949
FRANQUEO PAGADO
CONCESION Nº 840
TARIFA REDUCIDA

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, electronica, fotografia y librerias.

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.
Avda. DIAZ VELEZ 4149 (1200) Capital Federal Tel.: 981-1980/9212

ZX SPECTRUM +" ALL RIGHTS RESERVED SINCLAIR RESEARCH LTD

Dirección General:
Dr. Héctor J. Silva

Dirección financiera
Enrique H. Ramírez

Arte y diagramación:
Norberto L. Silva
Francisco J. Fenili
Miguel Angel Bravo

Colaboran en este número:
Alfredo Eduardo Argenti
Norberto Horacio Domizzi
Jorge G. F. Gianella
Eduardo A. Grosclaude
Jorge Ariel Guglielmone
Carlos Daniel Presali
Gustavo Rigo
Daniel Weselka

Redacción, Administración y Publicidad:
Paso 421, piso 7
(1031) Capital Federal
Tel. 48-8691

Correspondencia y suscripción
Casilla de Correo 118
Sucursal 53

Composición
Ernesto González
Impresión:
Reprografías JM.A.S.A.
Fotocromía:
Gráfica Boedo

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

PROGRAMACION POPULAR es una publicación de Ediciones del Futuro S.A. con domicilio en la calle Paso 421, 7º Piso (1031) Capital Federal. Tel.: 48-8691. 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 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

Estamos muy contentos y agradecidos por la cantidad y calidad de los programas recibidos para el concurso, lo que nos enorgullece y demuestra la capacitación de nuestros lectores argentinos y por la confianza depositada en Tu Revista **Programación Popular**. *En este número publicamos nada menos que 5 (si, leyeron bien) programas ganadores del mes.*

También iniciamos el primer Curso argentino de lenguaje Basic por Instrucción Programada (para los no/recién iniciados) y Notas sobre Lenguaje de Máquina (para los "adelantados")

Desde ya que también tenemos Ficha Técnica (Microdrive Spectrum), Novedades, y muchas sopresas más.

Hasta el número 5.

Héctor J. Silva

EQUIPOS

Fecha Técnica: Microdrive Spectrum e Interfase 1.
Detalles y características técnicas de estos interesates periféricos de la Spectrum (aún no disponibles en el país) 21

NOVEDADES

Noticias, anticipos y comentarios de nuestro mundo de la microcomputación 38

PROGRAMACION

Curso Basic - Introducción
Iniciamos la entrega de un curso de idioma Basic I por "Instrucción Programada" 12

Lenguaje de Máquina - PEEK-POKE -USR.
Iniciamos esta serie de notas sobre LM, las que han sido muy solicitadas por nuestros lectores 32

Tablas (II)
En este número; Tablas alfanuméricas 18

PROGRAMAS

Indice (CZSpectrum - TK90-TS2068) 36

Diccionario (CZSpectrum - TK90 - TS 2068) 6

Organo y efectos (CZ1000/1500 - TK83/85) 16

Ataque (CZ Spectrum - TK90 - TS 2068) 30
Clarividencia (CZ Spectrum - TK90 - TS 2068) 4
Generala (CZ1500 / TK85) 25
Piedritas de la discordia (todas) 28

Manejo de fechas (III) (todas) 20

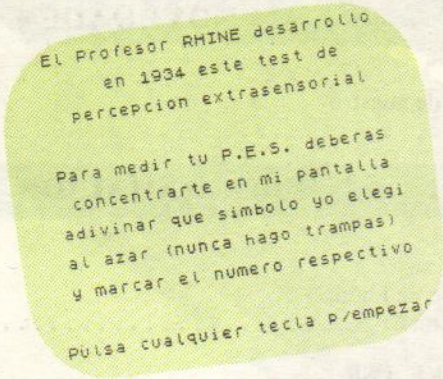
VARIOS

- Bolsa del Usado 11
- Concurso de Programas (Bases) 11
- Mataburros 41
- Softi Club 8

CLARIVIDENCIA

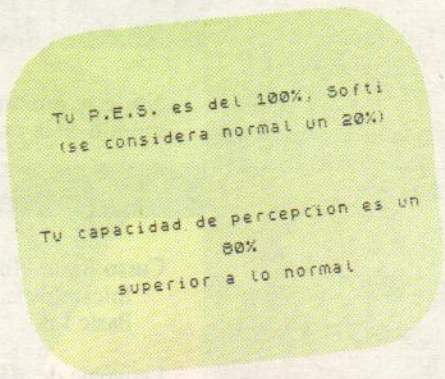
Softi está imposible!!! Ahora dice que es clarividente Que tiene poderes paranormales Que es un fenómeno por su percepción extrasensorial . . . Tendrá razón???. . .

Este programa, si bien lo clasificamos en la categoría de "juegos" reproduce, en realidad, una versión simplificada del test de ***Rhine*** para medir la capacidad de percepción de cualquier persona de predecir o "adivinar" acontecimientos ignorados por ella, llamada *"percepción extrasensorial"* o *"clarividencia"*.

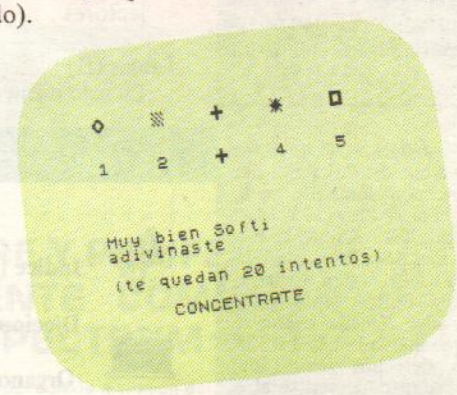


El problema, incluso a nivel científico consiste en medir esta facultad de *percepción extrasensorial* que tienen los seres humanos y que es dable apreciar intuitivamente, difiere entre una persona y otra. Pero cómo precisar con exactitud y rigor científico la capacidad de una persona dada???

Pues bien, el profesor ***Rhine*** desarrolló un método por el cual en determinadas condiciones, se le realizan al sujeto en cuestión una serie de preguntas acerca de cuál de una cantidad de objetos figurativos fue elegido en cada caso por quien pregunta. Se anotan cada una de las respuestas y luego se confrontan con las respuestas correctas. Si en una cantidad suficientemente elevada de preguntas, los porcentajes de respuestas correctas superan la media normal (que resulta ser la probabilidad esperada en función de la cantidad de respuestas correctas respecto de las respuestas posibles), significa que esta persona en cuestión resulta poseer una capacidad de *"percepción extrasensorial"* superior a lo normal.



Si esta capacidad supera en mucho lo normal estaríamos quizás en presencia de un auténtico. . . brujo (o algo parecido).



Estructura del Programa

10 / 30	Inicialización
40 / 100	Pantalla de Instrucciones
110	Pide e ingresa nombre del encuestado
120	Inicia al ciclo de las 25 preguntas
130	Selecciona un símbolo al azar
140 / 160	Presenta en pantalla la totalidad de los símbolos posibles
170 / 180	Valida la consistencia de la respuesta. Si es imposible, va a 170
190 / 220	Muestra la respuesta dada y el símbolo elegido por la propia maquina mediante el "flasheo" de los símbolos respectivos en pantalla
230 / 270	Evalúa si la respuesta es o no correcta, acumulando la cantidad de aciertos, en su caso. Continúa el ciclo de preguntas.
280	Calcula el *P.E.S. (capacidad de percepción extrasensorial)*
290 / 320	Muestra en pantalla los resultados del test. Si la evaluación supera la media normal, informa y festeja la situación
330 / 390	Rutina de generación de símbolos del test de ***Rhine*** (conjuntamente con los DATA de líneas 410 a 450)
400	Rutina de sonido de festejos

Por supuesto que a través de nuestra computadora se presentan condiciones ideales para desarrollar un examen de las facultades de cada persona, o sea: elección al azar de los dibujos sin posibilidad de una "secuencia" lógica que permita al encuestado prever, incluso a nivel inconciente, la respuesta correcta, de modo de dejar absolutamente librado al azar y a la capacidad de percepción extrasensorial la elección en cada caso. También nuestra máquina nos asegura un cómputo exacto de las respuestas y tiene la capacidad necesaria como para repetir tantas veces como queramos el test sin "cansarse" en absoluto, posibilitando entonces resultados más confiables cuanto mayor sea la cantidad de preguntas y respuestas logradas.

El modo de uso del programa es muy sencillo, luego de cargarlo correctamente, lo haremos correr con GOTO 30. El programa nos explicará el objetivo del juego/test, nos pedirá el nombre del jugador/conejillo de indias/pichón de brujo. Y ahí comienza la cosa, pues la máquina elegirá un símbolo sin que nosotros sapamos cual es, y nos pedirá que nosotros lo adivinemos.

Aquí es donde debemos apelar a nuestra intuición/percepción extrasensorial/clarividencia/o como quieras llamarlo y presionar una tecla del 1 al 5 que representa al símbolo que "sentimos" que es el que eligió la máquina. No debemos "pensar" sino "sentir" y guiarnos por la primera sensación, sin reflexionar antes de digitar. Tampoco debemos anticiparnos a la máquina y esperar a que ésta primero haga su elección y luego nos pida nuestra adivinanza. Y ahora veamos el programa . . .



Tabla de Variables

base	Código del carácter inicial de los Gráficos definidos por el usuario que representan los símbolos del test de *Rhine*
ok	Cantidad de respuestas correctas
n$	Nombre del encuestado/jugador
t	Variable FOR/NEXT representativo del ciclo de 25 preguntas del test
azar	Número al azar (del 1 al 5)
a$	Respuesta (alfanumérica)
resp	Respuesta (numérica) = VAL a$
pes	Cociente de Percepción Extra Sensorial (P.E.S.)
n	Variable FOR/NEXT: ciclo de generación de 5 caracteres gráficos
f	Idem utilizado para realizar las 8 definiciones de cada carácter
x	Variable intermedia

TS2068 / CZ Spectrum / TK90

```
  10 REM SOFTI 1985
  20 REM test P.E.S.
  30 BORDER 0: PAPER 0: INK 9: C
LEAR
  40 PRINT AT 1,2;"El Profesor R
HINE desarrollo";AT 3,7;"en 1934
 este test de";AT 5,4;"percepcio
n extrasensorial"
  50 PRINT AT 9,2;"Para medir tu
 P.E.S. deberas:";AT 11,3; FLASH
 1;"concentrarte"; FLASH 0;" en
mi pantalla";AT 13,2;"adivinar q
ue simbolo yo elegi";AT 15,2;"al
 azar (nunca hago trampas)";AT 1
7,2;"y marcar el numero respecti
vo"
  60 LET base=144: GO SUB 330
  70 PRINT AT 21,1;"Pulsa cualqu
ier tecla p/empezar"
  80 PAUSE 4E4
  90 CLS
 100 LET ok=0
 110 INPUT "Como te llamas?: ";
LINE n$
 120 FOR t=1 TO 25
 130 LET azar=INT (RND*5)+1
 140 BEEP .2,50
 150 PRINT AT 5,6;CHR$ base;AT 5
,11;CHR$ (base+1);AT 5,16;CHR$ (
base+2);AT 5,21;CHR$ (base+3);AT
 5,26;CHR$ (base+4)
 160 PRINT AT 8,6;"1";AT 8,11;"2
";AT 8,16;"3";AT 8,21;"4";AT 8,2
6;"5"
 170 LET a$=INKEY$
 180 IF a$<="0" OR a$>"5" THEN G
O TO 170
 190 LET resp=VAL a$
 200 PRINT AT 8,resp*5+1; FLASH
1;CHR$ (base+resp-1)
 210 PRINT AT 5,azar*5+1; FLASH
1;CHR$ (base+azar-1)
 220 IF resp=azar THEN PRINT FLA
SH 1;AT 13,6;"Muy bien ";n$;AT 1
4,6;"adivinaste": LET ok=ok+1: G
O SUB 400
 230 IF resp<>azar THEN PRINT AT
 13,6;"Lo siento ";n$;AT 14,6;"n
o acertaste": BEEP .1,-10-resp:
BEEP .9,-20-azar
 240 IF t<25 THEN PRINT AT 16,6;
"(te quedan ";25-t;" intentos)";
AT 18,10; FLASH 1;" CONCENTRATE
"
 250 PAUSE 130
 260 CLS
 270 NEXT t
 280 LET pes=ok/25*100
 290 LET a$="Tu P.E.S. es del "+
STR$ pes+"%, "+n$
 300 PRINT AT 5,16-LEN a$/2;a$;A
T 7,2;"(se considera normal un 2
0%)"
 310 IF pes>20 THEN PRINT AT 13,
0;"Tu capacidad de percepcion es
un";AT 15,15; FLASH 1;pes-20;"%
"; FLASH 0;AT 17,6;"superior a l
o normal": GO SUB 400
 320 STOP
 330 FOR n=base TO base+4
 340 FOR f=0 TO 7
 350 READ x
 360 POKE USR CHR$ (n)+f,x
 370 NEXT f
 380 NEXT n
 390 RETURN
 400 FOR x=1 TO 60 STEP 2: BEEP
.03,azar+x-20: NEXT x: RETURN
 410 DATA BIN 00011000,BIN 00111
100,BIN 01100110,BIN 11000011,BI
N 11000011,BIN 01100110,BIN 0011
1100,BIN 00011000
 420 DATA BIN 10010010,BIN 01001
001,BIN 00100100,BIN 10010010,BI
N 01001001,BIN 00100100,BIN 1001
0010,BIN 01001001
 430 DATA BIN 00011000,BIN 00011
000,BIN 00011000,BIN 11111111,BI
N 11111111,BIN 00011000,BIN 0001
1000,BIN 00011000
 440 DATA BIN 10001001,BIN 01011
010,BIN 00111100,BIN 11111110,BI
N 01111111,BIN 00111100,BIN 0101
1010,BIN 10010001
 450 DATA BIN 11111111,BIN 11111
111,BIN 11000011,BIN 11000011,BI
N 11000011,BIN 11000011,BIN 1111
1111,BIN 11111111
```

DICCIONARIO





El programa que presentamos a continuación permite la carga, correcciones y consulta de un Diccionario inglés-castellano y fue realizado fundamentalmente en lenguaje de máquina, en combinación con el Basic.

Lo primero permite lograr un acceso casi inmediato a cualquiera de las 2000 palabras susceptibles de ser incorporadas al Diccionario.

Veamos entonces los programas y las instrucciones que nos enviara *Alfredo*.

CZSpectrum - TK90 - TS 2068

```
  1 LOAD ""CODE : LOAD ""CODE
  2 CLEAR 34500
  5 '
 10 BORDER 0: INK 7: PAPER 0: C
LS : POKE 23658,8: POKE 23609,5:
 LET PN=0: LET W=0
 20 CLS : PRINT
                DICCIONARIO

            INGLES/CASTELLANO

              MARZO DE 1985


        CAPACIDAD:2000 PALABRA
 DE 15              CARACTERES CA
DA UNA"
 25 PRINT
 30 PRINT
"  TIEMPO MAX.DE ACCESO:0.5 SEG
"
 40 RANDOMIZE USR 64890: LET DI
=255*PEEK 23670+PEEK 34971
 50 PRINT : PRINT : PRINT
                1  UTILIZAR EL
 DICCIONARIO
                2  CORRECCIONE
S AL ARCHIVO"
 55 PRINT "
                3  FIN DE OPERAC
IONES
                4  INFORMES DEL
SISTEMA"
 60 LET X$=INKEY$: IF X$="" THE
N GO TO 60
 65 LET X=VAL X$: IF X=2 THEN G
O TO 5000
 66 IF X=2 THEN GO TO 6000
 70 IF X=3 THEN GO TO 4000
 72 IF X=4 THEN GO TO 8000
 75 IF X=1 THEN GO TO 85
 79 GO TO 60
 85 IF DI=35000 THEN CLS : PRIN
T AT 0,5;"
                        LA BASE D
E DATOS ESTA            EN BLANCO
.PUEDE OPTAR
                          #A   A
RMARLA                    #L   L
EERLA DEL TAPE"
 87 IF DI<>35000 THEN GO TO 99
 88 IF CODE INKEY$=65 THEN CLS
: GO TO 100
 90 IF CODE INKEY$=76 THEN GO T
O 9993
 92 GO TO 88
 99 CLS
100 DIM I$(1,15): DIM C$(1,15)
110 INPUT "PALABRA EN INGLES:
                          "; LINE X
$: IF LEN X$>15 THEN GO TO 110
115 IF X$="0" OR X$="" THEN GO
TO 20
120 LET I$(1)=X$: GO SUB 2000
130 RANDOMIZE USR 65080: LET DB
=(256*PEEK 34980+PEEK 34981)+30+
35000
140 IF DB<=64850 THEN GO TO 100
0
145 PRINT I$(1);
150 INPUT "Lo siento,no la cono
zco. Por    favor ingrese su sig
nificado
                          "; LINE x$
155 PRINT TAB 15;X$
160 IF LEN X$>15 THEN GO TO 150
170 LET C$(1)=X$: GO SUB 3000
190 GO TO 100
999 REM
       IMPRIMIR PALABRAS

1000 FOR I=DB TO DB+29: PRINT CH
R$ PEEK I;: NEXT I: PRINT : GO T
O 100
1999 REM
       PASAR AL BUFFER

2000 FOR I=0 TO 14: POKE (34985+
I),CODE (I$(1)(I+1)): NEXT I: RE
TURN
2999 REM
       PALABRAS NUEVAS
3000 LET P$=I$(1)+C$(1)
3010 FOR I=0 TO 29: POKE (DI+I),
CODE (P$(I+1)): NEXT I
3020 LET DI=DI+30: LET PN=PN+1:
RETURN
3999 REM
       GRABAR NOVEDADES

4000 CLS : IF PN=0 THEN PRINT "
     NO SE HAN REGISTRADO
     PALABRAS NUEVAS, POR
     LO TANTO PUEDE FINA-
     LIZAR DE CORRER EL
     PROGRAMA SIN NECESI-
     DAD DE GRABAR EL AR-
          CHIVO EN CINTA.": INPUT "
  1 VELVE A CORRER EL PROGRAMA "
;W: IF W=1 THEN GO TO 20
4010 IF W<>0 THEN STOP
4020 PRINT "SE HAN REGISTRADO ";
PN;" PALABRAS": PRINT "PREPARE
SU GRABADOR, GRACIAS": SAVE "DAT
OS"CODE 35000,DI-35000
4025 PRINT : PRINT : PRINT "
 AHORA VERIFIQUEMOS": VERIFY ""C
ODE : GO TO 10
5999 REM
       CORRECCIONES

6000 CLS : PRINT TAB 10;"CORRECC
IONES": PRINT TAB 10;"
     ": PRINT
6005 DIM I$(1,15): DIM C$(1,15)
6010 INPUT "PALABRA A CORREGIR:
                          "; LINE X$
: IF LEN X$>15 THEN GO TO 6010
6015 IF X$="" OR X$="0" THEN GO
TO 20
6020 LET I$(1)=X$: GO SUB 2000
6030 RANDOMIZE USR 65080: LET DX
=(256*PEEK 34980+PEEK 34981)+30+
35000:
6040 IF DX>64850 THEN PRINT : PR
INT TAB 7;"PALABRA INEXISTENTE":
 PRINT : GO TO 6010
6045 LET PN=PN+1
6050 INPUT "PALABRA CORRECTA:
                          "; LINE
X$: IF LEN X$>15 THEN GO TO 6050
6055 LET I$(1)=X$
6060 INPUT "SIGNIFICADO CORRECTO
:
                          "; LINE
X$: IF LEN X$>15 THEN GO TO 6060
6070 LET C$(1)=X$: LET P$=I$(1)+
C$(1)
6080 FOR I=0 TO 29: POKE (DX+I),
CODE P$(I+1): NEXT I: PRINT P$
6090 GO TO 6010
8000 CLS : LET X$="INFORMES DEL
SISTEMA": PRINT TAB ((32-LEN X$)
/2);X$: PRINT : PRINT "---------
------------------------"
8010 PRINT : PRINT : PRINT "MEMO
RIA RAM LIBRE:"; FREE ;" Bytes"
8020 PRINT : PRINT "PALABRAS REG
ISTRADAS:";(DI-35000)/30: PRINT
: PRINT "---------------
-----------------": PRINT
8030 PRINT : PRINT : PRINT "   ©
 EDUARDO ALFREDO ARGENTI"
8040 PAUSE 0: GO TO 20
9990 REM
       CARGAR DATABASE

9993 CLS : PRINT                "
     COLOQUE EL GRABADOR
     EN POSICION DE LEER
     EL DICCIONARIO DE
          DATOS"
9995 LOAD "DATABASE"CODE : GO TO
 10
9998 REM
       GRABAR PROGRAMAS

9999 SAVE "P1" LINE 1: SAVE "C1"
CODE 65080,285: SAVE "C2"CODE 64
890,180
```

(48 Kb)

1. Cargar el programa en Basic y luego, mediante cualquier método de los conocidos READ-DATA, POKE, etc) introducir en memoria los programas en lenguaje de máquina (listados C1 y C2) a partir de las direcciones de memoria indicadas en cada caso.

El siguiente programa puede utilizarse para ese fin:

```
7000 INPUT "DIRECCION INICIAL: "
;DI
7010 INPUT "DIRECCION FINAL: ";D
F
7020 FOR I=DI TO DF
7030 PRINT I;" = ";
7040 INPUT "BYTE: ";B
7050 PRINT B
7060 POKE I,B
7070 NEXT I
7080 STOP
```

El procedimiento a seguir para cargar los programas en lenguaje de máquina con el programa "cargador" anterior será

1.1. Ingresar el programita anterior (líneas 7000 a 7080)

1.2. GOTO 7000

1.3. Ingresar como "dirección inicial" y "dirección final" las correspondientes a cada uno de los listados C1 y C2 respectivamente.

1.4. Ingresar el código del *"byte"* correspondiente a cada una de las direcciones de memoria, según los listados C1 y C2.

1.5. A continuación vemos el listado C1:

C1

```
65080 = 33      65081 = 169
65082 = 136     65083 = 34
65084 = 114     65085 = 255
65086 = 33      65087 = 184
65088 = 136     65089 = 34
65090 = 90      65091 = 255
65092 = 33      65093 = 0
65094 = 0       65095 = 34
65096 = 112     65097 = 255
65098 = 42      65099 = 114
65100 = 255     65101 = 110
65102 = 38      65103 = 0
65104 = 235     65105 = 42
65106 = 90      65107 = 255
65108 = 110     65109 = 38
65110 = 0       65111 = 235
```

```
65112 = 167      65113 = 237
65114 = 82       65115 = 124
65116 = 181      65117 = 46
65118 = 1        65119 = 40
65120 = 2        65121 = 46
65122 = 0        65123 = 38
65124 = 0        65125 = 124
65126 = 181      65127 = 32
65128 = 3        65129 = 195
65130 = 111      65131 = 254
65132 = 195      65133 = 183
65134 = 254      65135 = 33
65136 = 159      65137 = 135
65138 = 34       65139 = 114
65140 = 255      65141 = 42
65142 = 112      65143 = 255
65144 = 235      65145 = 33
65146 = 1        65147 = 0
65148 = 235      65149 = 25
65150 = 34       65151 = 112
65152 = 255      65153 = 42
65154 = 112      65155 = 255
65156 = 235      65157 = 33
65158 = 30       65159 = 0
65160 = 235      65161 = 205
65162 = 104      65163 = 52
65164 = 235      65165 = 33
65166 = 184      65167 = 135
65168 = 235      65169 = 25
65170 = 34       65171 = 90
65172 = 255      65173 = 42
65174 = 112      65175 = 255
65176 = 235      65177 = 33
65178 = 30       65179 = 3
65180 = 235      65181 = 167
65182 = 237      65183 = 82
65184 = 124      65185 = 181
65186 = 46       65187 = 1
65188 = 40       65189 = 2
65190 = 46       65191 = 0
65192 = 38       65193 = 0
65194 = 124      65195 = 181
65196 = 32       65197 = 3
65198 = 195      65199 = 160
65200 = 254      65201 = 195
65202 = 135      65203 = 254
65204 = 195      65205 = 74
65206 = 254      65207 = 42
65208 = 114      65209 = 255
65210 = 235      65211 = 33
65212 = 1        65213 = 0
65214 = 235      65215 = 25
65216 = 34       65217 = 114
65218 = 255      65219 = 42
65220 = 190      65221 = 255
65222 = 235      65223 = 33
65224 = 1        65225 = 0
65226 = 235      65227 = 25
65228 = 34       65229 = 90
65230 = 255      65231 = 42
65232 = 114      65233 = 255
65234 = 235      65235 = 33
65236 = 184      65237 = 135
65238 = 235      65239 = 167
65240 = 237      65241 = 82
65242 = 46       65243 = 0
65244 = 48       65245 = 2
65246 = 46       65247 = 1
65248 = 38       65249 = 0
65250 = 124      65251 = 181
65252 = 32       65253 = 3
65254 = 195      65255 = 236
65256 = 254      65257 = 195
65258 = 74       65259 = 254
65260 = 42       65261 = 112
65262 = 255      65263 = 235
65264 = 33       65265 = 0
65266 = 1        65267 = 235
65268 = 112      65269 = 179
65270 = 200      65271 = 124
65272 = 37       65273 = 33
65274 = 0        65275 = 0
65276 = 5        65277 = 16
65278 = 203      65279 = 17
65280 = 255      65281 = 237
65282 = 106      65283 = 237
65284 = 82       65285 = 68
65286 = 48       65287 = 16
65288 = 246      65289 = 244
65290 = 84       65291 = 14
65292 = 203      65293 = 17
65294 = 255      65295 = 237
65296 = 106      65297 = 167
65298 = 237      65299 = 90
65300 = 68       65301 = 242
65302 = 40       65303 = 237
65304 = 16       65305 = 242
65306 = 203      65307 = 17
65308 = 255      65309 = 25
65310 = 71       65311 = 96
65312 = 105      65313 = 34
65314 = 108      65315 = 255
65316 = 33       65317 = 164
65318 = 138      65319 = 229
65320 = 42       65321 = 108
65322 = 255      65323 = 235
65324 = 225      65325 = 115
65326 = 42       65327 = 108
65328 = 255      65329 = 235
65330 = 33       65331 = 0
65332 = 1        65333 = 235
65334 = 205      65335 = 104
65336 = 52       65337 = 34
65338 = 138      65339 = 255
65340 = 33       65341 = 165
65342 = 138      65343 = 229
65344 = 42       65345 = 112
65346 = 255      65347 = 235
65348 = 42       65349 = 138
65350 = 255      65351 = 235
65352 = 167      65353 = 237
65354 = 82       65355 = 235
65356 = 225      65357 = 115
65358 = 201      65359 = 0
65360 = 0        65361 = 0
```

1.6. Seguidamente vemos el listado C2:

C2

```
65362 = 0        65363 = 0
65364 = 0        65365 = 0
64890 = 33       64891 = 164
64892 = 136      64893 = 34
64894 = 94       64895 = 255
64896 = 42       64897 = 94
64898 = 255      64899 = 110
64900 = 38       64901 = 0
64902 = 235      64903 = 33
64904 = 0        64905 = 0
64906 = 235      64907 = 167
64908 = 237      64909 = 82
64910 = 124      64911 = 181
64912 = 46       64913 = 1
64914 = 40       64915 = 2
64916 = 46       64917 = 0
64918 = 38       64919 = 0
64920 = 124      64921 = 181
64922 = 32       64923 = 3
64924 = 195      64925 = 152
64926 = 253      64927 = 195
64928 = 203      64929 = 253
64930 = 42       64931 = 94
64932 = 255      64933 = 235
64934 = 33       64935 = 30
64936 = 0        64937 = 235
64938 = 25       64939 = 34
64940 = 94       64941 = 255
64942 = 42       64943 = 94
64944 = 255      64945 = 235
64946 = 33       64947 = 82
64948 = 253      64949 = 235
64950 = 167      64951 = 237
64952 = 82       64953 = 46
64954 = 0        64955 = 48
64956 = 2        64957 = 46
64958 = 1        64959 = 38
64960 = 0        64961 = 124
64962 = 181      64963 = 32
64964 = 3        64965 = 195
64966 = 203      64967 = 253
64968 = 195      64969 = 128
64970 = 253      64971 = 42
64972 = 94       64973 = 255
64974 = 235      64975 = 33
64976 = 255      64977 = 0
64978 = 235      64979 = 122
64980 = 179      64981 = 200
64982 = 124      64983 = 77
64984 = 33       64985 = 0
64986 = 0        64987 = 6
64988 = 16       64989 = 203
64990 = 17       64991 = 23
64992 = 237      64993 = 106
64994 = 237      64995 = 82
64996 = 63       64997 = 48
64998 = 16       64999 = 16
65000 = 244      65001 = 24
65002 = 14       65003 = 203
65004 = 17       65005 = 23
65006 = 237      65007 = 106
65008 = 167      65009 = 237
65010 = 90       65011 = 66
65012 = 242      65013 = 40
65014 = 237      65015 = 16
65016 = 242      65017 = 203
65018 = 17       65019 = 23
65020 = 255      65021 = 71
65022 = 96       65023 = 105
65024 = 34       65025 = 112
65026 = 255      65027 = 33
65028 = 164      65029 = 136
65030 = 229      65031 = 34
65032 = 112      65033 = 255
65034 = 235      65035 = 42
65036 = 136      65037 = 255
65038 = 115      65039 = 235
65040 = 235      65041 = 33
65042 = 255      65043 = 0
65044 = 235      65045 = 205
65046 = 104      65047 = 52
65048 = 34       65049 = 136
65050 = 255      65051 = 33
65052 = 165      65053 = 136
65054 = 229      65055 = 42
65056 = 94       65057 = 255
65058 = 235      65059 = 42
65060 = 136      65061 = 255
65062 = 235      65063 = 167
65064 = 237      65065 = 82
65066 = 235      65067 = 225
65068 = 115      65069 = 201
65070 = 0        65071 = 0
```

Una vez ingresada toda la información, hacer GOTO 2 para correr el programa y verificar su funcionamiento. Luego de depurado de posibles errores de "dedos" SAVE' ar los tres programas, tecleando GOTO 9999.

Al correr el programa aparecerá en pantalla el Menú general.

```
DICCIONARIO
INGLES/CASTELLANO
MARZO DE 1985

CAPACIDAD: 2000 PALABRA DE 15
CARACTERES CADA UNA
TIEMPO MAX.DE ACCESO:0.5 SEG

1  UTILIZAR EL DICCIONARIO
2  CORRECCIONES AL ARCHIVO
3  FIN DE OPERACIONES
4  INFORMES DEL SISTEMA
```

La *primera* opción nos lleva a otro Menú donde es posible escoger entre armar un nuevo diccionario o bien cargar de un casete uno que ya haya sido grabado anteriormente.

La *segunda* opción nos permite enmendar o cambiar palabras reconocidas por el diccionario.

La *tercera* opción siempre es obligatoria realizarla luego de haber empleado el programa pues detecta las modificaciones que hubieran sido efectuadas, procediendo a grabarlas en el casete que contendrá las palabras del diccionario.

La *cuarta* y última opción nos permite obtener algunos datos del programa.

Softinotas

1. *El programa prevé la búsqueda del significado en castellano de una palabra en inglés. No está prevista una doble búsqueda, o sea la posibilidad de conocer también la traducción al inglés de una palabra en castellano. Hubiera sido interesante este doble acceso.*
2. *También sería interesante una opción que permitiera listar el diccionario con una clasificación previa.*
3. *Insisto, **Alfredo**, en que resulta más conveniente para nuestros amigos lectores, el contar con una mejor documentación del programa, principalmente de su estructura lógica y la tabla de variables.*
4. *Muy bueno lo tuyo **Alfredo**. Ah, no me mandaste la dirección así que si no nos comunicamos por teléfono, escribime para decírmela o pasá por la Editorial a retirar tu cheque.*
   Chau. . .

Softi



SOFTI CLUB

Sirvan estas primeras líneas para expresarles mis felicitaciones y desearles el mayor de los éxitos a todos los integrantes del equipo responsable de la edición de *Programación Popular.*

Si bien es cierto que actualmente existen en el mercado numerosas publicaciones (la mayoría importadas) sobre la microcomputación que nos llenan los ojos de equipos, tecnologías, etc., también es cierto, que la mayoría, lamentablemente, tenemos los bolsillos vacíos y todo ese micromundo no se adapta a nuestra realidad socio-económica.

Programación Popular creo sin equivocarme, responde a las necesidades de una inmensa cantidad de aficionados que para iniciarse en el maravilloso mundo de la informática ha adquirido, con mucho esfuerzo en algunos casos, una modesta microcomputadora.

Con relación a los temas de vuestra publicación sólo me resta sugerirles, que de ser posible, incluyan aquellos relacionados con la computación aplicada a la educación primaria en una primera etapa y secundaria en una etapa posterior.

Como docente pienso que es en las escuelas donde el niño debe tener sus primeras experiencias con un ordenador. Ocurre que la mayoría de las publicaciones se ocupan de cómo enseñar BASIC utilizando a la computadora como un fin pero son muy pocas las que se ocupan de cómo emplear a la computadora como un medio para el aprendizaje, que en definitiva, salvo mejor opinión, debería ser el objetivo primordial de la informática en la educación.

El tema es para hablar un largo rato, pero considerando la posibilidad de que en alguna oportunidad esta carta sea publicada, he tratado de ser lo más breve posible.

Hasta la próxima.

Jorge Osvaldo Salcedo
Rafaela - Santa Fe

Gracias por tus elogios.
Pensamos igual respecto de la importancia de la computación en la educación como un medio y no como un fin.
Y en eso estamos.
Te saludo con afecto.

Softi

Creí necesario escribir esta carta para felicitarlos por la revista. Tengo 16 años y realmente me interesa mucho todo lo referente a la programación. Quisiera saber los programas disponibles para la TI 99/4A que hay en el mercado y sus precios.

Espero que sigan publicando las fichas técnicas de cada computadora y sus accesorios. Mi último pedido es que me informen dónde puedo conseguir el simulador de vuelo para la TI.

Sin más, y agradeciéndoles su colaboración, los saluda cordialmente

Alejandro Gustavo Schmidt
Capital

Estamos organizando el envío de material técnico-comercial a nuestros Softi-lectores, Alejandro..
En la revista solo estoy autorizado a responder consultas técnicas (no comerciales). Lo siento.
Chau.

Softi

Me dirijo a ustedes con el fin de saludarlos y felicitarlos por la genial idea que tuvieron de poner en circulación la maravillosa revista *Programación Popular.* No es que los esté adulando, pero realmente hacía falta en el mercado una revista del tipo de las que ustedes publican. Como haber, había muchas revistas de computación, pero todas ellas estaban dirigidas a usuarios de grandes máquinas (grandes para mí) olvidándose de los que poseemos pequeñas microcomputadoras como la CZ 1000. Ustedes se preocuparon por nosotros y de ahí mi agradecimiento.

Les cuento que tengo ganas de comprarme una TK 85, pero para ello necesito vender la Sinclair CZ 1000, por eso me atrevo a pedirles si pueden publicarme en la parte de *Softi usados* el siguiente ofrecimiento (tal cual está o reducido de acuerdo a su necesidad de espacio):

Bueno ahora me despido recordándoles mi felicitación y deseándoles que sigan así.

Luis B. Brizuela
Capital Federal

Gracias, Luis.
Va tu aviso. Suerte.

Softi

En 1er término quiero decirles, que para mi, el descubrir la revista fue todo un descubrimiento, como lo fue también la computación, que hace ya poco tiempo que empecé, y llegó a entusiasmarme un montón. Tengo 13 (trece) años, y estudio Ciclo Básico Secundario. No tengo computadora, pero pronto vamos a comprar la Commodore 64, cosa que el Nº 2 de la revista me vino al pelo, ya que traía la Ficha Técnica de esta.

La revista es excepcional y me encanta, pero me llevé una decepción al ver que los programas que traía, no tenían cabida para la Commodore. Quisiera que me contestaran si el programa de Calendario, del Nº 2 sirve para esta, o en el caso, si se haría posible, para que me lo traduzcan y me lo manden por correo, y así lo consiga hacer funcionar en la Commodore.

También me gustaría saber donde puedo conseguir la Nº 1.

Pablo Sebastián Rovner
Capital

Ya estamos editando programas para la Commodore, Pablo.
Sólo respondemos a consultas a través de la revista, para que así el esfuerzo, por mínimo que sea, pueda ser aprovechado por todos nuestros amigos lectores Por otra parte no debes extrañarte que incluso sea otro lector el que te responda a través nuestro sobre el programa "Calendario". Qué lindo, no?...
El número 1 está agotado.
Chau...

Softi

Mi nombre es Ignacio Germán López, tengo 13 años y estoy muy interesado en todo lo referente a la microcomputación.

Tengo un computador personal TK85 para mi uso exclusivo. Trabajo con un grupo de amigos en una TI99/4A. Soy lector de vuestra revista, en la cual encuentro material de sumo interés para analizar y reproducir en mi computadora.

Quiero hacerles llegar mi felicitación por la calidad de vuestra publicación.

Les solicito, tenga a bien, en la medida de vuestras posibilidades, me hagan llegar información general sobre los equipos TK85 y TI99/4A.

Los saludo muy atte.

Ignacio Germán López
Capital Federal

Se agradece, Ignacio.
Tomamos nota de tus necesidades.

Softi

Soy Iván Fernando Sarachman, un estudiante de 5to. año bachillerato, tengo 18 años y una TI99/4A, a la que debo alimentar con buenos programas. Programas como el que publicaron en el Nº 2 de *P.P.*, por ejemplo. Comienzo con el sentido de mi carta, conocí vuestra publicación en Expousuaria '85 y les cuento que al ver la presentación, la tapa, dije: "La Compro", y así lo hice.

La exclamación vino a que era la primera revista, que yo veía, con programas para la TI 99.

El problema fue que solo pude comprar el Nº 2, porque según me explicaron en el Stand, solo podían venderme el Nº 1 si me suscribía, ya que tenían pocos de estos ejemplares, lamentablemente no me puedo suscribir, entonces quisiera que me digan si hay forma de que lo consiga suelto.

Otra cosa que quiero pedirles es que publiquen más programas para Texas, y sería bárbaro si fueran, por ejemplo, uno para TI Basic y dos para TI EXTENDED BASIC, por número.

Ya que los usuarios de esta computadora encontramos pocos programas disponibles en revistas.

Bueno me despido agradeciendo vuestra atención y felicitándolos por Programación Popular, que es un presente del futuro.

Juan Fernando Sarachman
Ituzaingo P.B.A.

Intentamos satisfacer a todos nuestros lectores, Iván. Tomamos nota de tu inquietud.

Respecto del Nº 1, efectivamente se reservaron unos pocos ejemplares para los suscriptores en Expousuaria, que quisieran completar su colección. De todos modos, estamos confeccionando una "lista de espera" para una futura reimpresión

Gracias y chau...

Softi

Me dirijo a ustedes con el fin de comunicarles mi opinión sobre su revista.

Se ve que está escrita en un idioma al alcance de principiantes y no tan principiantes, lo que la hace muy atractiva, sobre todo en la parte de juegos de video.

Me gustaría también que se editara otro tipo de material, como ser de aplicación en física, matemáticas, y sobre todo de aplicación en el manejo de archivos simples, stocks, etc.

En el 2º número ya empezaron a tratar estos temas, y espero que los continúen con más intensidad, así se podrá aprovechar con más profundidad la capacidad de estas "maquinitas" maravillosas aunque la verdadera capacidad dependa de la imaginación y habilidad del operador.

Carlos A. Ponce
Capital

Tomamos nota de tus datos y sugerencias. Carlos.

Chau...

Softi

Les escribo para darles mi opinión sobre su revista. Compré el número 1 y 2. Estaba a punto de asociarme, pero me di cuenta que para lo que yo necesito esta revista no me es útil.

Lo que sucede es que me voy a comprar una Commodore 64 y esta revista no trae ningún programa para esta computadora. Me la voy a comprar porque a mi entender es la computadora más completa en el mercado de las micro que se fabrican en el país. Además su precio es accesible comparado con otras computadoras como la TI99 que además de ser lenta y pasada de moda en la Argentina le hacen una propaganda como si fuera la mejor computadora y la cobran bastante cara comparada con lo que realmente es.

A pesar de no convencerme la revista en ese sentido me parece muy clara y concisa en la explicación de los programas y las tablas de variables y no se va en palabrerías inútiles que tienden a confundir al lector.

Es también muy bueno el tema de "Manejo de archivos" y "Tablas".

En cuanto a programas los que más me agradan son los relacionados con Matemáticas y Física.

Agradecería me informaran de algún libro que explique el lenguaje LOGO y tenga programa.

Adrián Cano
Rio Gallegos - Santa Cruz

1. *Como vos bien sabés,* Adrián, *la Commodore es una máquina de reciente aparición en el mercado argentino. Como comprenderás hemos debido tomarnos un tiempo para desarrollar programas que estén a la altura de nuestros lectores y de nuestra revista y por esono salieron antes. Hemos recibido muchos pedidos como el tuyo.*
2. *Los comentarios sobre la TI99/4A son tu opinión, que yo respeto y transcribo simplemente.*
3. *No hay mucho en Logo,* Adrián. *Podés consultar "Primeros pasos en Logo", de Monteil (Edit. Paraninfo). Gracias por tus críticas y comentarios*

Softi

Quiero felicitarlos por la revista; es muy buena y muy clara, tanto en impresión como en contenido.

Tengo 17 años, soy estudiante secundario y poseo una CZ 2000 Spectrum.

En el número 2 de *Programación Popular* adelantaron que en el número 3 se publicaría la ficha técnica de esta microcomputadora.

Me gustaría que en esa ficha (o de no ser posible, en otra sección de la revista) figuren los periféricos disponibles en nuestro país.

También desearía que incluyan una sección destinada a comentar los programas más importantes disponibles en el mercado.

Espero recibir pronto sus noticias y les deseo mucha suerte.

Roberto Santiago Chervo
San Nicolás - P.B.A.

1. En el número 3 publicamos la Ficha Técnica de la CZ Spectrum (ó CZ 2000), como habrás visto y en este número anticipamos la Ficha Técnica de los Microdrives y la Interfase.

2. Ya tenemos en desarrollo una sección destinada a comentar los programas del mercado.

Gracias y hasta pronto, Roberto.

Softi



NOTICIA IMPORTANTE

Todos los amigos que se comuniquen con nosotros, se publiquen o no sus cartas, serán automáticamente asociados, EN FORMA GRATUITA, al SOFTI CLUB.

En tal carácter, recibirán periódicamente información técnica y comercial que resulte de su interés (listas y precios de cassettes de programas, bibliografía, equipos, accesorios, etc.).

También gozarán de beneficios especiales, como ser: descuentos especiales en comercios adheridos, descuentos por suscripción a PROGRAMACION POPULAR, planes especiales y oferta de productos, y otros beneficios más que estamos gestionando y que oportunamente pondremos a su disposición.

ESPERAMOS SUS IDEAS Y COMENTARIOS...

(1) Es muy difícil adaptar un TK85 para que trabaje con logo?

(2) Es posible cambiar la salida de FM Canal 3 a otro canal: el canal 3 local se recepciona sin antena y crea interferencias.

(3) Pueden conseguirme los circuitos del CZ 1000 y TK 85 o los códigos de las fichas de salidas posteriores?

(4) Podrá publicar programas para calculadoras programables tipo Texas 55, 66 ó Casio, etc.? existe algún dispositivo que permita la conexión de estas a un TV., como puede hacerse con la Mini Sharp TS?

(5) Pueden obtenerse manuales en tellano para una SINCLAIR 2056? ¿dónde?

Aldo Antolín
Gral. Pico La Pampa

Softi:

1) No conocemos nada, por ahora. .. Habrá que esperar.

2. Es posible pero un cambio de ese tipo puede llegar a ser bastante costoso.

3. Trasladamos tu inquietud. Veremos también de publicar los circuitos en números posteriores.

4. No lo tenemos previsto.

5. Supongo que hablarás de la TIMEX Sinclair 2068 (no 2056). No hay manuales específicos en castellano, pero podrías utilizar los de la Spectrum.

Te saludo, Aldo

Softi

Me comunico con Uds. para expresarles mi conformidad con la revista, he leído los dos primeros ejemplares y me parecen de un gran contenido informativo y explicativo. Veo muy bien la gran variedad de programas que exponen permitiendo una amplia gama de aplicación; ya que como poseo escasos conocimientos en computación, gracias a sus programas y a los comentarios de los mismos puedo conocer la aplicación de la mayoría de las secuencias y realizar mis propios programas.

Ahora como sugerencia, me gustaría que publicaran una especie de diccionario de Basic, es decir en cada número algunas de las secuencias y su aplicación, ya que ésto permitirá realizar programas con mayor facilidad.

(Poseo una TK 85)

Sergio Alejandro Lupi
Quilmes - P.B.A.

A partir de este número comenzamos un Curso de Basic para principiantes y no iniciados.

Espero que te sirva.

Chau, Sergio.

Softi

Hace un mes compré una TS2068 y estamos con mi hijo de 15 años practicando con la máquina.

Como Uds. saben la computadora trabaja con color pero en sistema NTSC al respecto existen 2 versiones para poder usarla en nuestros televisores PAL N a saber:

1. Convertir la máquina a PAL N
2. Convertir el TV a NTSC (BINORMA)

Mi pregunta es:

¿Qué es lo recomendable?

Existe una diferencia sustancial en ambos costos la 1ª opción es mucho más económica.

Estimo que vuestra respuesta será muy apreciada por los poseedores de la TS2068.

Espero vuestras noticias.

Sin más por ahora, sólo reciban mi aplauso por lo que están haciendo.

José Strauss
Quilmes

No existe una respuesta contundente a tu pregunta José.

Veamos ventajas y contras:

1. *Convertir la TS2068 a PAL N es más económico, pero también representa un riesgo al tener que abrir la máquina y alterar el circuito de video, teniendo en cuenta la alta tecnología aplicada y la falta o escasez de repuestos originales. Sólo debieras confiarla a un taller responsable y muy competente.*
2. *Hemos visto algunos casos en que los resultados de la conversión de la TS 2068 a PAL N son muy buenos, pero hay otros que son deplorables.*
3. *Convertir el TV a birnorma es más caro, pero sus resultados son más seguros en la mayor parte de los casos, con la ventaja adicional de que así podrás utilizar tu televisor para pasar videos en NTSC (grabador mediante). La decisión es tuya. . .*

Softi

Me dirijo a ustedes para felicitarlos por las cualidades de su revista, observado en sus dos primeros números. Vuestra publicación viene a llenar un vacío en este incipiente aunque numeroso mercado de las microcomputadoras.

Me llamo Gastón Palacio, tengo 14 años y poseo una TK 85 y trabajo con una Sinclair 2068 y una Radio Shack en mi colegio.

Me gustaría que publicasen notas sobre el lenguaje de máquina de las TK a la vez que aplaudo la maravillosa idea de los comentarios sobre archivo que salen en la revista. Al leer unacarta publicada en la sección del *Softi Club* en el Nº 2, mandada por el Sr. Jorge Luis de Felippe en la cual pedía que el programa "Cálculo de Valor Futuro" para TI 99 lo publicasen para TK 85 y similares, y al haberlo transformado al programa por necesidad, se lo remito para ver si es de utilidad para su actividad. . .

Gastón Palacio
Córdoba

Muy bien Gastón. Te agradezco tu trabajo.

Como enla copia que nos llegó no estaban marcados los "" y "/", y dado el interés de muchos otros amigos, aproveché y "retoqué" algunas instrucciones de tu listado, como podrás verificar.*

Entonces el programa quedó así:

```
  5 REM SOFTI 1985
 10 REM CALCULO DE VALOR FUTURO
 (CONVERSION A TK BY GASTON PALA
CIO)
 20 REM TK83/85/90-CZ1000/1500/
SPECTRUM-TS2068
 30 CLS
 40 PRINT "CAPITAL INICIAL:";
 50 INPUT CI
 60 PRINT CI
 70 PRINT "TASA DE INTERES (%AN
UAL):";
 80 INPUT T
 90 PRINT T
100 PRINT "PLAZO TOTAL (EN DIAS
):"
110 INPUT D
120 PRINT D
130 PRINT "PERIODO DE CAPITALIZ
ACION (EN DIAS):";
140 INPUT P
150 PRINT P
160 LET C=CI
170 CLS
180 PRINT "DIA  INTERES       C
APITAL"
190 PRINT "====================
============"
210 PRINT "0";TAB 19;CI
225 FOR G=1 TO INT (D/P)
230 LET I=(INT ((C*T*P)/365))/1
00
240 PRINT P*G;TAB 5;I;TAB 19;C+
I
250 LET C=C+I
270 NEXT G
280 IF INT (D/P)=D/P THEN GOTO
330
290 LET I=(INT ((C*T*(D-P*INT (
D/P))/365)))/100
300 PRINT D;TAB 5;I;TAB 19;C+I
310 PRINT "====================
============"
320 PRINT "NOTA: ULTIMO PERIODO
=";D-P*INT (D/P);" DIAS"
330 PRINT "====================
============"
350 PRINT "DATOS:CI=";CI;"  PLA
ZO=";D;" D"
360 PRINT "       T=";T;"%AN  PE
RIODO=";P;" D"
380 PRINT "========<TECLA> P/SE
GUIR========"
390 IF INKEY$="" THEN GOTO 390
400 GOTO 30
```

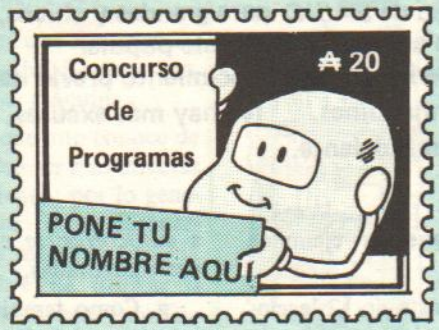
Las variables y comentarios al programa son similares al Nº 1.

Nuestros lectores notarán que el mismo fue desarrollado para todas las "Sinclair" considerando en consecuencia sus modelos más chicos.

Softi

FIN

CONCURSO DE PROGRAMAS



PREMIOS

1. **Programas Publicados**
   - A 20 (veinte australes) en efectivo.
   - Participación en el ranking del concurso semestral y anual.
   - Semestralmente, con todos los programas inscriptos en el ranking se elegirá el mejor programa que tendrá como premio una **CONSOLA DE COMPUTADORA TI99/4A o CZ Spectrum.**
   - 20 % de descuento en la suscripción anual de Tu Revista **Programación Popular.**
   - Inscripción del autor en la lista de colaboradores de Tu Revista **Programación Popular.**

2. **Programas no Publicados**
   - *Categoría A:*
     - Suscripción gratuita por 6 (seis) meses a Tu Revista **Programación Popular.**
     - 15 % de descuento en la suscripción anual de Tu Revista **Programación Popular.**
   - *Categoría B:*
     - Suscripción gratuita por 3 (tres) meses a Tu Revista **Programación Popular.**
     - 15 % de descuento en la suscripción anual de Tu Revista **Programación Popular.**
   - *Categoría C:*
     - 15 % de descuento en la suscripción anual de Tu Revista **Programación Popular.**

3. **TODOS** los autores de Programas sean publicados o no, tendrán los siguientes **beneficios:**
   - Inscripción automática en el **Softi Club** (Club de lectores de Tu Revista **Programación Popular.**
   - Participación en el sorteo del mes de libros técnicos especializados (a partir del ejemplar Nro. 4 de Tu Revista **Programación Popular).**
   - Sorteos semestrales con importantes premios (libros técnicos, becas de cursos, módulos y casetes para TI 99/4A, casetes vírgenes, y otros premios "sorpresa".

NO TE LO PIERDAS

PARTICIPA Y GANA

Nota: ver Bases del Concurso en el número 2 de Tu Revista Programación Popular.

USADOS USADOS USADOS USAD

OS USADOS

COMPRO usada o nueva una de las siguientes microcomputadoras: CZ1000 CZ 1500, TK83 ó TK85. Javier Rosenstein. Barrio SUPE M 21 C 28 G.C. (5547) Villa Hipódromo. Godoy Cruz. Mendoza. Teléfono 270359.

VENDO Timex Sinclair 2068 con impresora Alphacom 32, todo en excelente estado. Se vende con casetes, cables, manuales en ingles y castellano y 1 libro de coputación.

Llamar al Tel. 747-2083 (Luis o Francisco Agote).

VENDO Timex Sinclair 2068 completa con adaptaciones para Joystick universales bandas de entrada/salida y programas con juegos. Tel. 92-9347.

VENDO Sinclair CZ 1000 ¡nueva! con manual, transformador y cableado U$S 50 ó equivalente. Luis T.E. 941-9145

GRATIS

Para publicar un aviso en esta sección solo tienes que llamar al **T.E. 48-8691** o escribir a Programación Popular - Casilla de Correo Nº 118 - Sucursal 53

OS USADOS

USADOS USADOS USADOS USAD

CURSO BASIC (I)

Softi inicia en este número un curso de _BASIC_ para grandes y chicos, porque quiere que todos aprendan a programar en este popular y universal lenguaje. No es necesario ningún conocimiento previo de computación y sirve para todas las máquinas. No hay más excusas. Será tan divertido como útil, así que, adelante. . .

Objetivos del Curso

Softi desarrolló y nos ofrece aquí el **Primer Curso de BASIC por el sistema de Instrucción Programada** que se realiza en nuestro país a través de un medio de difusión abierto al público, tal cual es Tu Revista **Programación Popular.**

Incluso es dable destacar que no sólo no registramos antecedentes en la Argentina de este tipo de experiencias educativas "abiertas" sino que tampoco hemos encontrado antecedentes en el exterior, principalmente por las características verdaderamente inéditas de este curso que sintetizamos seguidamente; y que esperamos por cierto que tenga el resultado esperado de difundir y clarificar el uso de esta importante herramienta como es la computadora:

a. El Curso está orientado a aquellos que quieran iniciarse en el campo de la microcomputación, o sea que no tengan ningún conocimiento del tema, como así también a quienes tengan experiencia lograda en forma autodidacta (generalmente a través de los manuales de las máquinas y la experiencia práctica) y que quieran consolidar, ordenar y ampliar sus conocimientos.

b. En general, es aplicable a todas las máquinas existentes en nuestro mercado, puesto que no se tratarán las particularidades que haya introducido cada fabricante a cada modelo específico. De ser necesario se aclarará la situación particular de cada máquina.

c. Se ha elaborado con un lenguaje claro y ameno (al estilo de nuestra revista **Programación Popular**) y ayudándonos con imágenes gráficas y permanentes ejemplos prácticos.

d. El curso consta de 12 lecciones o "unidades", estructuradas de modo tal que el avance en los conocimientos sea gradual y efectivo.

e. El esquema aplicado es el conocido como de *"instrucción programada"* o sea, que se basa en la autoevaluación por el propio alumno.

Expliquemos esto: al final de cada unidad se desarrolla un *"test"* de evaluación, con preguntas específicas y concretas, que deberá ser respondido por el alumno en forma también asertiva y precisa. A tal efecto, podrá repasar y revisar cuantas veces quiera la/s leccion/es anterior/es, hasta estar seguro de la respuesta. También deberá, y esto es fundamental, *anotar las respuestas* para su propia autoevaluación posterior.

Y allí está lo importante (e inédito de que hablábamos al principio) de esta forma, al estimularse el repaso sin condicionamientos y basado en la *confianza hacia el alumno*, se fijan así los conceptos principales y se minimiza la tendencia natural al "rechazo" del estudio.

En la siguiente unidad (o lección), lo primero que se hace es proporcionar las respuestas correctas que deberán ser confrontadas por el alumno con sus propias respuestas.

Aquí viene otra novedad: en caso que haya respuestas erróneas estará previsto y se indicará qué tema debe ser repasado por el alumno. Esto "cierra" el concepto de *"instrucción programada"*, pues ayuda al alumno no sólo a autoevaluarse sino, lo que es más importante aún, a *autocorregirse* en sus concepciones equivocadas.

g. Como también conocemos el interés en los distintos componentes del núcleo familiar de aprender y/o perfeccionar los conocimientos en estos temas, sugerimos que los "papis y mamis" verifiquen que sus hijos menores comprendan y asimilen los conocimientos, a través de ayudarlos a responder el "test" y revisar junto con ellos, si las respuestas son correctas o repasar los puntos respectivos, si no lo son.

Ojalá que de esta forma contribuyamos, humildemente, a lograr una mayor integración de la familia argentina.

h. Como broche de oro y también en forma inédita para una revista masiva, al finalizar el curso publicaremos un test general que deberá ser respondido por el alumno (con o sin ayuda de sus padres) y remitirlo a nuestra revista **Programación Popular.**

Todo aquel que apruebe el examen "escrito", tendrá derecho a rendir examen práctico en una microcomputadora y, de aprobarlo, recibirá el diploma correspondiente de **Ediciones del Futuro S.A.**, que lo acreditará como **Programador Basic I**

Y aquí viene lo mejor.

Entre todos los "graduados efectuaremos un

SORTEO con IMPORTANTES PREMIOS

cuyas bases difundiremos más adelante.

No se lo pierdan. . .

Curso BASIC - Lección I

1.1.

Empecemos hablando de quién es el responsable visible del viaje por la Informática que aquí iniciamos: nuestra computadora.

Sea de la marca que sea, se trata, al fin de cuentas y hablando en sentido amplio, de una máquina.

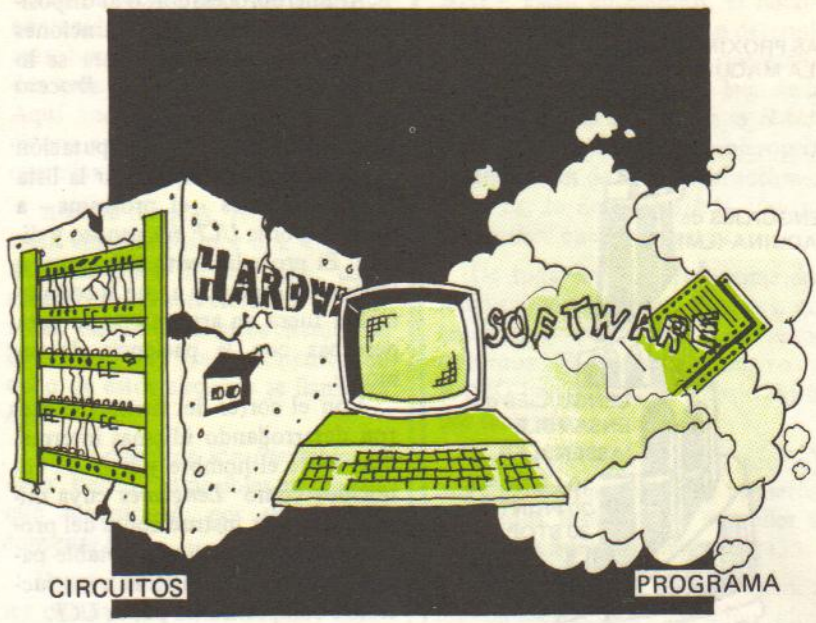
Las máquinas que uno conoce de la vida diaria suelen ser más directas en su funcionamiento: por lo general, convierten una forma de energía en otra. Esta máquina tiene la singular costumbre de utilizar la energía como medio para transformar información en. . . información



Y con eso que ganamos?

Bueno; en lo que va del momento en que la máquina "recibe información" hasta que "devuelve información", ha tenido lugar un cierto *Procesamiento* de la misma, cuya responsabilidad le compete al *Programador.*

El programador es la persona que, dado un problema relativo al manejo de una cierta información, concibe un esquema de la solución para éste y lo aplica por medio de una computadora, escribiendo un conjunto de instrucciones para ella –un *Programa*– que represente los pasos a seguir para efectuar la transformación de la información que se desea. Pero sin el paso previo de elaboración de la solución, y traducción de la misma a instrucciones interpretables por la máquina –paso donde es indispensable la intervención del hombre– será imposible que ella pueda prestarnos ninguna ayuda.



El funcionamiento y eficacia del computador, no depende exclusivamente de sus complicados y precisos circuitos electrónicos que observamos al inspeccionar su interior. Su funcionalidad depende de la asociación de dos factores:
El HARDWARE o estructura física:conjunto de componentes, circuitos y dispositivos que integran la máquina.
El SOFTWARE o elementos "intangibles" que pueden ser modificados o sustituidos: conjunto de órdenes, instrucciones y programas que "educan" a la estructura física y determinan el comportamiento específico de la máquina.
Las raíces de estos dos términos anglosajones, ilustran claramente su acepción informática.
HARD significa duro; y se utiliza para designar, genéricamente, a la totalidad de componentes y circuitos electrónicos que constituyen el ordenador.
En cambio SOFT (blando) *dá base al término que identifica a la globalidad de elementos de programación, que convierten a la estructura física en una máquina programable, o sea, en un computador.*

Por lo tanto, en nuestro camino a convertirnos en programadores, deberemos adquirir las aptitudes de:

1. Saber analizar un problema determinado, aplicando la lógica y elaborando una solución para él.
2. Saber volcar esta solución a un lenguaje comprensible por la computadora.

La primera de estas aptitudes consiste en poder confeccionar un *Algoritmo*, es decir, un conjunto de tareas que, seguidas paso a paso por la computadora, conduzcan a la respuesta esperada.

La segunda se refiere a la habilidad para traducir este *algoritmo* al lenguaje de computación que maneje la máquina.

Dáremos una serie de ejercicios, ejemplos e indicaciones tendientes a lograr la primera de ellas, pero su consecución estará dada fundamentalmente por una intensa práctica en nuestra computadora.

1.2.

En cuanto a la segunda, constituye el objetivo principal de este curso, a traves del cual daremos una descripción a fondo del vocabulario y sintaxis del **BASIC**, uno de tantos lenguajes de computación que existen, y cuyo número crece continuamente.

BASIC es el lenguaje de que vienen provistas la casi totalidad de las microcomputadoras. Es, con mucho el más popular porque

1. Es fácil de aprender.
2. Sirve para múltiples propósitos



La informática ha ido creando nuevos lenguajes de programación "de alto nivel", con una estructura semejante a la de los lenguajes humanos.

Comentaremos estas dos características.

1.2.1. Las microcomputadoras funcionan como lo hacen gracias a que están provistas de circuitos integrados miniaturizados (*Chips*) sumamente poderosos en su operación, capaces de cálculos matemáticos a velocidades alucinantes.

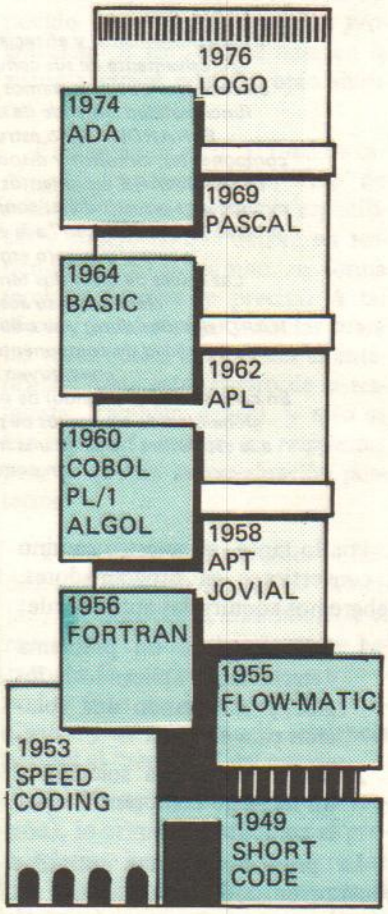
Las instrucciones para el microprocesador deben ser codificadas bajo la forma de números y alojadas en posiciones de memoria residentes en otro circuito integrado vecino al que está conectado.

El microprocesador las recaba una por una y las vá cumpliendo.

Al microprocesador, o al dispositivo que cumpla similares funciones dentro de una computadora se lo llama *Unidad Central de Proceso (UCP).*

En los inicios de la computación el procedimiento de escribir la lista de instrucciones –el programa– a cumplir por la *UCP* era penoso y difícil. El procedimiento daba lugar a frecuentes errores y a que la computación fuera un arte reservado para personas con la paciencia de un monje.

Con el correr del tiempo se fueron desarrollando idiomas intermedios entre el hombre y la *UCP*. conocidos como *Lenguajes* cuya misión es tomar instrucciones del programador en un idioma potable para el hombre y traducirlas a instrucciones comprensibles por la *UCP.*



El microcomputador es una máquina capaz de recibir una programación y comunicarse con el mundo exterior. Ello es posible debido a la existencia de lenguajes que permiten establecer una comunicación organizada, clasificables en tres categorías según sean más o menos próximos al de los circuitos electrónicos de la máquina.

Esta evolución llevó a la creación de lenguajes cada vez más parecidos al habla humana, y por lo tanto, más fáciles de aprender. El **BASIC** es uno de los últimos en esta escalada (no el último ni el más avanzado, pero sí el más apto para la computadora doméstica, o *Home-Computer*), más fácil de aprender, por su gran parecido con el inglés, que *COBOL* o *FORTRAN*, por ejemplo, lenguajes anteriores y que tienen gran aplicación en otras áreas.

1.2.2. *Cobol* o *Fortran* son lenguajes altamente especializados, diseñados para usos específicos, el primero, para aplicaciones administrativas y contables, y el segundo, para usos en ciencia y tecnología.

BASIC fue diseñado como una "aproximación para principiantes a la codificación simbólica de instrucciones", tal el significado de la sigla que lleva por nombre.

Lo que nos dice que originariamente fue pensado para la enseñanza. A raíz de ello se lo dotó de un vocabulario y una sintaxis que lo hacen apto para cualquier aplicación, sin ninguna tendencia en especial. Por eso decimos que es un lenguaje *"multipropósito"*.

El lenguaje de programación BASIC (Begginner's All-Purpose Symbolic Instruction Code) nació en 1964 en el Dartmouth College y fue concebido como un lenguaje interactivo, polivalente y de fácil aprendizaje y empleo.

Fur normalizado por el ANSI (American National Standards Institute) y de aquí parten las líneas originales del BASIC Luego surgió toda una gran familia de dialectos que se fueron desviando mas y más del lenguaje original.

En 1977, la empresa americana Microsoft desarrolló un dialecto que pretendía unificar criterios. Rápidamente fue aceptado por varios fabricantes de ordenadores, como Tandy, Apple, Commodore...

El gran boom del BASIC ha llegado con la irrupción de los microordenadores. Si bien en un principio el BASIC fue adoptado por los sistemas comerciales de tiempo compartido, de donde viene la popularidad del BASIC.

En los últimos años el BASIC se ha constituido en el lenguaje de programación más utilizado. Aunque se habla de varios lenguajes como futuros sustitutos (ADA, C, FORTH. . .) lo cierto es que ninguno amenaza seriamente la posición privilegiada que éste mantiene actualmente.

Según su forma de operación, los lenguajes pueden clasificarse en *compilables* o *interpretables*. Estos nombres significan, en el primer caso, que el lenguaje toma el programa que nosotros escribimos como un todo, y en una sola pasada lo traduce al microprocesador. En el segundo caso –lenguajes *interpretables*– el programa que escribimos (llamado programa *Fuente*) se traduce instrucción por instrucción a programa *Objeto* . La diferencia a primera vista no es notable, salvo por el hecho de que los lenguajes *interpretables* suelen ser *Interactivos;* es decir, que permiten la comunicación del programador con la máquina en "modo inmediato", en forma similar a una conversación. En cambio, en un lenguaje *compilable,* la relación entre programador y máquina es más indirecta.

BASIC está dentro de los lenguajes *interpretables* e *interactivos* (aunque existen versiones *compilables*).

1.3.

En nuestra microcomputadora, **BASIC** está implementado del siguiente modo: en su interior existen, además del microprocesador, otros dos circuitos integrados que son bloques de memoria, llamados *ROM* y *RAM*. En el bloque *ROM* está anotada, de forma fija e indeleble, la secuencia de instrucciones para el microprocesador, que no podemos alterar porque está protegida y que empieza a funcionar apenas encendemos la máquina. Esta secuencia se llama *Sistema operativo.* Se trata de un conjunto de programas escritos y grabados en la *ROM* por el fabricante, y que incluye al *Intérprete* **BASIC**, el programa traductor de las órdenes que demos a la *UCP.*

Las funciones del sistema operativo son, en general, tareas de mantenimiento de la información dentro de la computadora y control y distribución de las tareas que cumple

ROM significa, precisamente, *"memoria de lectura solamente"* y si pudiéramos escribir en ella, alterar un solo contenido de la misma, haríamos un desastre.

Por el contrario, para hacer nuestros garabatos en **BASIC** disponemos de la *RAM*, un verdadero block borrador donde podemos asentar contenidos y destruirlos sin problemas, borrándose éstos de todos modos cuando apaguemos la máquina. Aquí anotaremos nuestras instrucciones en **BASIC**, y los datos con que ellas trabajarán.

Ambos bloques, *ROM* y *RAM*, están compuestos por una larga tira de posiciones de memoria, cuyo asiento físico son circuitos electrónicos. Cada uno de estos circuitos se llama un *Bit.* Cada conjunto de ocho de estos circuitos se llama un *Byte.*

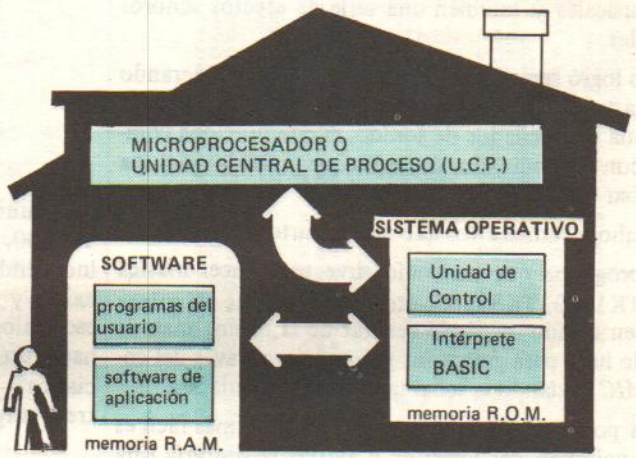
Estos circuitos son *Biestables* es decir, tienen dos posibles estados en que encontrarse: por ellos *pasará corriente o no.*

Dependiendo de cuántos y cuáles circuitos de entre los que componen cada grupo de ocho –cada *Byte*– estén encendidos, el microprocesador interpretará un determinado número o código operativo, tanto si estos números se leen de la *ROM* como si se leen de la *RAM*. Adónde debe dirigirse el microprocesador a fin de hallar instrucciones o datos, lo determinará la *Unidad de Control* que lleva provista.

De toda esta historia íntima del microprocesador y sus secuaces, no tendremos –por fortuna– que preocuparnos. El sistema operativo y **BASIC** haran todo el trabajo duro por nosotros.

Enseguida (en la próxima lección) nos encontraremos con los primeros comandos **BASIC** para aprender a dominar a nuestra computadora.

Mientras tanto, comprobemos los conocimientos ya adquiridos mediante este primer **Test.**



Una estructura edificada sobre los cimientos del hardware que con la colaboración indispensable brindada por el hombre a través del software da lugar al desarrollo del computador, un eficaz y disciplinado colaborador en cualquier actividad

TEST DE LA LECCION 1
(Escribir en hoja aparte y guardar)

1. Contestar Verdadero o Falso:

a. Las computadoras son máquinas capaces de manejar información.
b. El procesamiento de la información se hace a través de un programa.
c. Las computadoras no necesitan del hombre para resolver problemas.
d. El algoritmo es uno de los circuitos integrados que poseen las computadoras.
e. El algoritmo es un paso previo a la introducción del problema en la computadora.

2. Completar:

a. Existen muchos lenguajes de computación; tres de ellos son. . .
b. *BASIC* es un lenguaje . . . , . . . , y . . .
c. Un lenguaje de computación es. . .
d. La función del intérprete Basic es. . .
e. La diferencia entre las memorias ROM y RAM es que. . .
f. Byte y bit son . . .



ORGANO Y EFECTOS



El programa que presentamos a continuación fue preparado por ***Daniel Weselka*** de 21 años, especialmente para nuestro concurso de programas, en base a una excelente y original idea (preparate que ahí te lo digo):

Hacer "tocar" música a una máquina que ***no*** está preparada para emitir sonidos (*CZ1000* y equivalentes)!!!

Sí, escuchaste bien, el amigo ***Daniel*** ha intentado, y lo ha logrado, hacer que su *CZ1000* (que como todos sabemos no tiene ni color ni sonido incorporados) emita las notas musicales y también una serie de efectos sonoros adicionales.

Como logró semejante hazaña ***Daniel***? Pues generando la señal a través de un programa en lenguaje de máquina que simula un oscilador de sonido, el que funciona combinado con el modulador de audio que posee la máquina para grabar (SAVE'ar) los programas.

Pero ahora vamos a ver qué dice su autor:

"El programa que les envío sirve para hacer música con la TK1000, TK83, 85, etc. Aunque estas máquinas no poseen sonido se puede generar de la misma manera en que lo hace para grabar los programas, a través del conector *MIC* enviando la señal también al modulador.

"Para poder escuchar el sonido, la forma más fácil es subir el volumen del televisor y ajustar la sintonía fina del sintonizador hasta escuchar el sonido lo más claro posible, yo probé el programa en varios TV y algunos se escucha muy bien y en algunos con un poco de ruido.

"Mucho mejor es colocar el grabador como para grabar un programa normalmente y escuchar el sonido por el monitor del grabador, si no tiene monitor, colocar un parlante, audífono, auriculares, etc., en el conector *EAR* del grabador (por supuesto, también se puede grabar el sonido).

```
ORGANO Y EFECTOS

COMANDOS:

1  OSCILADOR NORMAL
2  VIBRATO NO
3  VIBRATO SI

SHIFT 1  VIBRATO 1
SHIFT 2  VIBRATO 2
SHIFT 3  VIBRATO 3

4  EFECTO DE (SUBIDA O BAJADA)
5  ACTIVA OSCILADOR 2  MODO 1
6  ACTIVA OSCILADOR 2  MODO 2
7  DESACTIVA OSCILADOR 2

SHIFT 8  MEZCLA 1
      9  MEZCLA 2
      0  MEZCLA 3

PULSE TECLA
```

```
COMANDOS:

SHIFT 4  OSCILADOR 1 ARRIBA
SHIFT 5  OSCILADOR 1 ABAJO

SHIFT 6  OSCILADOR 2 ARRIBA
SHIFT 7  OSCILADOR 2 ABAJO

LAS DOS HILERAS DE TECLAS
CENTRALES ACTUAN DE ORGANO

A=LA  S=SI  D=DO  F=RE  G=MI
H=FA  J=SOL K=LA  L=SI  (ETC.)

PULSE TECLA PARA COMENZAR
```

"Aunque nunca toqué otra máquina que no sea la TS1000, las máquinas más grandes tienen un oscilador independiente para el sonido, en la 1000 todo eso no existe y el oscilador hay que "fabricarlo" en *L.M.*, por eso dialogando con la almohada pensé –por qué si pude hacer un oscilador, no hacer 2 independientes y secuenciarlos?–, y así de a poquito, después de varios "desastres" surgió el programa que humildemente les envío.

"Cómo verán en la línea 3 se encuentran los números de las instrucciones en *L.M.* que luego (líneas 4 al 8) lo colocan en la línea 1 REM. Creo que este sistema es el mejor para que cualquier persona o chico de 8 años, pueda entrar el programa (la línea 1 REM debe contener como mínimo 76 caracteres).

"Hhaaaahh, el programa es para usar con expansión de 16 Kb pero entra perfectamente en 2 Kb de la siguiente forma: entrar las líneas del 1 al 8 inclusive, entrar RUN, eliminar las líneas del 3 al 8 (habrá que sacrificar el menú) y entrar las líneas del 250 en adelante.

"También se me ocurrió hacer la última parte en lenguaje de máquina, pero la línea 3 se complicaría tanto que sería demasiado difícil de entrar y más posibilidades de error (por otro lado para algo existe el Basic). . ."

Como ***Daniel*** también nos remitió la Estructura del Programa y la Tabla de Variables, te la ofrecemos junto con este super-programa.

CZ1000/1500 TK83/85

```
   1 REM .....................
.....................................
.....................
   2 FAST
   3 LET A$="024.011.058.194.064
.254.100.040.004.033.194.064.052
.219.254.205.193.064.211.255.205
.193.064.024.023.000.033.200.064
.052.033.155.064.052.203.102.040
.010.219.254.205.199.064.211.255
.205.199.064.205.187.002.044.200
.033.205.064.052.203.102.040.000
.024.193.062.100.061.032.253.201
.062.100.061.032.253.201.000."
   4 LET F=16514
   5 FOR A=1 TO 304 STEP 4
   6 POKE F,VAL A$(A TO A+2)
   7 LET F=F+1
   8 NEXT A
  10 PRINT TAB 5;" ORGANO Y EFEC
TOS "
  11 PRINT ,,"COMANDOS:"
  20 PRINT ,,"1  OSCILADOR NORMA
L"
  30 PRINT "2  VIBRATO NO"
  40 PRINT "3  VIBRATO SI"
  50 PRINT ,,"SHIFT 1  VIBRATO 1
"
  60 PRINT "SHIFT 2  VIBRATO 2"
  70 PRINT "SHIFT 3  VIBRATO 3"
  80 PRINT ,,"4  EFECTO DE (SUBI
DA O BAJADA)"
  90 PRINT "5  ACTIVA OSCILADOR
2  MODO 1"
 100 PRINT "6  ACTIVA OSCILADOR
2  MODO 2"
 110 PRINT "7  DESACTIVA OSCILAD
OR 2"
 120 PRINT ,,TAB 6;"8  MEZCLA 1"
,"SHIFT 8  MEZCLA 2",TAB 6;"9  M
EZCLA 3"
 130 PRINT ,,TAB 7;"PULSE TECLA"
 140 PAUSE 4E4
 150 CLS
 160 PRINT "COMANDOS:"
 170 PRINT ,,"SHIFT 4  OSCILADOR
 1 ARRIBA"
 180 PRINT "SHIFT 5  OSCILADOR 1
 ABAJO"
 190 PRINT ,,"SHIFT 6  OSCILADOR
 2 ARRIBA"
 200 PRINT "SHIFT 7  OSCILADOR 2
 ABAJO",,"
 210 PRINT AT 10,0;"LAS DOS HILE
RAS DE TECLAS","CENTRALES ACTUAN
 DE ORGANO"
 220 PRINT AT 14,0;"A=LA  S=SI
D=DO  F=RE  G=MI","H=FA  J=SOL
K=LA  L=SI (ETC.)"
 230 PRINT AT 20,0;"PULSE TECLA
PARA COMENZAR"
 240 PAUSE 4E4
 250 CLS
 260 RAND USR 16514
 270 LET X$=INKEY$
 280 IF X$="" THEN GOTO 270
 290 IF CODE X$=118 THEN POKE F,
87
 300 IF X$>"9" AND X$<"RND" THEN
 GOTO 510
 310 IF X$="1" THEN POKE 16515,1
1
 320 IF X$="0" THEN POKE 16515,2
1
 330 IF X$="2" THEN POKE 16574,0
 340 IF X$="3" THEN POKE 16574,2
48
 350 IF X$="4" THEN POKE 16515,0
 360 IF X$="5" THEN POKE 16538,5
 370 IF X$="6" THEN POKE 16538,1
 380 IF X$="7" THEN POKE 16538,2
3
 390 IF X$="8" THEN POKE 16549,1
18
 400 IF X$="9" THEN POKE 16549,8
6
 410 LET X=CODE X$
 420 IF X=117 THEN POKE 16572,11
8
 430 IF X=218 THEN POKE 16572,10
2
 440 IF X=222 THEN POKE 16572,86
 450 IF X=223 THEN POKE 16526,82
 460 IF X=114 THEN POKE 16526,53
 470 IF X=113 THEN POKE 16543,52
 480 IF X=112 THEN POKE 16543,53
 490 IF X=115 THEN POKE 16549,10
2
 500 GOTO 260
 510 LET F=16578
 520 IF PEEK 16515<>11 THEN LET
F=16520
 530 IF X$="A" THEN POKE F,245
 540 IF X$="S" THEN POKE F,218
 550 IF X$="D" THEN POKE F,200
 560 IF X$="F" THEN POKE F,175
 570 IF X$="G" THEN POKE F,155
 580 IF X$="H" THEN POKE F,145
 590 IF X$="J" THEN POKE F,127
 600 IF X$="K" THEN POKE F,110
 610 IF X$="L" THEN POKE F,95
 620 IF X$="Ñ" THEN POKE F,75
 630 IF X$="W" THEN POKE F,66
 640 IF X$="E" THEN POKE F,61
 650 IF X$="R" THEN POKE F,51
 660 IF X$="T" THEN POKE F,44
 670 IF X$="Y" THEN POKE F,37
 680 IF X$="U" THEN POKE F,34
 690 IF X$="I" THEN POKE F,30
 700 IF X$="O" THEN POKE F,27
 710 IF X$="P" THEN POKE F,20
 720 GOTO 260
 730 SAVE "ORGANO Y EFECTOS"
 740 RUN
 750 REM    DANIEL 5/1985 .
 760 STOP
```

DIAGRAMACION

1	Soporte del lenguaje de máquina
3 / 8	Coloca el lenguaje de máquina en la línea 1.
6	Transfiere de a uno los números de la línea 3
10 / 140	Menú 1
150 / 240	Menú 2
260	Llama a rutina en lenguaje de máquina
270 / 280	Esperan que se pulse una tecla
300	Si la tecla apretada es una nota salta a la parte final para su descodificación
370 / 490	Descodifican teclas "función"
510 / 520	Elige adonde diablos meto la nota, según la función elegida (una pavada que me llevó un montón de tiempo poder hacerla)
530 / 720	Descodifica las notas dándole la frecuencia correcta
730 / 740	Graba el programa para que se auto ejecute

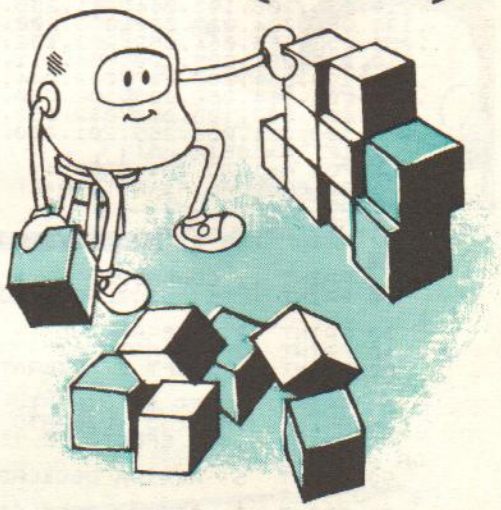
TABLA DE VARIABLES

A$	Contiene los números del *L.M*
F	Primero contiene la dirección de memoria que coloca el LM En el final, la dirección de memoria que coloca la nota elegida
A	La parte de A$ que se explorará
X$	Toma el valor de INKEY$, para su posterior descodificación
X	Contiene la codificación de X$, para usar por ejemplo las flechas con *SHIFT*

Softinotas

1. Esta muy buena idea de ***Daniel*** puede aplicarse a todas las máquinas "Menores" de la línea Sinclair.
2. Lamento no poder publicar la estructura del programa en lenguaje de máquina pues me has mandado una copia poco legible, ***Daniel.***
3. Nuestros lectores te agradecerán seguramente que nos hayas enviado la estructura del programa y la tabla de variables.
4. Como habrás visto, ***Daniel,*** las pantallas del programa las tuve que "retocar", así como el texto de tu carta.
5. Te felicito efusivamente, ***Daniel*** por tu idea y esfuerzos (va cheque por correo)... ***Softi***

TABLAS (II)



***SOFT!* continúa dándonos herramientas para aprovechar al máximo nuestra microcomputadora. En este número nos desarrolla las TABLAS ALFANUMERICAS.**

En el número dos, hablamos de las tablas numéricas, tanto de una, como de dos dimensiones. Hoy vamos a hablar de las *tablas alfanuméricas.*

El *nombre* de una tabla alfanumérica se define con *una sola letra* seguida del signo $ (por ej.: A$)

Tablas alfanuméricas de una dimensión (Unidimensional). . .

A diferencia de las numéricas, en una tabla alfanumérica de una dimensión, cada elemento tiene una *longitud fija* y de *un solo carácter*. O sea, dentro de un elemento (o item), no "entra" una palabra o frase.

NOTA: *Hay algunos equipos que no tienen esta limitación pero son equipos grandes.*

Veamos una representación gráfica. . .

Tomemos, por ejemplo, la palabra "ENERO".

Recordemos que el nombre de la tabla debe ser una letra seguida del signo $.

información	E	N	E	R	O
subíndice	1	2	3	4	5

Esta es la *tabla* A$

–Aquí vemos la tabla **A$** compuesta de 5 items.
–Cada item contiene una letra.
–El conjunto de los items de la tabla forman una palabra
–Su definición en BASIC es **DIM A$ (5)**

En este tipo de tablas, a diferencia de las numéricas, podemos mencionar a la tabla en su conjunto. ¿¿¿Cómo???, muy sencillo, no utilizando el subíndice.

De esta manera decimos que mientras A$(1) es igual a "E", A$ es igual a "ENERO"; en este último caso utilizamos la tabla como una *String*, de la cual también podemos obtener Sub-strings (subdivisiones), entonces por ejemplo: A$ (1 TO 3) es igual a "ENE", las tres primeras letras de la tabla.

Resumamos. . .

```
A$ (1)  = "E"
   A$   = "ENERO"
A$ (1 TO 3) = "ENE"
```

Tablas Alfanuméricas de dos dimensiones (Bidimensional). . .

Mediante el uso de este tipo de tablas, tenemos la posibilidad de fijar *longitudes* a los elementos y exceder la longitud de 1 caracter.

Veamos como. . .

En el siguiente gráfico, se encuentran los nombres de los primeros seis meses del año:

Columna / Fila	1	2	3	4	5	6	7
1	E	N	E	R	O		
2	F	E	B	R	E	R	O
3	M	A	R	Z	O		
4	A	B	R	I	L		
5	M	A	Y	O			
6	J	U	N	I	O		

De esta manera, decimos que. . . la tabla está compuesta de *6 filas* de *7 columnas* cada una. Para definirla en BASIC es:

DIM A$ (6,7)

Tenemos la tabla A$ de 6 por 7 elementos y, al igual que las numéricas, podemos acceder a un elemento mediante dos coordenadas (subíndices). Por ejemplo, para acceder a la letra "Z" de la palabra "MARZO", sería A$ (3,4), puesto que en la fila 3 columna 4 se encuentra la letra "Z".

Ahora veamos una diferencia fundamental con las tablas numéricas. . .

En tablas alfanuméricas, podemos utilizar sólo un subíndice (fila) y acceder así a una palabra o cadenas de caracteres, o sea basándonos en esta definición y el gráfico anterior tenemos que A$ (3) es igual a "MARZO".

Como vemos podemos mencionar a la fila sin mencionar la columna. Obtenemos así una String de la cual podemos también obtener Sub-strings. ¿¿¿Cómo???. . ., reemplazando el segundo subíndice (columna), por una expresión "n TO n". Siguiendo con el gráfico anterior...

A$ (3, 1 TO 3) es igual a "MAR"

El 3 es la "fila" y "1 TO 3" es la expresión que nos indica de dónde a dónde de la String queremos utilizar (en este caso los caracteres ubicados en la posición 1 al 3 del string).

Veamos un programa por ejemplo. . .

Pensemos que partiendo de una fecha como por ejemplo, 220385, la deseamos imprimier en la pantalla como 22 de marzo de 1985. ¿Cómo hacerlo? ? ?. . .

1. Como primer paso **definamos una tabla alfanumérica** de dos dimensiones, dentro de la cual ingresaremos el nombre de cada mes. La tabla estará compuesta de 12 filas de 10 columnas cada una (10 es el número máximo de letras del mes de más letras).

```
10 DIM A$(12,10)
```

2. Ahora **ingresamos los nombres de los meses** (observen que sólo mencionamos un subíndice, la "fila").

```
20 LET A$(1)="  ENERO"
30 LET A$(2)=" FEBRERO"
40 LET A$(3)="  MARZO"
50 LET A$(4)="  ABRIL"
60 LET A$(5)="   MAYO"
70 LET A$(6)="  JUNIO"
80 LET A$(7)="  JULIO"
90 LET A$(8)="  AGOSTO"
100 LET A$(9)="SEPTIEMBRE"
110 LET A$(10)=" OCTUBRE"
120 LET A$(11)="NOVIEMBRE"
130 LET A$(12)="DICIEMBRE"
```

NOTA: *Aunque no ingresemos 10 posiciones, cada elemento es es completado con espacios.*

3. Ahora **ingresamos la fecha. . .**

```
140 PRINT "INGRESE FECHA (DDMMA
A)"
150 INPUT F$
```

De esta manera tenemos en las dos primeras posiciones de F$ el DIA, en las dos siguientes el MES y en las dos últimas el AÑO.

4. El mes lo usamos como subíndice para **obtener el nombre** correspondiente; por lo tanto debemos desglosar la variable F$, colocando en variables numéricas DIA, MES y AÑO.

```
160 LET DD=VAL F$(1 TO 2)
170 LET MM=VAL F$(3 TO 4)
180 LET AA=VAL F$(5 TO 6)
```

NOTA: *Para mover de una variable alfanumérica a una numérica se debe usar la función* VAL.

5. . . . **imprimimos** la fecha. . .

```
190 CLS
200 PRINT DD;" DE ";A$(MM);" DE
19";AA
```

En la variable DD se encuentra el Día, luego imprimimos como literal la palabra "DE": A$ (MM) nos indica el nombre del mes utilizando a MM (el mes) como subíndice; luego imprimimos como el literal "DE 19" y a continuación el año (AA)

Entonces, ¿qué esperamos para probar lo visto hasta aquí? ? ? . . .



RECOMPENSA U$S 1.000 U$S DOLARES

A quien descifre el mensaje codificado por el programa "Caja de Seguridad", del TOMO IV de la colección "Juegos y entretenimientos para Microcomputadora Basic".

eci

RIVADAVIA 14056 - Piso 1° Of. 5
Tel. 658 - 4161
Ramos Mejía
EDITORIAL CONSULTORA EN INFORMATICA

MANEJO DE FECHAS (III)

Cuando nosotros ingresamos un dato, por ejemplo, a una variable y nos interesa que el mismo cumpla determinados requisitos, lo que tenemos que hacer es "validarlo", o sea, verificar a través de una rutina de programa que dicha variable sea "lógica" y esté dentro de los parámetros y condiciones aceptables previstos.

Validación de Fecha Lógica

En "Manejo de fechas" en números anteriores decíamos (ver Nº 2, página 32): "Por tratarse de una rutina se asume que la fecha es válida y lógica". Pues bien, ya es hora de mostrar *cómo determinar* si una fecha es lógica (válida) o no.

Una fecha **no** es lógica cuando por ejemplo es; 300285, ya que febrero no tiene 30 días.

Qué requiere una fecha para considerarse lógica?

1) Tener *6 dígitos*, **(DDMMAA)**
2) que sea *numérica* y *positiva*
3) y que sus componentes cumplan, por lo menos, con lo siguiente:

- *Año:* distinto de cero
- *Mes:* distinto de cero y no mayor a 12
- *Día:* distinto de cero y no mayor a la cantidad de días del mes asociado (caso especial: febrero en años bisiestos)

TK83/85/90X/TI99/TS2068/ CZ1000 - 1500 - Spectrum

```
   10 REM SOFTI 1985
   20 REM VALIDACION DE FECHA
   30 PRINT "INGRESE FECHA: ";
   40 INPUT F$
   45 PRINT F$
   50 GOSUB 5000
   55 SLOW
   60 IF E=1 THEN GOTO 90
   70 PRINT "FECHA CORRECTA"
   80 STOP
   90 PRINT "FECHA ERRONEA,PULSE
1"
  100 IF INKEY$="" THEN GOTO 100
  110 CLS
  120 GOTO 10
 5000 LET A$="312831303130313131303
13031"
 5010 LET E=0
 5020 FAST
 5030 IF LEN F$<>6 THEN GOTO 6000
 5040 FOR I=1 TO 6
 5050 IF F$(I)<"0" OR F$(I)>"9" T
HEN GOTO 6000
 5060 NEXT I
 5080 LET DD=VAL (F$(1 TO 2))
 5090 LET MM=VAL (F$(3 TO 4))
 5100 LET AA=VAL (F$(5 TO 6))
 5110 IF AA=0 THEN GOTO 6000
 5120 IF MM=0 OR MM>12 THEN GOTO
6000
 5130 IF DD=0 THEN GOTO 6000
 5140 IF MM=2 THEN GOTO 5170
 5150 IF DD>VAL (A$(MM*2-1 TO MM*
2)) THEN GOTO 6000
 5160 RETURN
 5170 IF AA=INT (AA/4)*4 THEN GOT
O 5200
 5180 IF DD>28 THEN GOTO 6000
 5190 RETURN
 5200 IF DD>29 THEN GOTO 6000
 5210 RETURN
 6000 LET E=1
 6010 RETURN
```

Esta vez el problema de *Softi* es establecer si una fecha es "buena" o "rechazable" Qué lío, no???...

Para correr en TS2068 hay que omitir líneas: 55 y 5020.

La ejecución es muy simple, después de cargado el programa y dado "RUN"..., solicitará ingreso de fecha.

Luego de ingresada, validará y exhibirá el resultado.

Por supuesto podríamos hacer que la rutina nos informe qué tipo de error tiene la fecha incorrecta, pero eso se los dejo para ustedes. Se animan???...

Comentario sobre las líneas del programa

10 / 20	Comentarios
30 / 45	Solicta fecha y exhibe la misma
50	GOSUB a la rutina de validación
60	Mediante la variable "E" determina si hubo o no error
70	Comunica que la fecha es correcta
80	Stop
90 / 120	Comunica que la fecha es errónea, recicla
5000 / 6000	**Rutina de validación de fecha** propiamente dicha
5000 / 5010	Inicializa variables
5030	Valida que F$ contenga 6 dígitos
5040 / 5060	Valida que cada dígito sea numérico
5080 / 5100	Divide F$ en DD,MM y AA
5110 / 5130	Valida que el año no sea cero, el mes ni cero ni mayor a 12, y el día distinto de cero
5140	Si el mes es 2 (febrero), va a ver si es bisiesto
5150 / 5160	Valida que el día no sea mayor a la cantidad de días del mes correspondiente. Return.
5170 / 5210	Si es bisiesto va a 5200, por el contrario va a 5180, en ambas valida día, no mayor a 29 ó 28 según corresponda.
6000 / 6100	Determina error (E=1) Return.

(continúa en pág. 29)

FICHA TECNICA
ZX MICRODRIVES SPECTRUM E INTERFASE 1

El **Microdrive** tiene unas *dimensiones externas* de 90 x 85 x 40 mm. En la parte delantera presenta la abertura frontal para la carga del cartucho, y en la trasera dos conectores sobre una de las dos tarjetas de circuito impreso que aloja. La otra tarjeta, vertical, lleva la ROM y la cabeza que hace las funciones de lectura y grabación en la cinta. Todos los elementos mecánicos de la unidad son sencillos y robustos. La única preocupación podría ser el posible desgaste de la cabeza debido a la acción abrasiva de la cinta.

El pequeño cartucho **ZX Microdrive** (compárense sus dimensiones de 45 x 35 x 7 mm. con los 75 mm. de un diskette) contiene una banda continua de alta calidad, que mide aproximadamente 15 metros y tiene una anchura de 1,9 mm. Pese a las prometidas 100 K, su *capacidad* garantizada es de 85 K, aunque en la práctica suele resultar ligeramente superior (normalmente 89 ó 90 K). Pueden montarse simultáneamente hasta un *máximo de 8 unidades*, con lo que se dispone de una *capacidad total* de 680 K o superior. Los Microdrives se alimentan de la fuente del Spectrum, y se conectan a él a través de una pequeña tarjeta que se ha denominado **ZX Interface 1**. Esta tarjeta entra sencillamente en el conector de la parte de atrás del Microordenador. Dos de los tornillos de la base del **Spectrum** se sustituyen por otros dos incorporados a la interfase y que sirven para que quede *firmemente asegurada*. Esta vez, por tanto, no va a pasar como con el famoso módulo de RAM del **ZX81**, cuya insegura conexión daba lugar a toda clase de problemas. Por el contrario, tiene la ventaja añadida de levantar ligeramente el teclado del **Spectrum**, dejándolo en un ángulo más cómodo. Quienes hayan sustituido el teclado original por un modelo "propio" juegan en este caso con desventaja.

Gracias a la Interfase I, el **Spectrum** puede gobernar una *impresora RS232*. Asimismo, gracias a esta tarjeta, ahora es posible crear una **red** de hasta 64 **Spectrum** enlazados La interfase RS232 opera en un margen de 50 a 19.200 baudios, sin paridad, ocho bits por carácter y dos bits de stop.

Existen dos *modos de operación*. El *canal t* se utiliza normalmente para listados; los códigos de control no se envían (a excepción del 13, retorno de carro), y los códigos característicos se expanden. No dispone de capacidad gráfica; todos estos códigos se sustituyen por 63. El *canal b* envía códigos de ocho bits completos y se utiliza precisamente para mandar caracteres de control a impresoras, etcétera.

La **ZX NET** *(red en área local)* promete ser una herramienta muy poderosa. La red se establece simplemente uniendo dos interfases de **Spectrum** con un cable de 10 pies. El manual habla de una red funcionando a 100 Kbaudios con handshaking entre dos equipos conectados. Pueden enlazarse hasta un *máximo de 64 Spectrum*. Existe también una función de comunicación para enviar información a cualquiera que esté conectado a la red. Gracias a esta posibilidad, un profesor podría enviar, por ejemplo, un programa simultáneamente a toda una clase. Otro uso podría ser el establecimiento de una estación impresora compartida por todos los **Spectrum** de la red.

No obstante, la principal función de la **Interface I**, y en la que nos encontraremos aquí, es la de controlar los **Microdrives.**

Cada unidad se conecta a la interfase mediante un cable plano de corta longitud. Las unidades adicionales se conectan cada una a la siguiente mediante conectores rígidos (con cada Microdrive viene uno de estos conectores), fijándose mediante una abrazadera que se *atornilla* en la base de cada unidad.

La interfase *amplía el BASIC* de **Sinclair** para que incluya las necesarias funciones de manejo de ficheros y de comunicaciones, y utiliza el *BASIC como un sistema operativo.* El manual del Microdrive y la Interfase 1 da todo tipo de detalles sobre las ampliaciones del lenguaje que se han introducido y sobre su utilización. Incluye también los nuevos códigos de error asociados con los Microdrives. Cuando ha sido posible se ha mantenido la sintaxis de las instrucciones precisas para indicar que es el Microdrive al que se refieren.

COMO SE UTILIZA EL MICRODRIVE

Una vez conectado un Microdrive con el cable plano a la tarjeta de interfase y ésta al **Spectrum** la primera tarea consiste en *formatear los cartuchos* vírgenes. La manera de hacerlo es introducir sencillamente FORMAT "m"; 1; "nombre", donde "m"; 1 identifica la unidad que se está utilizando (en este caso Microdrive 1), y "nombre" puede ser cualquiera que tenga 10 caracteres o menos. El formateado *tarda* unos 30 segundos, durante los cuales la pantalla se ilumina y se borra alternativamente hasta que aparece el OK que indica el fin de la operación. El formateado de un soporte es esencial para que el ordenador identifique las áreas en las que lee, escribe o marca como no utilizables.

Cuando se introduce CAT 1 (seguimos en el supuesto de que se está utilizando la unidad 1), aparecerá en pantalla el nombre que se ha dado al cartucho, los nom-

bres de los ficheros que contiene y el espacio no ocupado de que se dispone.

La *protección contra escritura* se logra quitando una pestaña de plástico, y, al igual que con los cassettes, para poder grabar de nuevo se utiliza el sencillo procedimiento de tapar el hueco con un trocito de cinta adhesiva.

Para *guardar un programa* en el Microdrive hay que dar la orden SAVE* "m"; "nombre" (el asterisco indica al microordenador que la instrucción se refiere al Microdrive y no al cassette, y "m"; 1 identifica el número de unidad. La comprobación de que el almacenamiento de un programa ha sido correcto se hace con VERIFY* "m ; 1; "nombre", y la instrucción de carga desde el Microdrive es LOAD* "m"; 1; "nombre".

Se pueden *mezclar* programas con MERGE* "m"; 1; "nombre", así como utilizar la posibilidad de auto ejecución usual del Spectrum añadiendo LINE x a la orden SAVE.

El Microdrive tiene su propia *función de carga y ejecución* conjuntas, es decir que basta con escribir RUN y no LOAD y RUN cuando se quiere ejecutar un programa residente en el Microdrive.

Para ello es preciso que el programa en cuestión se llame precisamente "run" –tecleándolo letra a letra– y que el cartucho esté en la unidad 1. Esta posibilidad hay que utilizarla inmediatamente después de conectar el aparato o de haber hecho un NEW. El programa se guarda con la orden SAVE "m"; 1; "run" LINE número. Un RUN dado ahora desde el teclado, cargará y ejecutará el programa. Debe observarse que la instrucción MERGE no funciona con programas que se hayan guardado utilizando LINE.

Aunque lo anterior se refiere a la transferencia de programas en BASIC, las rutinas en *lenguaje de máquina* se manejan también fácilmente en la manera usual, añadiendo CODE al final de la correspondiente instrucción. Un detalle bastante molesto es que la lista de ficheros que proporciona CAT no hace distinción entre programas en código de máquina y programas en BASIC. Con el cassette, por el contrario, el BASIC de **Sinclair** proporciona una serie de prefijos muy prácticos, como "Programa", "Bytes", "Matriz numérica", etcétera.

Borrar un programa o un fichero es lo más sencillo del mundo; basta con introducir ERASE "M"; 1; "nombre".

Si se hace un BREAK mientras se está guardando un programa en el cartucho quedará un fichero sin cerrar, y cualquier intento de cargarlo terminará en un poco reconfortante mensaje de "file not found" (no se encuentr el fichero).

La instrucción ERASE borra también este tipo de ficheros no cerrados, pero el micro tarda unos 30 segundos en hacer varias comprobaciones del cartucho, en un intento de buscar el inexistente final.

En general, hay que tener cuidado de no sacar el cartucho de la unidad mientras el LED testigo esté encendido. Asimismo, tampoco es nada recomendable conectar o desconectar la alimentación sin retirar el cartucho del Microdrive.

CANALES DE DATOS y Streams

Todo lo dicho hasta ahora se refiere a la utilización del Microdrive para el almacenamiento de programas. Sin embargo, el aspecto más interesante de la nueva unidad no es éste, sino su uso en el *almacenamiento y recuperación de datos* desde los programas.

En relación con esta función se introducen los conceptos de *canales* y de *streams*. Al hablar de *canales*, nos estamos refiriendo a los diversos dispositivos componentes de un sistema. Así diremos que los datos pueden *enviarse* a la pantalla, a la impresora ZX, al Microdrive, a otro **Spectrum** si ambas máquinas están conectadas en una red, o a la interfase RS232 y desde aquí a una impresora u otro periférico. De otra parte, los datos pueden *provenir* de los siguientes canales; el teclado, un fichero del Microdrive, otro **Spectrum**, o, a través de la interfase RS232, de un modem o un terminal.



Los datos se envían y se reciben por estos canales por medio de *streams* (llamémoslas *rutas*). Este concepto resultará ya familiar a los usuarios del **Spectrum**. El Manual de Introducción al BASIC da algunos detalles al respecto, y los usuarios quizá hayan utilizado streams para imprimir en las dos últimas líneas de la pantalla, por ejemplo.

El **Spectrum** es capaz de utilizar 16 rutas, numeradas de 0 a 15 (estos números se utilizan siempre precedidos del signo #).Cuatro rutas están permanentemente asociadas a ciertos canales.

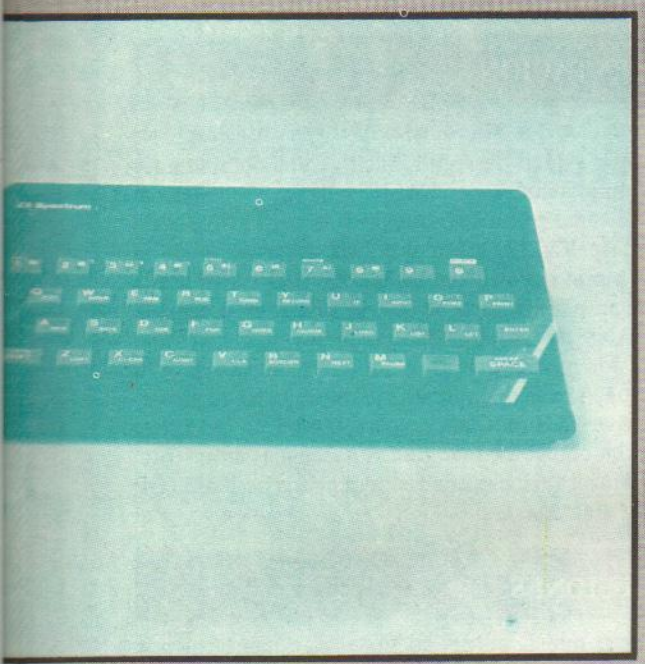
Así, las rutas #0 y #1 envían datos a la parte inferior de la pantalla y sirven de entrada desde el teclado; la ruta # 2 actúa de salida a la parte superior de la pantalla, pero no puede ser entrada; la ruta # 3, finalmente, está asociada a la Impresora ZX y tampoco puede ser entrada.

Cada una de las instrucciones de entrada o salida utiliza una de estas rutas. Las sentencias PRINT y LPRINT

son, de hecho, expresiones abreviadas de PRINT# 2 y PRINT# 3 respectivamente. Una instrucción puede hacer uso de una ruta diferente sin más que especificar el número correspondiente LPRINT# 2, por ejemplo, dirigiría la salida a la pantalla en lugar de hacerlo a la impresora.

El usuario no está limitado a utilizar estas cuatro rutas "predefinidas". Los números #4 a # 15 están disponibles para cualquier trabajo. Con los especificadores de canal, el usuario determina el periférico que desea. Así, "K" para el teclado, "S" la pantalla, "P" la impresora ZX, "T" para texto (utilizado con el RS232), "B" para binario (también RS232), "N" para red y "M" para Microdrive.

Para poder utilizar estas rutas y estos periféricos, se precisa de entrada una instrucción OPEN#.OPEN# 4, "S" abre, por ejemplo, la ruta # 4 y la asocia al canal S (la pantalla). Si se ejecuta ahora PRINT# 4; "texto", es-



te texto aparecerá en la pantalla. Obsérvese que K, S y P son canales establecidos y requieren que se utilicen comas como separadores en las instrucciones OPEN #. Los otros dispositivos admiten que el separador son "," o ";".

Visto esto, pasemos ya a ver *cómo se utilizan estas rutas y canales.*

Después de *abrir un fichero de datos* con la instrucción, por ejemplo, OPEN #4; "m"; 1; "números" (habilitando así un nuevo canal "m"; 1; "números" y asociándolo con la ruta # 4), ya pueden introducirse los datos. El manual pone como ejemplo un fichero de datos que contiene los números 1 al 10 y sus cuadrados. Este programita:

```
10 FOR N = 1 TO 10
20 PRINT 4, N, N*N
30 NEXT N
```

produce los datos que se requieran. A primera vista podría suponerse que basta con este programa para *grabar en el Microdrive* los números generados, pero es preciso decir que el microordenador no transfiere nada hasta que está lleno el buffer del Microdrive (tiene 512 bytes) o se ejecuta una instrucción CLOSE. En nuestro caso, pues, la grabación no se consumaría si no añadimos una orden CLOSE#4. Al igual que para OPEN, no deben cerrarse las rutas 0-3.

Para *recuperar ahora los datos almacenados* en el fichero "números" del Microdrive, se podría utilizar el siguiente programa:

```
10 OPEN#4; "m"; 1; "números"
20 FOR b = 1 TO 10
30 INPUT4; m, n
40 PRINT "el cuadrado de"; m; "es"; n
50 NEXT b
60 CLOSE#4
```

También se puede utilizar INKEY$ para *leer de un fichero*. Esta función devuelve siempre el siguiente carácter.

Un aspecto a resaltar cuando se utiliza INPUT, es que el ordenador espera siempre un ENTER después de un número o una cadena de caracteres. Así pues, a la hora de grabar los datos, cada item que luego se quiera leer con INPUT ha de escribirse independiente.

Hay que tener cuidado cuando se trata de introducir una cadena de caracteres que contenga comillas. La manera más segura de manejarlas es utilizando LINE a$ en lugar de simplemente a$.

Después de utilizar un canal que no sea la pantalla, puede comprobarse que las órdenes PAPER e INK no funcionan. Esto se arregla introduciendo PRINT antes de fijar los colores.

Los ficheros quedan automáticamente *catalogados* (hasta 50 por cartucho) y la lista que se obtiene con CAT puede sacarse por una ruta cualquiera, lo que permite, por ejemplo, imprimir el contenido de un determinado cartucho.

Para *"ocultar" un fichero*, esto es, para que no salga en la lista que proporciona CAT, basta con añadir CHR$(80) a su nombre. Así por ejemplo, con la instrucción OPEN# 4; "m"; 1; CHR$ (80) + "PCW" nos aseguramos de que el fichero "PCW" no va a aparecer en el directorio del cartucho que estamos utilizando.

Cuando se excede el límite de 50 ficheros por cartucho, los nombres de los ficheros empiezan a desaparecer de una manera aparentemente aleatoria, aunque continúan ocupando espacio y todavía se pueden cargar.

Una importante instrucción dirigida al Microdrive es MOVE. Como su nombre sugiere, permite *trasladar los datos de un canal a otro*. Para trasladar datos del teclado a la pantalla, por ejemplo, se podría escribir MOVE #1 TO #2. Cualquier cosa que ahora se teclee, aparecerá en la pantalla del televisor.

MOVE puede utilizarse para *examinar el contenido de un fichero*; MOVE "m"; 1; "números" TO # 2, sacará por pantalla el contenido del fichero "números". MOVE también permite copiar ficheros:

Tabla de formato de los comandos del microdrive

LOAD* "M"; 1; "NOMBRE"	El * indica la carga a partir de un microdrive.
SAVE * "M"; 1; "NOMBRE"	
LINE 10	Establece el nombre para el comienzo automático del mismo. El nº de línea es opcional.
VERIFY* "M"; 1; "NOMBRE"	
MERGE* "M"; 1; "NOMBRE	No mezclará programas guardados con un número de línea.
ERASE "M"; 1; "NOMBRE"	
FORMAT "M"; 1; "TITULO"	Título se refiere a una cabecera para el cartucho.

Tabla de los comandos del Basic extendido

CAT Y	Y es el número del Microdrive de 1 a 8.
CAT #Z; Y	Z es el número de corriente[1] de 0 a 15.
CLOSE #Z	
ERASE "M"; Y; "NOMBRE"	NOMBRE es el nombre de un fichero.
FORMAT "M"; Y; "TITULO"	TITULO es el nombre de un cartucho.
FORMAT "N"; X	N significa red (network) y X estación número 0 a 64.
FORMAT "T"; S	T indica RS232 texto y S es la velocidad de transferencia en baudios.
FORMAT "B"; S	B es para RS232 para datos binarios.
INKEY$ #Z	Devuelve un caracter o nulo
INPUT #Z; Var	Introduce una variable.
LOAD * CHAN	Carga programas, datos o ficheros de canales B, M o N.
MERGE * CHAN	
MOVE desde TO destino	Solamente desplaza datos
OPEN #Z; CHAN	Enlaza corriente a canal.
PRINT #Z	Da salida a una corriente especificada.
SAVE*; CHAN	Guarda (save) programas, datos o codificaciones solamente en canal B, M o N.
VERIFY *; CHAN	

1 **Corriente:** *La rutina que va de un canal a otro es llamada una corriente. En un sistema basado en Spectrum existen 16 corrientes.* **Canales** *(Channels): Los datos pueden ser enviados a diferentes dispositivos, la pantalla, la impresora ZX, fichero en Microdrive, otro Spectrum de la misma red y el interfase RS232. Los datos pueden ser recibidos desde los siguientes dispositivos: el teclado, un fichero en Microdrive, otro Spectrum de la misma red y el interfase RS232. Estos dispositivos son conocidos como canales.*

MOVE "m"; 1; "números" TO "m"; 1; "número 2" .

copiará el contenido del fichero "números" en el fichero "número 2". Cambiando simplemente el número de Microdrive, se pueden hacer *copias de seguridad* de ficheros de datos en otro cartucho. Por ejemplo:

MOVE "m"; 1; "números" TO "m"; 2; "número 2".

Este tipo de copias sólo sirve para fichero de datos. Para copiar programas es preciso cargarlos en el Spectrum y luego volver a guardarlos con SAVE.

TIEMPOS

Aunque no se trata de pruebas exhaustivas de laboratorio, las cifras que aportamos pueden muy bien servir de indicación comparativa de prestaciones del cassette y el Microdrive. Tratándose de una cinta magnética, la modalidad de acceso a la información de que puede hablarse es del *acceso secuencial*, más que del auténticamente aleatorio de los discos. Para los usuarios de **Spectrum** que previamente han trabajado con cassettes, el aumento de velocidad resultará sencillamente formidable, aunque los que estén acostumbrados a los discos quizá no les cause demasiado placer los tiempos de espera para la localización y carga de un fichero. El *acceso secuencial* supone que los tiempos de búsqueda pueden ser incluso superior a la propia carga. El tiempo máximo de búsqueda es de 3,5 segundos, dependiendo lógicamente en cada caso, de la posición del fichero en el soporte.

Se han utilizado para las pruebas de velocidad una selección de programas en BASIC y en lenguaje de máquina. Primero se cargaron desde el cassette, anotándose el tiempo empleado, se volvieron a guardar en el Microdrive, para, finalmente, pasarlos de nuevo al **Spectrum**. Como era de esperar, las diferencias de tiempo más notables se dieron con programas largos; un programa de juegos de 35 k, por ejemplo, cuya carga desde cassette toma 3 minutos 25 segundos, tardó algo menos de 8 segundos desde el Microdrive.

TABLA DE TIEMPOS

	Cassette	Microdrive
Programa Basic (35 K) . .	3.25.0	7,8 seg.
Programa en cod. máquina (7000 bytes).	40.0	7,5 seg.
Datos (DIM a (4000)). . .	1.21.30	6,8 seg.
SCREEN$	32.0	3,8 seg.

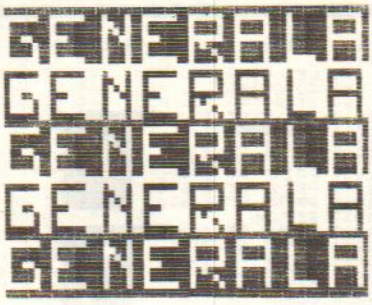
DOCUMENTACION

El *manual* de los **Microdrives** y de la **Interfase 1** es comprensible y fácil de seguir. Incluye una completa lista de las ampliaciones del BASIC y de las variables del sistema utilizadas por los Microdrives, la red en área local y la interfase RS232, junto con una descripción de la RAM de usuario que utilizan los nuevos periféricos. Cada vez que se abre un fichero, se reserva una zona designada como CHANS de 595 bytes (que incluyen los 512 del buffer del Microdrive). Para la red, este área es de 276 bytes. Es poco probable que ésto dé problemas ni siquiera a los programas más largos, pero tendrá efecto sobre las rutinas en código de máquina almacenadas en instrucciones REM (las rutinas situadas por encima de RAMTOP no se verán afectadas).

CONCLUSIONES

Aunque el nuevo periférico tiene ligeramente menos capacidad, cuesta un poco más caro y es algo más lento que lo esperado, podemos decir que con el **Microdrive, Sinclair** ha conseguido que por primera vez los propietarios de un **home computer** tengan acceso a una memoria masiva de bajo costo y con una velocidad de transferencia considerable, comparativamente hablando. Esto supone la apertura de una puerta hacia un montón de aplicaciones "serias" para el **Spectrum** y va a tener el efecto lateral de alargar la vida de esta máquina en el mercado.

Conectado el Microdrive, el **Spectrum** puede utilizarse como un pequeño equipo de gestión, que dispondrá de aplicaciones incorporadas como pueden ser el tratamiento de textos y la hoja electrónica de datos. Pero, lo que es más importante, los Microdrives van a ser sin duda decisivos a la hora de fijar un standard en almacenamiento de datos para microordenadores pequeños y baratos. **Sinclair** ha logrado ser un importante factor en la difusión de la microinformática entre el gran publico, y el Microdrive no hace sino continuar en esta línea: Más potencia para sus proyectos.





Desde nuestra "República de Corrientes", **Jorge** nos envía este excelente programa para jugar en nuestras Sinclair "medianas" el difundido juego de dados "la generala".

Pero escuchemos a su autor:

Qué hace y para qué sirve

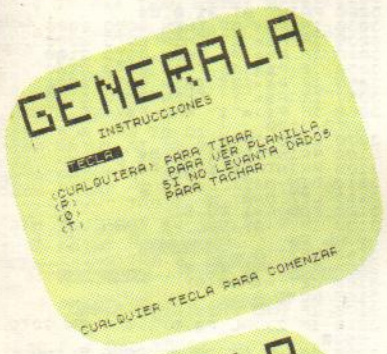
Se trata de un muy vistoso juego de "generala", en el que pueden participar de 1 a 5 jugadores, siguiendo las reglas de este conocido juego de dados.

Lo interesante es que no se pueden hacer trampas en ningún momento del partido. Es decir que la computadora es una "garantía de honestidad"

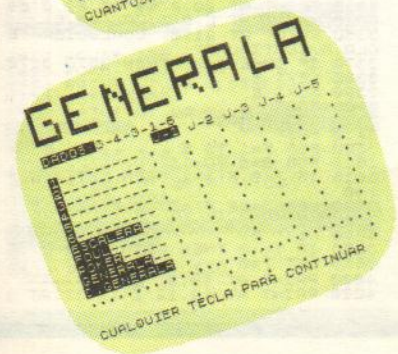
Aclaramos que no se juega "contra" la computadora, sino entre jugadores. Si fuera uno solo, sirve para un entretenimiento personal.

Instrucciones

Las reglas del juego son las ya conocidas de la "generala"







Estructura del Programa

1	Inversor de video (ver **Softinota**)
2 / 140	Presentación
145 / 148	Inicialización
170 / 180	Pregunta la cantidad de jugadores
250 / 253	Los dados toman valor
255 / 270	Presentación de los dados
280 / 310	Cantidad de dados que levanta y qué dados levanta
400 / 450	Bucle para cuando se quiere ver la planilla
1000 / 1067	Formación gráfica de los dados
1100 / 1153	Ubicación de los dados
2000 / 2350	Analiza los juegos
2400 / 2490	Indica la finalización del turno
2500 / 2610	Ordena los dados ascendentemente
3003 / 3060	Envía a la subrutina correspondiente para analizar
3100 / 3176	Analiza y suma la cantidad a anotarse
3200	Analiza si ya está anotado
3205 / 3271	Pregunta si se equivocó o no
3305 / 3370	Escribe el puntaje
3380	NEXT de los turnos
3390	NEXT del total de "vueltas" (11)
3600 / 3640	Saca los totales
4000 / 4055	Escribe los totales
4100 / 4150	Analiza quién es el ganador
4160 / 4270	Felicita al ganador
5000 / 5280	Hace la planilla de puntos
5300 / 5310	Tacha cuando hace la planilla
5400 / 5440	Indica que lo que pidió anotarse es incorrecto
5500 / 5590	Indica que ya tiene lo que quiso anotarse
5600 / 5640	Indica que ya está tachado lo que se quiso tachar
5700 / 5800	Tacha lo que se pide tachar
5805 / 5860	Escribe lo que quiere anotar
5900 / 5980	Forma una pantalla por la "generala servida"
6000 / 6152	Dá las instrucciones

Tabla de Variables

- L$ Palabra "Instrucciones"
- B$ Línea en blanco
- C$ Dados que levanta
- P$ Variables para tachar o anotar
- Z$ Cantidad de jugadores
- X$ Cantidad de dados que levanta
- K Dados
- Z Puntaje
- N Cantidad de tiros por cada jugador
- I - J Ubicación de los dados
- M Dados con el número que pidió anotarse
- H Puntaje que corresponde a lo que quiere anotarse
- L - O Ubicación del puntaje
- G Ciclo FOR/NEXT de diversa utilidad
- F Idem G
- E Ciclo FOR/NEXT de la cantidad de vueltas a jugarse
- B Ciclo FOR/NEXT de la cantidad de jugadores

CZ 1500 - TK 85

```
    1 REM ..........
    2 DIM L$(21,32)
    3 LET B$="
    5 LET L$(1)="
    6 LET L$(2)="
    7 LET L$(3)="
    8 LET L$(4)="
    9 REM  13 014 KBYTES OCUPADOS
   10 REM TIEMPO APROX. CARGA 5M.
40S.
   20 PRINT AT 5,10;"SOFTWARE GUG
LIELMONE",,TAB 12;"MAYO 1985",,T
AB 10;"------GENERALA------"
   25 PRINT AT 21,0;"CUALQUIER TE
CLA"
   27 RAND USR 16514
   30 IF INKEY$="" THEN GOTO 30
   31 CLS
   33 FOR F=1 TO 4
   34 FOR G=1 TO 15
   35 PRINT AT F-1,G-1;L$(F,G);AT
 F-1,31-G;L$(F,32-G)
   37 NEXT G
   38 NEXT F
   39 RAND USR 16514
   40 FOR G=1 TO 4
   50 PRINT L$(1);L$(2);L$(3);L$(
4)
   60 FOR F=1 TO 7
   65 RAND USR 16514
   70 FOR I=1 TO 4
   71 NEXT I
   75 NEXT F
   80 NEXT G
   90 FOR G=1 TO 21
   95 IF G>4 THEN LET L$(G,1 TO 3
6)="
  100 SCROLL
  105 PRINT L$(G)
  110 NEXT G
  120 LET P$="                INSTRUC
CIONES ? S/N"
  125 PRINT AT 14,0;P$
  130 LET P$=P$(2 TO )+P$(1)
  135 IF INKEY$="S" THEN GOTO 600
0
  140 IF INKEY$<>"N" THEN GOTO 12
5
  145 DIM K(5)
  146 DIM C$(5,1)
  147 DIM Z(12,5)
  148 LET T=0
  150 CLS
  160 PRINT L$(1);L$(2);L$(3);L$(
4)
  170 PRINT AT 10,0;"CUANTOS JUGA
DORES SON ? (1-5)"
  172 LET Z$=INKEY$
  174 IF Z$>"5" OR Z$<"0" OR Z$="
" THEN GOTO 172
  178 LET N=0
  180 PRINT ,,"SOMOS:";Z$
  182 FOR G=1 TO 20
  183 NEXT G
  185 FOR E=1 TO 11
  190 FOR B=1 TO VAL Z$
  195 CLS
  201 PRINT L$(1);L$(2);L$(3);L$(
4)
  220 PRINT AT 20,0;"JUGADOR ";B;
" TIRE LOS DADOS "
  225 LET N=0
  240 IF INKEY$="" THEN GOTO 240
  241 PRINT AT 20,0;B$
  250 FOR G=1 TO 5
  252 LET K(G)=INT (RND*6+1)
  253 NEXT G
  255 FOR G=1 TO 5
  260 GOSUB G*10+1100
  262 PRINT AT I,J-1;"";TAB J-1;
"";TAB J-1;"";TAB J-1;G;TAB J-
1;"";TAB J-1;"";TAB J-1;""
  268 GOSUB K(G)*10+1000
  270 NEXT G
  273 LET N=N+1
  275 IF N>=3 THEN GOTO 2400
  278 PRINT AT 16,22;"          "
  279 PRINT AT 15,25;"  "
  280 PRINT AT 21,0;"CUANTOS DADO
S LEVANTA ?"
  285 LET X$=INKEY$
  286 IF X$="P" THEN GOTO 400
  287 IF X$>"5" OR X$<"0" OR X$="
" THEN GOTO 285
  288 IF X$="0" THEN GOTO 2500
  290 PRINT AT 21,0;"QUE DADOS LE
VANTA ?          "
  291 PRINT AT 15,20;"TIRA ";X$;A
T 16,22;
  292 FOR F=1 TO VAL X$
  295 LET C$(F)=INKEY$
  297 IF C$(F)>"5" OR C$(F)<"1" O
R C$(F)="" THEN GOTO 295
  300 PRINT C$(F);"-";
  302 LET G=VAL C$(F)
  305 LET K(G)=INT (RND*6)+1
  310 NEXT F
  315 PRINT AT 21,0;B$
  320 GOTO 255
  405 FAST
  410 GOSUB 5000
  420 PRINT AT 21,0;"CUALQUIER TE
CLA PARA CONTINUAR "
  430 IF INKEY$="" THEN GOTO 430
  435 CLS
  440 PRINT L$(1);L$(2);L$(3);L$(
4)
  445 LET N=N-1
  450 GOTO 255
  999 STOP
 1010 PRINT AT I,J;"
 1011 PRINT AT I+1,J;"
 1012 PRINT AT I+2,J;"
 1013 PRINT AT I+3,J;"
 1014 PRINT AT I+4,J;"
 1015 PRINT AT I+5,J;"
 1016 PRINT AT I+6,J;"
 1017 RETURN
 1020 PRINT AT I,J;"
 1021 PRINT AT I+1,J;"
 1022 PRINT AT I+2,J;"
 1023 PRINT AT I+3,J;"
 1024 PRINT AT I+4,J;"
 1025 PRINT AT I+5,J;"
 1026 PRINT AT I+6,J;"
 1027 RETURN
 1030 PRINT AT I,J;"
 1031 PRINT AT I+1,J;"
 1032 PRINT AT I+2,J;"
 1033 PRINT AT I+3,J;"
 1034 PRINT AT I+4,J;"
 1035 PRINT AT I+5,J;"
 1036 PRINT AT I+6,J;"
 1037 RETURN
 1040 PRINT AT I,J;"
 1041 PRINT AT I+1,J;"
 1042 PRINT AT I+2,J;"
 1043 PRINT AT I+3,J;"
 1044 PRINT AT I+4,J;"
 1045 PRINT AT I+5,J;"
 1046 PRINT AT I+6,J;"
 1047 RETURN
 1050 PRINT AT I,J;"
 1051 PRINT AT I+1,J;"
 1052 PRINT AT I+2,J;"
 1053 PRINT AT I+3,J;"
 1054 PRINT AT I+4,J;"
 1055 PRINT AT I+5,J;"
 1056 PRINT AT I+6,J;"
 1057 RETURN
 1060 PRINT AT I,J;"
 1061 PRINT AT I+1,J;"
 1062 PRINT AT I+2,J;"
 1063 PRINT AT I+3,J;"
 1064 PRINT AT I+4,J;"
 1065 PRINT AT I+5,J;"
 1066 PRINT AT I+6,J;"
 1067 RETURN
 1110 LET I=5
 1111 LET J=1
 1113 RETURN
 1120 LET I=5
 1121 LET J=14
 1123 RETURN
 1130 LET I=5
 1131 LET J=26
 1133 RETURN
 1140 LET I=13
 1141 LET J=1
 1143 RETURN
 1150 LET I=13
 1151 LET J=14
 1153 RETURN
 2000 REM  GENERALA
 2010 FOR G=1 TO 4
 2020 IF K(G)=K(G+1) THEN GOTO 20
30
 2025 GOTO 5400
 2030 NEXT G
 2040 IF N=1 THEN GOTO 5900
 2045 IF Z(10,B)=-1 THEN GOTO 550
0
 2050 IF Z(10,B)<>0 THEN GOTO 205
5
 2051 LET H=50
 2055 LET P$="10"
 2060 GOTO 3260
 2065 IF Z(10,B)<>50 THEN GOTO 59
00
 2066 IF Z(11,B)=-1 THEN GOTO 560
0
 2067 IF Z(11,B)>0 THEN GOTO 5500
 2070 LET H=100
 2075 LET P$="11"
 2080 GOTO 3260
 2100 REM  POKER
 2110 IF K(1)=K(2) AND K(3)=K(2)
AND K(4)=K(3) AND K(1)<>K(5) THE
N GOTO 2185
 2125 IF K(2)=K(3) AND K(4)=K(3)
AND K(5)=K(4) AND K(2)<>K(1) THE
N GOTO 2185
 2130 IF K(3)=K(4) AND K(5)=K(4)
AND K(1)=K(5) AND K(3)<>K(2) THE
N GOTO 2185
 2135 IF K(4)=K(5) AND K(1)=K(5)
AND K(2)=K(1) AND K(4)<>K(3) THE
N GOTO 2185
 2140 IF K(5)=K(1) AND K(2)=K(1)
AND K(3)=K(2) AND K(5)<>K(4) THE
N GOTO 2185
 2150 GOTO 5400
 2155 IF Z(9,B)=-1 THEN GOTO 5500
 2156 IF Z(9,B)>0 THEN GOTO 5500
 2157 IF N=1 THEN LET H=45
 2190 IF N<>1 THEN LET H=40
 2192 LET P$="9"
 2195 GOTO 3260
 2200 REM  FULL
 2210 IF K(1)=K(2) AND K(3)=K(2
AND K(3)<>K(4) AND K(4)=K(5) THE
N GOTO 2285
 2215 IF K(2)=K(3) AND K(4)=K(3)
AND K(4)<>K(5) AND K(5)=K(1) THE
N GOTO 2285
 2220 IF K(3)=K(4) AND K(5)=K(4)
AND K(5)<>K(1) AND K(1)=K(2) THE
N GOTO 2285
 2225 IF K(4)=K(5) AND K(1)=K(5)
AND K(1)<>K(2) AND K(2)=K(3) THE
N GOTO 2285
 2230 IF K(5)=K(1) AND K(2)=K(1)
AND K(2)<>K(3) AND K(3)=K(4) THE
N GOTO 2285
 2250 GOTO 5400
 2285 IF Z(8,B)=-1 THEN GOTO 5600
 2286 IF Z(8,B)>0 THEN GOTO 5500
 2287 IF N=1 THEN LET H=35
 2290 IF N<>1 THEN LET H=30
 2292 LET P$="8"
 2295 GOTO 3260
 2300 REM  ESCALERA
 2302 FOR G=1 TO 4
 2310 IF K(G)=K(G+1)-1 THEN GOTO
2330
 2320 GOTO 5400
 2330 NEXT G
 2332 IF Z(7,B)=-1 THEN GOTO 5600
 2335 IF Z(7,B)>0 THEN GOTO 5500
 2340 IF N=1 THEN LET H=25
 2345 IF N<>1 THEN LET H=20
 2347 LET P$="7"
 2350 GOTO 3260
 2400 PRINT AT 15,20;"        ";AT
 16,20;"
 2405 PAUSE 150
 2410 POKE 16437,255
 2450 CLS
 2460 PRINT AT 10,0;"BIEN JUGADOR
 ";B;" YA TIRO LOS      3 TIROS
PERMTIDOS".
 2470 PRINT ,,,,"AHORA VAMOS A AN
OTAR SU PUNTAJE"
 2480 PAUSE 400
 2490 POKE 16437,255
 2500 FAST
 2510 FOR G=1 TO 4
 2520 FOR F=G TO 5
 2530 IF K(F)>K(G) THEN GOTO 2600
 2540 LET Q=K(G)
 2550 LET K(G)=K(F)
 2560 LET K(F)=Q
 2600 NEXT F
 2610 NEXT G
 3000 FAST
 3002 GOSUB 5000
 3003 PRINT AT 20,0;B$
 3004 PRINT AT 5,16;"ANOTAS:
"
 3005 PRINT AT 21,0;"ANOTAS (1-2-
3-4-5-6-E-F-P-G-D-T)"
 3010 LET P$=INKEY$
 3015 IF P$="" THEN GOTO 3010
 3017 GOSUB 5805
 3020 PRINT AT 21,0;B$
 3030 IF P$="G" THEN GOTO 2000
 3035 IF P$="E" THEN GOTO 2300
 3040 IF P$="P" THEN GOTO 2100
 3050 IF P$="F" THEN GOTO 2200
 3055 IF P$="D" THEN GOTO 2000
 3057 IF P$="T" THEN GOTO 5700
 3060 IF P$>"1" AND P$<="6" THEN
 GOTO 3100
 3070 GOTO 3003
 3100 PRINT AT 21,0;B$
 3105 PRINT AT 5,23;P$
 3110 DIM M(5)
 3115 LET M=0
 3120 FOR G=1 TO 5
 3130 IF K(G)=VAL P$ THEN LET M(G
)=VAL P$
 3140 NEXT G
 3150 FOR G=1 TO 5
 3160 LET M=M+M(G)
 3165 NEXT G
 3175 IF M=0 THEN GOTO 5400
 3200 IF Z(VAL P$,B)<>0 THEN GOTO
 5500
 3205 PRINT AT 20,0;"PARA EL NO.
";P$;" TIENES:";M
 3210 PAUSE 150
 3215 POKE 16437,255
 3250 PRINT AT 21,0;"LO ANOTAS ?
(S/N)          "
 3265 IF INKEY$="S" THEN GOTO 330
0
 3267 IF INKEY$="N" THEN LET Z(VA
L P$,B)=0
 3270 IF INKEY$="N" THEN GOTO 300
3
 3271 GOTO 3265
 3300 IF B=1 THEN LET Q=12
 3306 IF B=2 THEN LET Q=16
 3307 IF B=3 THEN LET Q=20
 3308 IF B=4 THEN LET Q=24
 3309 IF B=5 THEN LET Q=28
 3310 LET L=VAL P$+6
 3312 LET Z(VAL P$,B)=M
 3315 PRINT AT 20,0,B$;AT 21,0;B$
 3320 PRINT AT L,Q;Z(VAL P$,B)
 3350 PRINT AT 21,0;"CUALQUIER TE
CLA P/CONTINUAR"
 3370 IF INKEY$="" THEN GOTO 3370
 3380 NEXT B
 3390 NEXT E
 3400 FAST
 3420 GOSUB 5000
 3430 FAST
 3600 FOR F=1 TO 5
 3610 FOR G=1 TO 11
 3615 IF Z(G,F)=-1 THEN LET Z(G,F
)=0
 3620 LET Z(12,F)=Z(12,F)+Z(G,F)
 3630 NEXT G
 3640 NEXT F
 3650 IF T<>0 THEN GOTO 4020
 3655 LET T=0
 4000 PRINT AT 19,0;"TOTAL-----"
 4020 PRINT AT 19,12;Z(12,1);AT 1
```

```
9,16;Z(12,2);AT 19,20;Z(12,3);AT
 19,24;Z(12,4);AT 19,28;Z(12,5)
4050 PRINT AT 21,0;"CUALQUIER TE
CLA"
4052 SLOW
4055 IF INKEY$="" THEN GOTO 4055
4100 LET MX=Z(12,1)
4110 FOR G=2 TO 5
4120 IF Z(12,G)>MX THEN LET MX=Z
(12,G)
4130 NEXT G
4132 FOR G=1 TO 5
4135 IF MX=Z(12,G) THEN LET GN=G
4140 IF MX=Z(12,G) THEN GOTO 416
0
4150 NEXT G
4160 CLS
4170 RAND USR 16514
4175 PRINT L$(1);L$(2);L$(3);L$(
4)
4180 FOR G=0 TO 31
4190 PRINT AT 6,G;"*";AT 16,G;"*
"
4195 NEXT G
4200 PRINT AT 8,2;"EL GANADOR ES
 EL JUGADOR N. ";GN
4210 PRINT AT 10,2;"OBTUVO ";MX;
" PUNTOS"
4220 PRINT AT 12,0;"MIS FELICITA
CIONES, TIENES UN   GRAN FUTURO
"
4230 PRINT AT 17,12;"J-1";TAB 16
;"J-2";TAB 20;"J-3";TAB 24;"J-4"
;TAB 28;"J-5"
4235 PRINT AT 18,0;"TOTAL-----";
TAB 12;Z(12,1);TAB 16;Z(12,2);TA
B 20,Z(12,3);TAB 24;Z(12,4);TAB
28;Z(12,5)
4250 PRINT AT 20,0;"'R' PARA OTR
O JUEGO"
4265 IF INKEY$="R" THEN GOTO 81
4270 GOTO 4265
4999 STOP
5000 CLS
5005 LET A$="."
5010 PRINT L$(1);L$(2);L$(3);L$(
4)
5020 PRINT AT 5,0
5025 PRINT TAB 12;"J-1";TAB 16;"
J-2";TAB 20;"J-3";TAB 24;"J-4";T
AB 28;"J-5"
5026 IF B=1 THEN PRINT AT 6,12;"
J-1"
5027 IF B=2 THEN PRINT AT 6,16;"
J-2"
5028 IF B=3 THEN PRINT AT 6,20;"
J-3"
5029 IF B=4 THEN PRINT AT 6,24;"
J-4"
5030 IF B=5 THEN PRINT AT 6,28;"
J-5"
5035 FOR G=6 TO 19
5038 PRINT TAB 11;A$;TAB 15;A$;T
AB 19;A$;TAB 23;A$;TAB 27;A$;TAB
 31;A$
5040 NEXT G
5050 FOR G=0 TO 31
5060 PRINT AT 18,G;A$
5065 NEXT G
5070 PRINT AT 6,0
5080 PRINT "1----------";TAB 0;"2
----------";TAB 0;"3----------";TA
B 0;"4----------";TAB 0;"5-------
--";TAB 0;"6----------"
5090 PRINT "ESCALERA--";TAB 0;"F
OUL------";TAB 0;"POKER-----";TA
B 0;"GENERALA--";TAB 0;"5.GENERA
LA"
5100 PRINT AT 5,0;"DADOS ";K(1);
"-";K(2);"-";K(3);"-";K(4);"-";K
(5)
5200 FOR G=1 TO 11
5210 LET O=6
5220 FOR F=1 TO 5
5225 IF Z(G,F)=-1 THEN GOSUB 530
0
5230 IF Z(G,F)>0 THEN PRINT AT G
+6,O+4;Z(G,F)
5240 LET O=O+4
5250 NEXT F
5260 NEXT G
5270 SLOW
5280 RETURN
5300 PRINT AT G+6,O+4;"XX"
5310 RETURN
5400 PRINT AT 20,0;" HEY, NO SEA
S TRAMPOSO,          TUS DADOS NO
 INDICAN ESO          "
5410 PAUSE 300
5420 POKE 16437,255
5430 PRINT AT 20,0;B$;B$
5440 GOTO 3003
5500 PRINT AT 20,0;"HEY, NO PUED
ES ANOTARTE ESO,          YA LO TIENES
          "
5510 PAUSE 300
5520 POKE 16437,255
5530 PRINT AT 20,0;B$;B$
5590 GOTO 3003
5600 PRINT AT 20,0;" HEY, NO SEA
S TRAMPOSO,          ESO YA TACH
ADO          "
5610 PAUSE 300
5620 POKE 16437,255
5630 PRINT AT 20,0;B$;B$
5640 GOTO 3003
5650 PRINT AT 20,0;" HEY NO SEAS
 TRAMPOSO, NO PUEDES TACHARLO YA
 ESTA ANOTADO          "
5660 PAUSE 300
5670 POKE 16437,255
5680 PRINT AT 20,0;B$;B$
5690 GOTO 3003
5700 IF B=1 THEN LET O=12
5705 IF B=2 THEN LET O=16
5710 IF B=3 THEN LET O=20
5715 IF B=4 THEN LET O=24
5720 IF B=5 THEN LET O=28
5730 PRINT AT 5,16;"TACHAS:
"
5750 PRINT AT 21,0;"TACHAS (1-2-
3-4-5-6-E-F-P-G-D-T)"
5760 LET P$=INKEY$
5765 IF P$="" THEN GOTO 5760
5766 GOSUB 5805
5767 IF P$="T" THEN GOTO 3003
5770 IF P$="E" THEN LET P$="7"
5772 IF P$="F" THEN LET P$="8"
5774 IF P$="P" THEN LET P$="9"
5775 IF P$="G" THEN LET P$="10"
5777 IF P$="D" THEN LET P$="11"
5778 IF Z(VAL P$,B)=-1 THEN GOTO
 5600
5779 IF Z(VAL P$,B)>0 THEN GOTO
5650
5781 PRINT AT 21,0;B$;AT 21,0;"L
O TACHAS ? (S/N)"
5783 IF INKEY$="N" THEN GOTO 573
0
5785 IF INKEY$<>"S" THEN GOTO 57
83
5787 LET Z(VAL P$,B)=-1
5788 LET L=VAL P$+6
5790 PRINT AT L,O;"XX"
5795 PRINT AT 21,0;B$
5800 GOTO 3350
5810 IF P$="E" THEN PRINT AT 5,2
3;"ESCALERA"
5815 IF P$="F" THEN PRINT AT 5,2
3;"FOUL"
5820 IF P$="P" THEN PRINT AT 5,2
3;"POKER"
5825 IF P$="G" THEN PRINT AT 5,2
3;"GENERALA"
5830 IF P$="D" THEN PRINT AT 5,2
3;"DOBLE GEN."
5840 IF P$>"0" AND P$<"7" THEN P
RINT AT 5,23;P$
5850 RETURN
5900 RAND USR 16514
5905 PAUSE 170
5906 POKE 16437,255
5910 CLS
5915 IF Z(10,B)=50 THEN LET Z(10
,B)=100
5920 IF Z(10,B)=0 THEN LET Z(10,
B)=50
5925 RAND USR 16514
5926 PRINT L$(1);L$(2);L$(3);L$(
4)
5927 FOR G=0 TO 31
5928 PRINT AT 6,G;"*";AT 16,G;"*
"
5929 NEXT G
5930 PRINT AT 7,3;"BRAVO JUGADOR
 ";B;" CON ESE TIRO";AT 9,3;"ACA
BAS DE GANAR EL PARTIDO";AT 11,2
;"OBTUVISTE UNA GENERALA SERVIDA
"
5940 PRINT AT 17,20;"■"
5954 PRINT TAB 12;"J-1";TAB 16;"
J-2";TAB 20;"J-3";TAB 24;"J-4";T
AB 28;"J-5"
5960 PRINT AT 19,0;"TOTAL-----"
5965 LET T=1
5980 GOTO 3600
5999 STOP
6000 FOR G=0 TO 21
6003 PRINT AT G,0;B$
6005 NEXT G
6006 FOR F=1 TO 4
6007 FOR G=1 TO 15
6010 PRINT AT F-1,G;L$(F,G);AT F
-1,32-G;L$(F,32-G)
6015 NEXT G
6016 NEXT F
6020 PRINT ,,TAB 8;"INSTRUCCIONE
S"
6025 PRINT ,,TAB 4;"TECLA:"
6030 LET P$=" <CUALQUIERA> PARA
TIRAR          <P>          PARA
VER PLANILLA   <0>          SI NO
LEVANTA DADOS <T>          PARA T
ACHAR "
6050 LET D=LEN P$
6090 FOR G=1 TO D
6093 PRINT P$(G TO G);
6095 NEXT G
6100 PRINT AT 21,0;"CUALQUIER TE
CLA PARA COMENZAR"
6120 IF INKEY$="" THEN GOTO 6120
6130 FOR G=0 TO 21
6135 PRINT AT G,0;L$(5)
6140 PRINT AT G,0;B$
6150 NEXT G
6152 GOTO 145
6350 PRINT AT G,0;L$(5)
8000 FOR F=1 TO 11
8010 PRINT Z(F,1)
8020 LET Z(12,1)=Z(12,1)+Z(F,1)
8030 NEXT F
8040 PRINT Z(12,1)
8999 STOP
9000 SAVE "GENERALA"
9010 RUN
9500 FOR N=16514 TO 16543
9510 PRINT N;" = ";
9520 INPUT B
9530 PRINT B;
9540 POKE N,B
9550 NEXT N
```

Softinotas

a) Debemos aclarar que la línea 1 (REM. . .) contenía originariamente (según nos lo mandó ***Jorge****) un programita en lenguaje de máquina para realizar la "inversión" de video en la computadora, interesante efecto para realzar la presentación en pantalla, de este juego. Pero resulta que en el listado que nos acompañó* ***Jorge****, emitido por impresora, estaban, como es lógico, impresos los caracteres respectivos y no los códigos identificatorios de los* "bytes" del L.M.

Por ello, ***Softi*** *debió recurrir al casete que recibiéramos y decodificar los bytes de dicho programita para que todos nuestros lectres puedan así incorporarlo sin mayores problemas a su programa. A tal fin, hemos agregado las líneas* 9500 *a* 9550 *al programa original, y les recomendamos seguir las siguientes instrucciones:*

1) Tipear el programa tal cual fue listado anteriormente, verificando que no existan errores de "dedos"

2) Hacer GOTO 9500

3) A medida que el programa nos vaya dando la "dirección" de máquina, ingresar el código del byte respectivo, según la siguiente lista:

CARGADOR BEM LINEA 1-INVERSOR-

16514	= 58	16515	= 12
16516	= 64	16517	= 111
16518	= 58	16519	= 13
16520	= 64	16521	= 103
16522	= 35	16523	= 1
16524	= 23	16525	= 3
16526	= 126	16527	= 254
16528	= 118	16529	= 40
16530	= 3	16531	= 198
16532	= 128	16533	= 119
16534	= 35	16535	= 11
16536	= 121	16537	= 176
16538	= 32	16539	= 242
16540	= 201	16541	= 36
16542	= 37	16543	= 28

*4) Correr el programa normalmente (*RUN*)*

*5) Una vez verificado totalmente su funcionamiento y depurado de eventuales errores en el ingreso de las instrucciones, grabar el programa (*SAVE*'ar).*

b) La adaptación para sus "hermanas mayores" (CZ Spectrum TK90 y TS2068) es muy simple, eliminando el SCROLL y los FAST/SLOW de sus instrucciones.

c) Te felicito muy especialmente, Jorge, por tu programa que seguramente hará las delicias de nuestros amigos, principalmente de los muchos aficionados a este "deporte" de los dados.

Chau (va cheque por correo). . . ***Softi***



LAS PIEDRAS DE LA DISCORDIA

Softi **te propone un juego de ingenio**
Sacá vos, después saca Softi,
después vos, después Softi,
después . . .
Y a ver quién gana??. . .

Este es un juego de "inteligencia", en el cual la computadora (**Softi**) escribirá en la pantalla 41 "XXXX. . .XXX" cada una de las cuales representa ser una "piedrita".

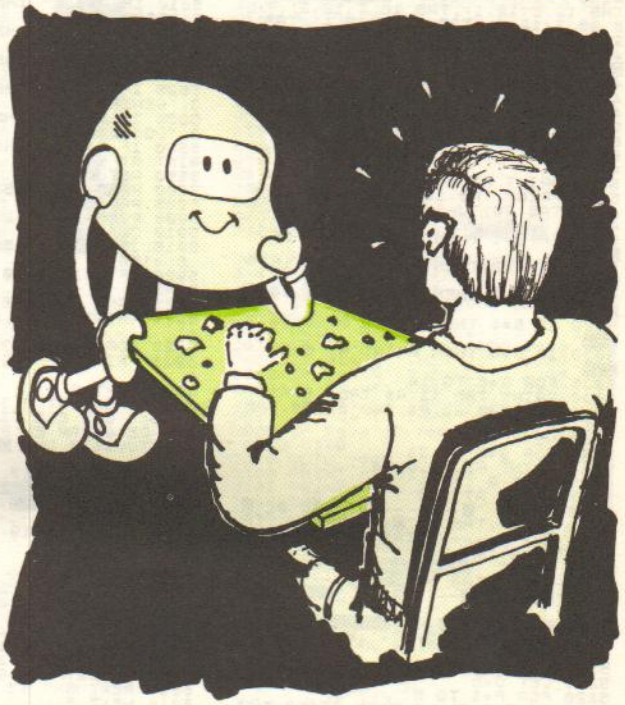
```
HAY 41 PIEDRITAS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX

CUANTAS SACAS?

QUEDAN 38 PIEDRITAS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX

SOFTI SACA 1 PIEDRITAS
QUEDAN 37 PIEDRITAS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXX

CUANTAS SACAS?
```



Nosotros tendremos que ir tomando piedritas alternativamente con la computadora, tratando de dejarle al maldito **Softi** la última piedrita, porque:

El que se queda con la última piedrita, PIERDE!!!

```
QUEDAN 8 PIEDRITAS
XXXXXXXX

SOFTI SACA 3 PIEDRITAS
QUEDAN 5 PIEDRITAS
XXXXX

CUANTAS SACAS?

QUEDAN 2 PIEDRITAS
XX

SOFTI SACA 1 PIEDRITAS
QUEDAN 1 PIEDRITAS
X

TE QUEDASTE CON LA ULTIMA.
GANO SOFTI ! ! !
```

El juego es muy simple, pero tiene una pequeña restricción *cada jugador sólo puede tomar 1, 2 o 3 piedritas por vez.*

Así que vamos al grano. . . o mejor dicho a las piedras, y veamos el programa:

Tabla de variables

F$	"QUEDAN"
D$	"PIEDRITAS"
X	Cantidad de piedritas (41 iniciales)
C	Piedritas que saca el jugador
I	Piedritas que saca **Softi**
D	Tiempo de demora
P	Variable FOR/NEXT para la demora
T	Variable FOR/NEXT para imprimir las "X"

TK83/85/90X/TI99/TS2068/ CZ1000 - 1500 - Spectrum

```
100 REM SOFTI 1985
110 REM
120 REM LAS PIEDRITAS
130 REM DE LA DISCORDIA
140 REM
150 REM ESTE PROGRAMA
160 REM SE PUEDE CORRER
170 REM EN CUALQUIER
180 REM COMPUTADORA
190 REM (OJO LINEAS 300,310,420
Y 560)
200 LET D=200
210 LET F$="QUEDAN "
220 LET D$=" PIEDRITAS"
230 LET X=41
240 PRINT "HAY ";X;D$
250 PRINT
260 GO SUB 460
270 PRINT "CUANTAS SACAS?"
280 INPUT C
290 LET C=INT C
300 IF C>3 THEN GO TO 440
310 IF C<1 THEN GO TO 440
320 LET X=X-C
330 PRINT
340 PRINT F$;X;D$
350 GO SUB 460
360 LET I=4-C
370 PRINT "SOFTI SACA ";I;D$
380 LET X=X-I
400 PRINT F$;X;D$
410 GO SUB 460
420 IF X=1 THEN GO TO 550
430 GO TO 270
440 PRINT "NO HAGAS TRAMPA. SAC
A 1, 2 O 3"
450 GO TO 270
460 FOR P=1 TO D
470 NEXT P
480 FOR T=1 TO X
490 PRINT "X";
500 NEXT T
510 PRINT
520 PRINT
530 PRINT
540 RETURN
550 PRINT "TE QUEDASTE CON LA U
LTIMA.      GANO SOFTI ! ! !"
560 STOP
```

Estructura del Programa

150 / 250 Inicialización de variables preparación de la pantalla

260 Ejecuta la rutina 460, imprimiendo en la pantalla las piedritas que quedan

270 / 280 **Softi** nos pregunta cuántas piedritas sacamos nosotros.

290 Esta instrucción está para los tramposos. Así, "C" toma el valor de la parte entera que le damos.
Ejemplo: si deseamos sacar 1,5 piedritas nuestra computadora interpretará que sacamos solamente 1 (una)

300 / 310 Verifica que C esté entre 1 y 3. Caso contrario, imprime un mensaje

320 / 350 Determina e imprime la cantidad de piedras restantes

360 **Softi** decide aquí qué cantidad de piedras retira la máquina, dependiendo de las que hayamos sacado nosotros.
Así: si sacamos 1, saca 3
si sacamos 2, saca 2
si sacamos 3, saca 1
La expresión matemática que responde a esto sería:

$$I = 4 - C$$

Siendo: I = Piedritas **Softi**
4 = Piedritas a sacar por cada vuelta
C = Piedritas que sacamos nosotros.

370 / 410 Imprime I, Recalcula e imprime las piedritas restantes

420 / 430 Determina si finaliza el juego
- si X= 1 queda una sola piedra y el juego termina (GOTO 550)
- si nó, continúa (GOTO 270)

440 / 450 Imprime el merecido cartel si osamos hacerle trampa al "honesto" de **Softi**

460 / 540 Rutina de demora e impresión de piedritas restantes en pantalla

550 / 560 Fin del juego siendo, por supuesto, **Softi** el ganador

Para la TI99/4A:
- Líneas 300, 310 y 420: eliminar el GOTO
- Línea 560: reemplazar por END



(viene de la pág. 20)

Tabla de Variables

A$	Vector que contiene la cantidad de días de cada mes (los 2 primeros 31 - de enero– los 2 siguientes 28 (de febrero) y así hasta los 2 últimos con los 31 días de diciembre)
E	Permite determinar si la fecha es válida o no (igual a cero: válida)
F$	Contiene la fecha DDMMAA
I	Utilizada para el FOR/NEXT
DD	Contiene el día de la fecha
MM	Contiene el mes de la fecha
AA	Contiene el año de la fecha

Softinota:

Les informo que en la nota anterior sobre "Manejo de fechas (II)" hay un error, que se corrige así:

- *reemplazar la línea 5030 por:*

```
5030 IF Y(C)>=S THEN GOTO 5045
```

- *agregar las líneas:*

```
5045 IF C>1 THEN GOTO 5050
5047 LET DG=S
5049 GOTO 5060
```

Gracias y sorry...

Softi



ATAQUE



Este programa que desarrolló **Gustavo** intenta reproducir un ataque a una nave espacial.

Veamos entonces el programa y la explicación que nos dá el amigo **Gustavo.**

CZSpectrum - TK90 - TS 2068

```
  1 REM **********************
  2 REM *      A T A Q U E     *
  3 REM *                      *
  4 REM *HECHO EL 14/2/85 POR*
  5 REM *                      *
  6 REM *    GUSTAVO RIGO      *
  7 REM **********************
 10 REM Formacion de grafico
 definido
 11 FOR g=0 TO 7: READ sof: POK
E USR "a"+g,sof: NEXT g
 12 DATA BIN 01000010,BIN 01000
010,BIN 01000010,BIN 01111110,BI
N 00011000,BIN 00011000,BIN 0001
1000,BIN 00011000
 14 REM Preparar pantalla
 15 BORDER 4: PAPER 7: INK 2: C
LS : FOR l=6 TO 26: PRINT ; FLAS
H 1;AT 3,l;"*";AT 9,l;"*": NEXT
 l: FOR l=4 TO 9: PRINT ; FLASH 1
;AT l,6;"*";AT l,26;"*": NEXT l
 17 PRINT INK 2;AT 5,9;" A T T
A Q U E";AT 7,8;"E S P A C I A L
";AT 11,8;"Por: Gustavo Rigo";AT
 14,8;"MAXIMO 35 PUNTOS"
 18 PRINT AT 17,2; INK 2;"CUALQ
UIER TECLA PARA EMPEZAR": PRINT
AT 21,0;"TECLA 0=DERECHA  TECLA
1=IZQUIER"
 19 PAUSE 0
 20 REM Subrutina
 25 GO SUB 200
 30 FOR i=1 TO 4: PRINT AT b(i)
,c(i);" ";: NEXT i
 40 LET v=v+(INKEY$="0" AND v<2
3)-(INKEY$="1" AND v>3)
 50 PRINT AT 20,v; INK 1;v$;
 60 FOR i=1 TO 4
 70 IF b(i)<20 THEN GO TO 120
 80 LET b(i)=4: LET c(i)=v-3+IN
T (RND*11)
 90 LET p=p+1: PRINT AT 0,8;p;
 95 IF p=35 THEN GO TO 500
 110 GO TO 130
 120 LET b(i)=b(i)+s
 130 BEEP .01,15: PRINT AT b(i);
c(i); INK i;s$;
 140 IF b(i)<20 THEN GO TO 160
 150 IF (c(i)-v<5 AND c(i)-v>0)
THEN GO TO 300
 160 NEXT i
 170 GO TO 30
 199 REM Comienzo
 200 BORDER 2: PAPER 5: CLS
 210 LET v$=" "+CHR$ 134+CHR$ 14
0+CHR$ 140+CHR$ 137+" "
 220 LET s$="Y"
 230 LET s=1: LET p=0: LET v=13:
 240 DIM b(4): DIM c(4)
 250 FOR i=1 TO 4: READ b(i): RE
AD c(i): NEXT i
 260 PRINT AT 0,0;"Puntos:0";
 270 RETURN
 280 DATA 16,4,12,8,4,16,8,2
 300 REM Efectos especiales
 500 FOR o=1 TO 50: BEEP .05,o:
NEXT o: FOR g=1 TO 5: FOR k=1 TO
 7: BEEP .01,k+15: BORDER k: PAP
ER k: CLS : NEXT k: NEXT g
 525 PAPER 5: CLS
 540 BEEP .1,9
 600 PRINT ; INK 7; PAPER 2; FLA
SH 1;AT 3,0;"    F E L I C I T A
C I O N E S  ": PRINT ; PAPER 2;
 FLASH 1;AT 4,0;"
                ": FLASH 1;AT 2
,0;"
 "
 620 PRINT ; INK 7; PAPER 3; FLA
SH 1;AT 9,8;"MUY BUEN SCORE:";p
 623 REM Juega de nuevo?
 630 PRINT ; INK 2;AT 11,8;"OTRO
 JUEGO?-(S/N)"
 635 INPUT w$
 640 IF w$="s" OR w$="S" THEN RU
N
 650 IF w$="" THEN GO TO 635
 670 IF w$="n" OR w$="N" THEN LI
ST
 820 STOP
9999 SAVE "Ataque" LINE 1
```

Estructura del Programa

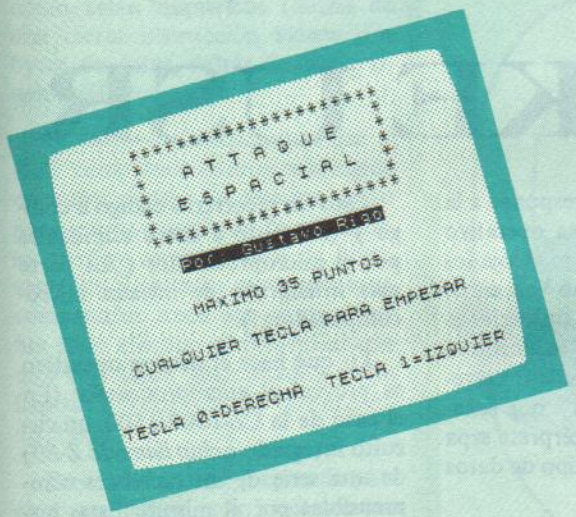
1 / 10	REM
11 / 12	Datos (DATA) para configurar los misiles
15 / 17	Primera presentación de pantalla
20 / 170	Imprime la nave, los misiles y los puntos
199 / 280	Se le da valores a todas las variables (LET, DATA)
300 / 540	Efectos especiales de festejo
600 / 670	Felicita al jugador y pregunta por otro partido También informa resultados finales
820	STOP
9999	SAVE'ado

Explicación

Consiste en manejar una nave esquivando misiles.

Se usan las teclas 1 = izquierda y 0 = derecha.

El máximo de puntos es 35 (cada misil esquivado vale 1 punto)



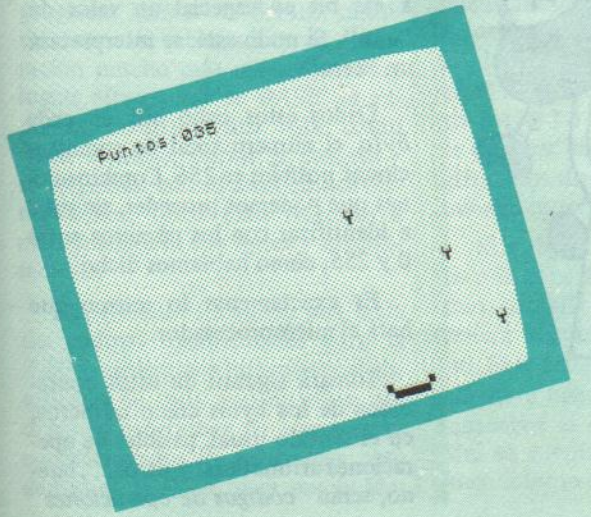
Al llegar al puntaje máximo se le preguntará si desea hacerlo nuevamente: si contesta S (por sí) empezará nuevamente, de lo contrario listará el programa.

Cada jugador tiene una sola vida.

Les recomiendo a los amigos que luego de copiar el programa, digiten RUN. Si todo anda bien, teclear GO-TO 9999 para SAVE' arlo

Softinota:

El carácter gráfico "a" de la línea 11, ingresarlo presionando previamente las teclas CAPS SHIFT y GRAPHICS.



Para ampliar el rango del puntaje, es decir, que el máximo no sea 35 sino un número más grande, sólo tienen que cambiar el 35 de la línea 95 por el pensado.

También si se quiere cambiar el ruido de los misiles, sólo tienen que cambiar los valores de la función BEEP por los que se deseen.

Tabla de Variables

l	Se usa en la presentación (pantalla) de comienzo
v$	Guarda el conjunto de funciones CHR$ que forman la "nave"
s$	Variable (string) que almacena el gráfico definido ("misiles") a esquivar
b (i) y c (i)	Según cómo varíe la variable i se irán imprimiendo los misiles
p	Puntaje del jugador (máximo 35 puntos)
w$	Para ingresar información acerca de si juega de nuevo o no
k	Efectos especiales al llegar a los 35 puntos

Softinotas

1. *El desplazamiento de la nave y los misiles son algo "lentos"*

2. *El programa siempre "felicita" al jugador y pone "MUY BUEN SCORE", aunque el puntaje sea el mínimo posible.*

3. *Hice algunas correcciones en la Estructura del Programa que me mandaste, **Gustavo**.*

4. *De todos modos tu esfuerzo y el programa bien merecen que seas uno de los ganadores del concurso del mes.*

5. *Otra cosa, no me dijiste tu edad.*

Te mando el cheque por correo, chau. . .

Softi

LENGUAJE DE MAQUINAS (I)

PEEK POKE USR

Sobre el teclado de nuestra "Sinclair" conviven tres funciones BASIC de nombres PEEK, POKE y USR.

Una gran parte de la humanidad es indiferente a estas tres palabras. Pero la minoría restante sufre, ante su sola mención, síntomas varios que van desde una leve inquietud hasta un terror pánico.

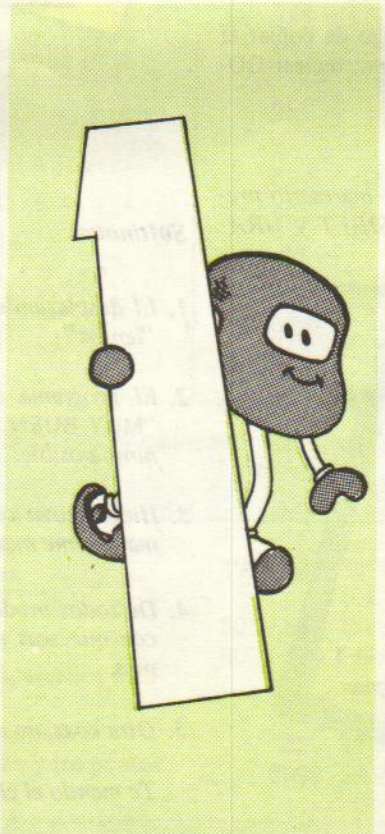
Una reacción similar produce, en este segundo grupo, la mención de claves cabalísticas como *"lenguaje de máquina"* o *"assembler"*.

Para tranquilizar a estas personas explicaremos brevemente qué es el *Lenguaje de Máquina (LM)* y cuál es el papel de estas funciones en el mismo.

Cuando escribimos un programa en BASIC, estamos en realidad dando a consideración del *Intérprete* BASIC un conjunto de códigos que, línea a línea, son examinados para verificar su correcta sintaxis. Esta sintaxis o conjunto de reglas gramaticales es necesaria a consecuencia de que el *Sistema Operativo* está preparado para funcionar con datos que le proveamos, pero presentados de cierto modo. Lo contrario, lo que sería absoluta libertad sintáctica, requeriría un poder de análisis mucho mayor por parte de este sistema operativo, y ésto implicaría una complejidad y extensión mucho mayores de los programas destinados a este fin.

Si la línea que tipeamos pasa este examen previo del Intérprete, se almacena junto con su correspondiente número de línea en la zona de memoria específica reservada a los programas. La forma en que este programa se almacena responde a la manera como el sistema operativo "desea" que el Intérprete recoja los datos a evaluar, que no son otros que los códigos de las operaciones que estamos ordenando se ejecuten. Códigos que se almacenan suavemente "condimentados" por señales varias para que el intérprete sepa en cada momento qué tipo de datos está manejando.

Una vez que nuestro programa ha pasado la primera inspección, está listo para que el sistema operativo reciba la orden RUN y se abalance sobre él a ejecutarlo.



Esto es: recorrerlo código a código y traducirlo a datos asimilables por el microprocesador (¡La parte más inteligente de nuestra microcomputadora!).

Lo que hace el sistema operativo es proveer al *microprocesador* (En el caso de la CZ-1000/1500, un circuito integrado o*chip llamado Z-80*) de una serie de instrucciones comprensibles por el mismo. Estas instrucciones están almacenadas en la ROM bajo la forma de una secuencia de *bytes*, de longitud y composición variables según el sistema operativo de que se trate.

Estos bytes cargan, cada uno, un "estado de memoria" fijo e inalterable, que podríamos interpretar como un número entre 0 y 255.

La razón de esta interpretación para el contenido de cada byte es que cada uno de ellos está compuesto por ocho circuitos –llamados *bits*– afectados cada uno por la presencia o ausencia de una tensión eléctrica. Si el bit en cuestión está cargado, la computadora le asignará a ese bit en especial un valor de "uno". Si no lo está, se interpretará un "cero".

Dados estos ocho bits en cada *byte*, el número total de combinaciones posibles es 256. Combinaciones que podemos proceder, sin más, a identificar con los números entre 0 y 255, como habíamos dicho.

Es exactamente lo mismo que hará el microprocesador.

Tomará algunos de estos contenidos de los bytes como números, en el sentido usual, pasibles de operaciones aritméticas, y otros... bueno, serán *"códigos de operaciones"*

a realizar. Toda la larga tira de ocho mil y algunos bytes que están grabados "de fábrica" en nuestra *ROM* son en realidad un extenso programa en Lenguaje de Máquina, donde las instrucciones y los datos se almacenan del mismo modo; el microprocesador los distinguirá perfectamente, porque tiene forma de reconocer cómo están dispuestos (detrás de una cierta instrucción vienen dos bytes de datos, detrás de otras vienen dos bytes, etc.).

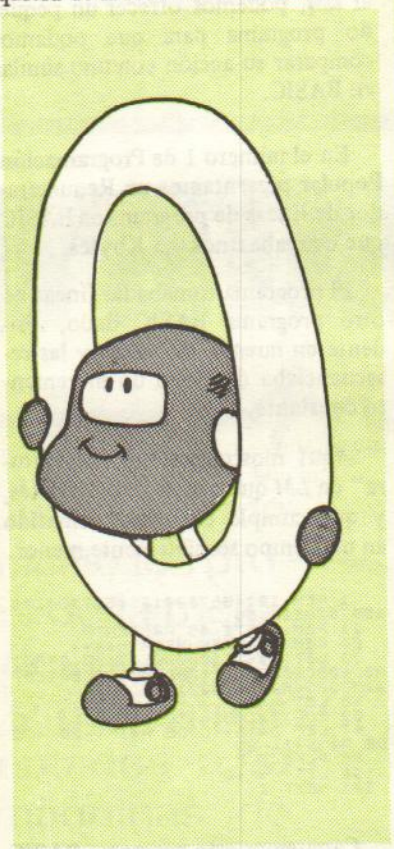
El juego de instrucciones que el *Z-80* comprende es muy restringido en capacidades. Se pueden alojar números en posiciones de memoria, se pueden cambiar éstos de lugar, sumarles y restarles otros números, compararlos unos con otros, y sobre todo efectuar ciertos malabarismos con datos, guardándolos en los llamados *"registros"* de que el *Z-80* dispone.

Pero éste no está preparado para ejecutar, por sí solo, operaciones más complejas como una multiplicación. Hay que enseñárselo mediante un programa.

De donde se infiere que, dado que el manejo que la computadora hace de la información es bastante versátil, no sólo multiplicando sino desarrollando operaciones matemáticas mucho más complejas, los programas sobre los que descansa su funcionamiento deben ser más bien complicados.

Es así, en efecto. Un programa en *LM*, es decir, una secuencia de bytes que contengan códigos de operaciones asimilables por el microprocesador, es, al lado de un programa similar en BASIC, infinitamente más rebuscado y de elaboración mucho más minuciosa. Para lograr efectos que en BASIC se expresan mediante una sola orden —por ejemplo, borrar la pantalla— se requieren en *LM* unas quince instrucciones, todas ellas referidas a manipulaciones con bytes y sus contenidos.

Si tuviéramos a mano una tabla de códigos operacionales para el *Z-80* y supiéramos interpretar el significado de cada uno de ellos, podríamos enterarnos de cómo están confeccionados los programas, residentes en la *ROM*, para que la Sinclair funcione de la forma en que lo hace. Simplemente usando la función PEEK de la manera que se indica en el manual, podemos enterarnos del contenido de un byte cualquiera de la *ROM* o de la *RAM*.



La operación inversa, depositar un cierto contenido en un cierto byte, y que permite al usuario crear sus propias rutinas en *LM*, es POKE Sólo se puede cambiar el contenido de un byte que no resida en la *ROM* (por suerte, ya que si lo hacemos, casi con seguridad volveremos idiota al sistema operativo). De hecho, mientras tecleamos un programa, o mientras estamos trabajando en modo *comando*, estamos continuamente depositando ciertos contenidos en la estructura íntima de la memoria *RAM*. Lo que POKE nos permite es hacer esta carga selectiva, dirigiéndonos directamente a un determinado byte, y sólo a uno por vez.

Si el byte que elegimos pertenece a la zona de memoria de programas, pantalla o variables, esta carga tendrá efectos diversos en cada caso. Si se trata de un byte correspondiente a una "variable de sistema", entonces el sistema operativo tendrá en cuenta este cambio en el manejo interno de la información.

De este modo, byte a byte, o sea instrucción a instrucción, es como podemos cargar una zona determinada de la memoria con un programa en *LM*, a semejanza de los programas que ya están cargados y protegidos en la *ROM*. Una vez cargados los contenidos correspondientes, podemos instruir a la máquina para que, abandonando el imperio del sistema operativo, ceda el control al microprocesador que debe pasar a ejecutar, una por una, esas instrucciones. Lo haremos mediante la orden USR seguida del número correspondiente al byte donde comienza nuestro programa.

A partir de ese momento, y liberado su accionar del control del sistema operativo, que continuamente interrumpe al *Z-80* para tareas varias, el mismo *Z-80* va a desplegar su alucinante velocidad: ejecutará las instrucciones de nuestro programa a razón de cientos de miles por segundo.

Todo lo que hay que hacer es aprender la manera en que decirle al *Z-80* qué y cómo necesitamos que haga. Empresa de mediana magnitud, que no porque sepamos BASIC se verá demasiado facilitada. El BASIC es un lenguaje evolucionado, en el sentido de que la estructura y sentencias de un programa son medianamente comprensibles a simple vista por una persona. Pero el *LM* resiste todo primer análisis, al ser "nada más" que un conjunto de números, tan números como cualquier otro... ¡y para peor, *binarios*!

Para paliar esta dificultad de interpretación y por razones operativas del microprocesador, se acostumbra codificar esos números en *sistema hexadecimal*, que es un sistema de numeración posicional *de base 16*, es decir, que consta de dieciseis dígitos. Aún así resulta muy evidente para el programador de *LM* que éste no es algo diseñado en función de su comodidad.

En BASIC, estamos acostumbrados a indicar a la computadora dónde ir a buscar datos mediante un "nombre" que nosotros sugerimos para esa posición (caso de las variables BASIC, o de las líneas de un programa BASIC). Aún cuando, por razones de manejo interno del espacio, las direcciones absolutas de

esos datos cambien, el sistema operativo los rastreará por nosotros, flexibilizando su uso. En *LM* las cosas son muy distintas. Para usar datos o partes de programas deberemos tener muy en claro las direcciones en que esos datos residen. Tendremos que hacer un montón de cuentas relativas a esas direcciones cada vez que eliminemos o insertemos instrucciones entre otras, y cuidarnos mucho de no impresionar zonas sensibles, reservadas para uso específico del sistema operativo.

Si es tan innatural e inconveniente el uso del *LM*. ¿Por qué no aprovechar el BASIC o cualquier otro de los idiomas superiores, más hechos al trato con la gente?

En primer lugar, porque así fue como empezó la cosa. En torno a un microprocesador, o a su equivalente antepasado, se desarrolló un sistema que devino en nuestra microcomputadora, provista de un lenguaje superior. Y porque éste u otro lenguaje superior, junto con sus ventajas trajo limitaciones. Por ejemplo, la complejidad de la tarea de traducción que realiza el Intérprete BASIC, hace que muchas operaciones equivalgan a mucho tiempo de operación. En *LM* esta etapa nos la ahorramos, con la correspondiente ganancia de tiempo.

En realidad, estaremos pasando la información "por debajo del mostrador" del sistema operativo, que hará "la vista gorda" a cosas que en otro momento serían imperdonables.

En segundo lugar, a pesar de que un programa en *LM* lleva más instrucciones, cada una ocupa sólo de uno a cuatro bytes en el *Z-80*. El ahorro de memoria es flagrante.

Las dificultades que ofrece el estudio y práctica del *LM* se empequeñecen al lado de lo interesante que puede resultar y de lo mucho que podemos aprender acerca de nuestra micro, por no mencionar lo conveniente que es en muchas oportunidades.

Para una práctica más fácil, conviene munirse de un *programa monitor*, que es el que permite desarrollar rutinas en *LM* con más comodidad y que apenas estemos bien en el tema podremos hacer nosotros mismos. Mejor aún si lo que tenemos es un *Ensamblador*, que nos permitirá entrar directamente el *LM* por sus *nemónimos*.

Para que podamos aproximarnos al *LM*, podemos ofrecer un pequeño programa para que podamos comparar su acción con uno similar en BASIC.

En el número 1 de **Programación Popular** presentamos un Renumerador de líneas de programa en BASIC que ocupaba unos 0,5 Kbytes.

El programa tomaba las líneas de otro programa BASIC dado, residente en nuestra máquina, y las resecuenciaba dándoles un incremento constante.

Aquí mostramos una "miniatura" en *LM* que ocupa sólo 36 bytes, y que cumple el mismo cometido en un tiempo sensiblemente menor.

```
   1 REM 12345678901234567890123
4567890123456
  11 REM SOFTI 85 PRESENTA
  24 REM ****RENUMFLASH****
  33 LET A$="11000021A740010A003
A0D40BC20073A0C40BD2001C9E809EB7
22373234E23460923 8E2"
  46 LET K=1
  59 FOR I=16514 TO 16549
  67 LET D=16*(CODE A$(K)-28)+CO
DE A$(K+1)-28
  88 POKE I,D
 199 LET K=K+2
 301 NEXT I
```

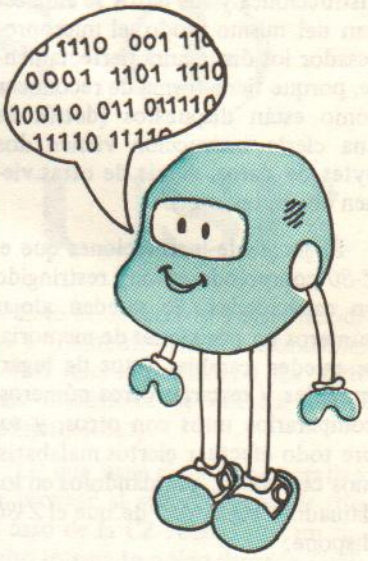
Carguemos este programa BASIC ***sin el más mínimo error*** y hagamos RUN. Si luego listamos veremos la línea 1 REM llena de caracteres que no habíamos puesto allí. La razón es que éste es el verdadero programa en *LM*: se compone de códigos que la computadora, al listar el programa BASIC que lo contiene, interpreta de la manera que vemos. Todo lo restante, de la línea 11 en adelante, es sólo un *programa cargador* para el mismo, que podemos borrar, o mejor aún, hacerlo objeto de nuestro primer experimento. Vamos a resencuenciarlo.

Siempre que queramos usar el programa en *LM* haremos:

```
LIST USR 16514
```

que es la dirección inicial del mismo

Tipeemos pues este comando y veamos desplegarse las líneas del



programa cargador (antes numeradas 11, 24, 33, etc.) ahora prolijamente dispuestas de 10 en 10.

Si queremos renumerar este u otro programa dado otro incremento, digamos de a cincuenta o de a cien, hacemos:

```
POKE 16521,INCREMENTO
```

donde "incremento" sea la diferencia entre línea y línea deseada... siempre que sea menor que 255.
En otro caso haremos:

```
POKE 16521,RESTO
POKE 16522,COCIENTE
```

donde "resto" y "cociente" sean respectivamente los de dividir el incremento por 256.

Debemos tener en cuenta que, al igual que su primo en BASIC ya presentado, este RENUMFLASH no tiene en cuenta las transferencias de control, es decir, IFs, GOTOs, GOSUBs, que enviaran el control del programa a líneas que fueran a ser renumeradas. Habrá que renumerar estas transferencias a mano. Un programa que las tuviera en cuenta no es imposible, pero sería muchísimo más complejo y largo.

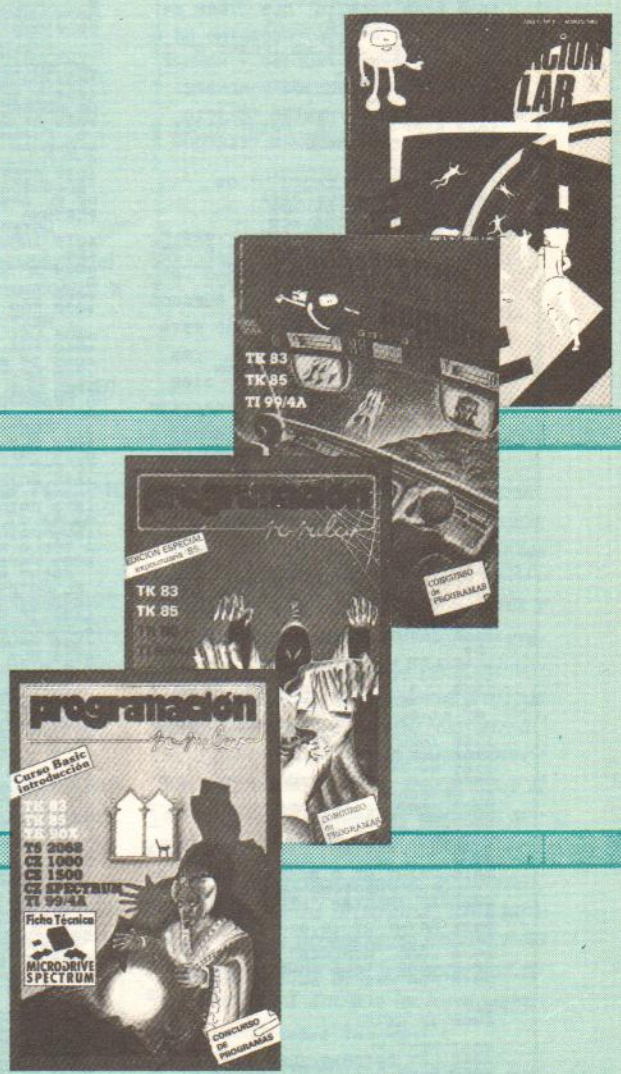


12

MUY BUENAS
RAZONES PARA
SUSCRIBIRTE A:

programación popular

Sólo así recibirás puntualmente cada mes en tu domicilio los próximos 12 ejemplares de la mejor revista argentina de computación, con artículos y programas especialmente orientados a la totalidad de las máquinas de nuestro mercado.



- *PRECIOS ESPECIALÍSIMOS (VER TALON)*
- CIRCULO DE SUSCRIPTORES (Concurso de amigos, sorteos y ofertas especiales).
- Softi Club: afiliación automática.
- Concurso de Programas

Sólo tienes que completar y enviar
HOY MISMO
el Talón adjunto (sin franqueo).

APROVECHA LA OFERTA Y SUSCRIBITE...



INDICE

Este programa que nos enviara **Francisco** (*de 16 años*), es un interesante utilitario cuya aplicación práctica es obvia y generalizada, pues permite cargar, mantener y recuperar un "índice telefónico", dándonos un ejemplo concreto de las posibilidades de nuestra microcomputadora para ayudarnos en nuestros quehaceres domésticos.

CZSpectrum - TK90 - TS 2068

```
   1 REM INDICE DE TELEFONOS
   2 BORDER 3: PAPER 6: CLS
   3 GO SUB 9000
   5 CLS : FOR F=-5 TO 15: BEEP
.03,F: NEXT F: LET E$="": PRINT
 AT 2,10;"* MENU *"; OVER 1;AT 2
,12;""; OVER 0
   6 PRINT AT 6,0;"1. ENTRAR NUE
VOS DATOS.";AT 8,0;"2. ENTRAR DA
TOS POR EL GRABADOR.";AT 10,0;"3
. GRABAR DATOS EN EL GRABADOR.";
   7 PRINT AT 12,0;"4. BUSCAR TE
LEFONO POR NOMBRE.";AT 14,0;"5.
BUSCAR NOMBRE POR TELEFONO.";AT
16,0;"6. LISTAR TODO EL INDICE."
;AT 18,0;"7. ESCAPAR DEL PROGRAM
A."
   8 PRINT AT 20,0;"PRESIONE EL
NUMERO DE OPCION Y  LUEGO ENTER.
"
   9 INPUT "SU OPCION : ";O$: IF
 CODE O$<49 OR CODE O$>55 THEN G
O TO 9
  10 GO SUB VAL O$*1000
  11 GO TO 5
1000 BEEP .025,15: CLS : REM ENT
RAR NUEVOS DATOS
1005 PRINT AT 1,6;"REGISTRO DE D
ATOS": BEEP .025,12: PRINT AT 11
,0;"DIGITE EL APELLIDO Y NOMBRE
A    REGISTRAR."
1010 INPUT "APELLIDO Y NOMBRE:
";B$
1015 IF B$="" THEN GO TO 1010
1020 BEEP .025,15: PRINT AT 11,0
;"DIGITE EL NUMERO DE TELEFONO D
E ";B$;"
"
1025 INPUT "TELEFONO: ";C$
1030 LET DATOS=DATOS+1
1035 LET N$(DATOS)=B$
1040 LET T$(DATOS)=C$
1050 FOR F=1 TO 15: BEEP .025,F:
 NEXT F: CLS : PRINT AT 1,6;"REG
ISTRO DE DATOS";AT 9,0;"APELLIDO
 Y NOMBRE";TAB 22;"TELEFONO";AT
11,0; INVERSE 1;N$(DATOS);" ";T
$(DATOS): PRINT '"      HA QUEDADO
 REGISTRADO"
1060 PRINT '''"EL REGISTRO ESTA
CORRECTO(S/N): "
1065 INPUT "SU RESPUESTA: ";R$
1070 IF R$="S" THEN RETURN
1080 IF R$="N" THEN GO TO 1100
1090 GO TO 1065
1100 CLS : PRINT AT 1,6;"REGISTR
O DE DATOS": BEEP .07,15: PRINT
AT 5,0;"QUE DATO ESTA INCORRECTO
?(1/2): "
1105 PRINT AT 9,9;"1. EL NOMBRE.
";AT 11,9;"2. EL TELEFONO.";AT 1
5,0;" * PRESIONE SU NRO. DE OPCI
ON *"
1110 INPUT "SU RESPUESTA: ";R$
1115 IF R$="1" THEN GO TO 1130
1120 IF R$="2" THEN GO TO 1150
1125 GO TO 1110
1130 CLS : BEEP .025,15: PRINT A
T 1,6;"REGISTRO DE DATOS": PRINT
 AT 11,1;"ENTRE EL APELLIDO Y NO
MBRE  CORRECTO Y LUEGO PRESIO
NE ENTER"
1135 INPUT "APELLIDO Y NOMBRE:
";B$
1140 LET N$(DATOS)=B$
1145 GO TO 1050
1150 CLS : BEEP .025,15: PRINT A
T 1,6;"REGISTRO DE DATOS": PRINT
 AT 11,0;"ENTRE EL TELEFONO CORR
ECTO Y    LUEGO PRESIONE  ENTER"
1155 INPUT "TELEFONO: ";C$
1160 LET T$(DATOS)=C$
1165 GO TO 1050
2000 CLS : REM ENTRAR DATOS POR
EL GRABADOR
2005 BEEP .2,15: PRINT AT 7,0;"R
EBOBINE LA CINTA DE DATOS Y    L
UEGO PRESIONE ENTER"
2010 IF INKEY$="" THEN GO TO 201
0
2015 PRINT AT 9,0;"PRESIONE PLAY
 Y LUEGO PRESIONE  ENTER"
2020 IF INKEY$="" THEN GO TO 202
0
2025 PRINT AT 12,0; FLASH 1;"CAR
GANDO LOS DATOS"
2030 LOAD "NOMBRES" DATA N$()
2040 LOAD "TELEFONOS" DATA T$()
2045 FOR F=1 TO 50: IF N$(F)="
                    " THEN GO TO 2
047
2046 IF N$(F)<>"
     " THEN LET S=S+1
2047 NEXT F
2048 LET DATOS=S: LET S=0
2050 BEEP 2,4: CLS : PRINT AT 11
,3;"**  CARGA  CONCLUIDA  **": P
AUSE 200: RETURN
3000 CLS : REM GRABAR DATOS EN E
L GRABADOR
3005 IF DATOS=0 THEN PRINT AT 11
,0; FLASH 1;" **NO HAY DATOS PAR
A OPERAR **  ": PAUSE 200: RETUR
N
3010 PRINT AT 11,0;"REBOBINE LA
CINTA LUEGO PRESIONE ENTER."
3015 IF INKEY$="" THEN GO TO 301
5
3020 PRINT "PRESIONE GRABACION Y
 LUEGO ENTER"
3030 IF INKEY$="" THEN GO TO 303
0
3035 PAUSE 100
3040 SAVE "NOMBRES" DATA N$()
3045 SAVE "TELEFONOS" DATA T$()
3050 PRINT TAB 6;"GRABACION CONC
LUIDA"
3055 CLS : PRINT AT 11,1;"PREPAR
E LA CINTA PARA VERIFICAR LA GRA
BACION Y LUEGO PRESIONE  ENTER"
3060 IF INKEY$="" THEN GO TO 306
0
3065 PRINT "PRESIONE PLAY Y LUEG
O ENTER."
3070 IF INKEY$="" THEN GO TO 307
0
3075 VERIFY "NOMBRES" DATA N$()
3080 VERIFY "TELEFONOS" DATA T$(
)
3085 PRINT "VERIFICACION CONCLUI
DA, DATOS OK"
3090 PAUSE 100: RETURN
4000 CLS : REM BUSCAR TELEFONO P
OR NOMBRE
4005 IF DATOS=0 THEN PRINT AT 11
,0; FLASH 1;" **NO HAY DATOS PAR
A OPERAR  ** ": PAUSE 200: RETUR
N
4010 BEEP .025,20: PRINT AT 1,6;
"BUSQUEDA DE DATOS";AT 11,3;"ENT
RE EL APELLIDO Y NOMBRE  DE LA
PERSONA  CUYO NUMERO DESEA
   USTED SABER."
4015 INPUT "APELLIDO Y NOMBRE: "
;B$
4020 IF LEN B$=20 THEN GO TO 404
0
4025 LET D=20-LEN B$
4030 FOR X=1 TO D-1: LET E$=E$+"
 ": NEXT X
4035 LET B$=B$+E$
4040 FOR F=1 TO DATOS
4045 IF B$=N$(F) THEN GO TO 4060
4050 NEXT F
4055 CLS : PRINT AT 11,0;"EL DAT
O DESEADO NO FIGURA EN EL REGIST
RO.                    ": BEEP
3,1: LET E$="": RETURN
4060 CLS : FOR A=1 TO 15: BEEP .
03,A: NEXT A: PRINT AT 7,6;"* DA
TO ENCONTRADO *"
4065 PRINT AT 11,0;"APELLIDO Y N
OMBRE";TAB 21;"TELEFONO"
4070 PRINT AT 13,0;N$(F);TAB 21;
T$(F)
4075 PRINT AT 19,0;"*PRESIONE UN
A TECLA PARA SEGUIR*"
4080 PAUSE 0: CLS : BEEP .025,15
: PRINT AT 11,0;"DESEA MODIFICAR
 EL DATO?(S/N)"
4090 INPUT "SU RESPUESTA: ";R$
4100 IF R$="N" THEN RETURN
4105 IF R$="S" THEN GO TO 4115
4110 PAUSE 100: RETURN
4115 CLS : BEEP .025,15: PRINT A
T 5,3;"QUE DATO ESTA INCORRECTO?
"
4120 PRINT AT 9,7;"1. EL NOMBRE.
";AT 11,7;"2. EL TELEFONO."
4125 INPUT "SU RESPUESTA: ";R$
4130 IF R$="1" THEN GO TO 4145
4135 IF R$="2" THEN GO TO 4150
4140 GO TO 4125
4145 CLS : BEEP .025,15: PRINT A
T 10,0;"ENTRE EL NOMBRE CORRECTO
.": INPUT "APELLIDO Y NOMBRE: ";
B$: CLS : LET N$(F)=B$: PAUSE 10
: BEEP .025,15: PRINT AT 7,6;"*
DATO CORREGIDO *": GO TO 4065
4150 CLS : BEEP .025,15: PRINT A
T 10,0;"ENTRE EL TELEFONO CORREC
TO.": INPUT "TELEFONO: ";C$: LET
 T$(F)=C$: CLS : PAUSE 10: BEEP
.025,15: PRINT AT 7,6;"* DATO CO
RREGIDO *": GO TO 4065
5000 CLS : REM BUSCAR NOMBRE POR
 TELEFONO
5005 IF DATOS=0 THEN PRINT AT 11
,0; FLASH 1;" **NO HAY DATOS PAR
A OPERAR **  ": PAUSE 200: RETUR
N
5010 BEEP .025,15: PRINT AT 11,0
;"ENTRE EL TELEFONO DE LA PERSON
A CUYO NOMBRE QUIERE UD. SABER"
5015 INPUT "TELEFONO: ";C$
5020 IF LEN C$=10 THEN GO TO 504
0
5025 LET D=10-LEN C$
5030 FOR F=1 TO D-1: LET E$=E$+"
 ": NEXT F
5035 LET C$=C$+E$
5040 FOR F=1 TO DATOS
5045 IF T$(F)=C$ THEN GO TO 5060
5050 NEXT F
5055 PRINT AT 11,0;"EL DATO SOLI
CITADO POR UD.NO FI-GURA EN EL R
EGISTRO.                ": BEEP 2,1: L
ET E$="": RETURN
5060 CLS : FOR X=1 TO 15: BEEP .
025,X: NEXT X: PRINT AT 7,6;"* D
ATO ENCONTRADO *"
5065 PRINT AT 11,0;"APELLIDO Y N
OMBRE";TAB 21;"TELEFONO"
5070 PRINT AT 13,0;N$(F);TAB 21;
T$(F)
5075 PRINT AT 19,0;"*PRESIONE UN
A TECLA PARA SEGUIR*"
5080 PAUSE 0: LET E$="": RETURN

6000 CLS : REM LISTAR TODO EL IN
DICE
6005 IF DATOS=0 THEN PRINT AT 11
,0; FLASH 1;" **NO HAY DATOS PAR
A OPERAR **  ": PAUSE 200: RETUR
N
6010 PRINT AT 10,0;"ESPERE POR F
AVOR, ESTOY ORDENAN-DO EL INDICE
 POR ABECEDARIO.": FOR X=1 TO 15
: BEEP .025,X: NEXT X
6015 FOR X=1 TO DATOS-1
6020 FOR Y=X+1 TO DATOS
6025 IF N$(X)>N$(Y) THEN GO TO 6
040
6030 NEXT Y
6035 NEXT X
6037 GO TO 6055
6040 LET G$=N$(X): LET N$(X)=N$(
Y): LET N$(Y)=G$
6045 LET G$=T$(X): LET T$(X)=T$(
Y): LET T$(Y)=G$
6050 GO TO 6030
6055 CLS : BEEP .025,15: PRINT A
T 10,0;"INDICE ORDENADO.DESEA UN
A COPIA EN LA IMPRESORA?(S/N)"
6060 INPUT "SU RESPUESTA: ";R$
6065 IF R$="S" OR R$="N" THEN GO
 TO 6075
6070 GO TO 6060
6075 CLS : PRINT AT 1,3;"LISTA C
OMPLETA DEL INDICE": PRINT : IF
R$="S" THEN LPRINT "LISTA COMPLE
TA DEL INDICE": LPRINT
6080 FOR F=1 TO DATOS
6083 IF N$(F)="
     " THEN GO TO 6095
6085 PRINT N$(F);" ";T$(F)
6090 IF R$="S" THEN LPRINT N$(F)
;" ";T$(F)
6095 NEXT F
6100 PRINT '''"*PRESIONE UNA TEC
LA PARA SEGUIR*": PAUSE 0: RETUR
N
7000 CLS : REM ESCAPAR DEL PROGR
AMA
7010 FOR X=20 TO 1 STEP -1: BEEP
 .025,X: NEXT X: PRINT AT 7,0;"P
ROGRAMA PARADO, PARA QUE VUELVAA
 ANDAR PRESIONE CAPS S Y LUEGO E
NTER. EL PROGRAMA VOLVERA A AN-D
AR CON LOS DATOS INTACTOS."
7020 STOP
9000 RESTORE : REM DEFINICIONE D
E VARIABLES Y DE GRAFICOS
9005 LET C=1
9010 READ A$: FOR F=0 TO 7: READ
 A: POKE USR A$+F,A: NEXT F
9020 IF C=9 THEN GO TO 9070
9030 LET C=C+1: GO TO 9010
9040 DATA "A",0,0,0,0,0,31,63,12
7,"B",0,0,0,0,0,255,255,255,"C",
0,0,0,0,0,248,252,254
9050 DATA "D",127,124,124,0,0,1,
3,7,"E",255,195,195,195,195,255,
255,129,"F",254,62,62,0,0,128,19
2,224
9060 DATA "G",15,31,63,127,127,1
27,127,0,"H",189,189,189,189,153
,129,255,0,"I",240,248,252,254,2
54,254,254,0
9070 DIM N$(50,20): DIM T$(50,10
)
9080 POKE 23658,8: POKE 23609,13
: LET DATOS=0
9090 LET E$=" ": LET S=0
9100 PRINT AT 5,8;"*** ESTE  ES
***"
9105 PRINT AT 7,0;"** UN INDICE
TELEFONICO PARA **"
9110 PRINT AT 10,6; INVERSE 1;"T
IMEX SINCLAIR 2068"
9115 PRINT AT 13,6;"ABC";TAB 14;
"ABC";TAB 22;"ABC";AT 14,6;"DEF"
;TAB 14;"DEF";TAB 22;"DEF";AT 15
,6;"GHI";TAB 14;"GHI";TAB 22;"GH
I"
9120 PRINT AT 18,0;"*PRESIONE UN
A TECLA PARA SEGUIR*"
9125 IF INKEY$="" THEN GO TO 912
5
9130 RETURN
```

". . . Este programa lo hice originariamente con una capacidad de almacenamiento de 50 nombres de 20 caracteres de largo y 50 teléfonos de 10 caracteres de largo. Así es como se los envío y *ocupa aproximadamente 10 Kbytes* de memoria.

–Es un programa de aplicación.

–Este programa hace actuar a la computadora como un índice de teléfonos y sirve para registrar nombres y teléfonos.

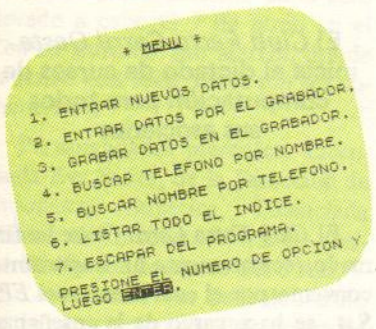


Modo de utilización

El modo de utilización es muy simple, sólo basta con seguir las instrucciones que va dando la máquina.

Instrucciones

. . . Lo primero que se ve es la presentación y luego de presionar cualquier tecla, se accede al *Menú* que ofrece las siguientes opciones:



Opción 1: Para entrar datos, sólo basta con entrar el apellido y nombre y luego el teléfono de alguna persona. Una vez registrado, el programa ofrece la posibilidad de modificar el dato.

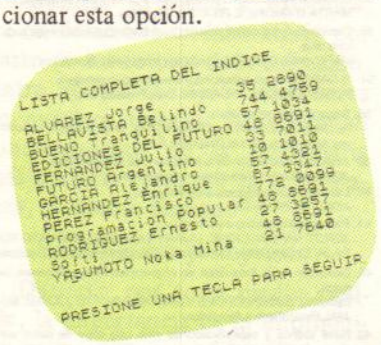
Opción 2: Permite entrar datos que fueron previamente grabados.

Opción 3: Permite grabar los datos del registro en una cinta. Primero graba los nombres y luego los teléfonos. Esta opción no puede ser utilizada si no hay datos (la computadora no permite usarla).

Opción 4: Permite buscar el teléfono de una persona a través de su nombre. Si el dato es encontrado, se ofrece la posibilidad de modificarlo. Si no hay datos, la computadora no permite seleccionar esta opción.

Opción 5: Permite buscar el nombre de una persona a través de su teléfono. Si no hay datos, la computadora no permite seleccionar esta opción.

Opción 6: Permite listar todo el índice que es previamente ordenado alfabéticamente. Se nos ofrece además la posibilidad de copiar los datos en la impresora. Si no hay datos la computadora no permite seleccionar esta opción.



Opción 7: Permite "escapar" del programa y explica cómo volver a él con los datos intactos.

Sugerencias:

- Si se desea ingresar *más de 50 datos,* modificar las líneas

2045 FOR F = 1 TO *....
9070 DIM N$ (*,20): DIM T$ (*, 10)
(* = cantidad máxima de datos a ingresar)

- Si se desea *anular el ruido* que hacen las teclas al ser presionadas, reemplazar el paso:

9080 POKE 23658,8:
POKE 23609,0: LET DATOS = 0

Advertencias:

- El programa, tal cual está, no permite ingresar más que 50 datos. Si se trata de ingresar más datos, dará error (SUBSCRIP WRONG).
- El programa utiliza letras mayúsculas. No utilizar minúsculas.

Tabla de Variables

O$	Variable que toma el número de opción elegida
B$	Toma el apellido y nombre entrados y luego los transfiere a N$ ( )
C$	Toma el número de teléfono entrado y lo transfiere a T$ ( )
DATOS	Cuenta la cantidad de datos existentes
N$()	Vector que acumula todos los nombres y apellidos
T$()	Vector que acumula todos los teléfonos
R$	Toma respuestas afirmativas o negativas.
S	Cuando se cargan los datos por el grabador, cuenta cuántos hay y transfiere, el valor a DATOS
D	Cuenta qué diferencia de largo hay entre el dato entrado y los registrados
E$	Acumula espacios vacíos que luego serán sumados al dato entrado para poder compararlo.
G$	Actúa como variable auxiliar en el ordenamiento alfabético de los datos
C	Es usada en la definición de gráficos como contador
A$	Toma la letra de los gráficos a definir
A	Toma los números de los gráficos a definir

Softinotas:

1. Al cargar las líneas de instrucciones 9040 *a* 9060 *y la instrucción número* 9115 *debemos presionar las teclas* (CAPS SHIFT) + (GRAPHICS) *(simultáneamente), antes y después de tipear los caracteres gráficos* A, B, C, D, E, H, I *(con minúsculas). Con estos caracteres gráficos (definidos por el usuario) se "arma" el dibujo del teléfono que aparece en la pantalla de la presentación y si se cargan erróneamente también saldrá mal dibujado dicho teléfono.*

2. Mis felicitaciones ***Francisco****, no sólo por tu programa, sino también por tu prolijidad y la completa documentación que nos acompañaste. Nuestros amigos lectores, agradecidos!!!*

Chau, ***Francisco*** *(va cheque por correo). . .*

Softi



USUARIA '85

Paralelamente con la exposición EXPOUSUARIA '85 se desarrolló en el Hotel Sheraton, durante los días 13 al 17 de mayo de 1985, el III Congreso Nacional de Informática y Teleinformática - USUARIA '85

Dada la importancia del mismo en lo referente a la divulgación y perfeccionamiento de la educación y el software en nuestro país exponemos a continuación la lista completa de los trabajos presentados y aprobados por el Comité Académico del Congreso, para conocimiento de nuestros lectores.

Trabajos presentados y aprobados por el Comité Académico del Congreso

1. Un sistema modular computarizado para el proyecto de centrales de generación de energía y una experiencia de transferencia de tecnología.
   JOSE VOVCHUK y JORGE NOVIZKI, Empresa Nuclear Argentina de Centrales Eléctricas.
2. Extensión de memoria por multiplicación de bloques.
   DANIEL COMESAÑA
3. Aproximación a un sistema que provea información para la toma de decisiones.
   JORGE ROSSI
4. Dirección estratégica de sistemas de información.
   EDUARDO JOSE BARRAL
5. Sistema de adquisición de datos en agencia de noticias, con microprocesador.
   ROBERTO KALIMAN y EDUARDO COHEN, Universidad Nacional de Tucumán.
6. Elección de las herramientas de desarrollo, en enseñansa asistida por ordenador.
   MARTA D'AMORE, E.R.I.A. (España).
7. Metodología de realización de un plan general de automatización de oficinas.
   JOSE MANUEL VAZQUEZ, E.R.I.A. (España).
8. Sistema de información para la toma de decisiones, en el desarrollo de productos.
   ARTURO J. RODRIGUEZ PONTI
9. Aplicación de técnicas de procesamiento de datos para la actividad de evaluación de performance de combustible nuclear.
   DEMETRIO AGUERO y MARIA ISABEL LOPEZ PUMAREGA, Comisión Nacional de Energía Atómica.
10. Informática aplicada a obras de ingeniería y arquitectura.
    ARIEL R. MAGATELLI, CECILIA MARIA BLANCO y FRANCISCO DE LA IGLESIA, Consultores para Empresas S.R.L.
11. Perspectiva y aplicación de la computadora en capacitación empresaria.
    MARIO BERNARDEZ, Pistrelli Diaz y Asociados S.A.
12. Software de cuarta generación.
    CLAUDIO SCHICHT, Price Waterhouse & Co.
13. Administración de obras sociales.
    HECTOR SALAZAR, Organización de Consultores Argentinos para el Desarrollo S.A.
14. Modelo para la proyección y control de la gestión de empresas agropecuarias.
    CIRO ECHESORTU, Delotte Haskins & Sells.
15. Sistema de información bibliográfica a través del computador.
    CONSTANTINO MENELLE, J. L. GHERARDI y F. LORENCES, Gas del Estado.
16. La microcomputadora en la enseñanza de las ciencias sociales en el nivel de post grado.
    LEOPOLDO SCHAPIRA, Universidad Nacional de Córdoba.
17. Una metodología para el desarrollo de sistemas.
    HERNAN HUERGO, Price Waterhouse & Co.
18. Apresentacao formal da "Astros". Una linguagem quase nao procedural.
    ROBERTO CARLOS MAYER, (Brasil).
19. Un modelo de comercialización de productos informáticos.
    LUIS SCHVIMER, Proceda S.A.
20. Selección e implantación de un sistema CAD/CAM.
    MANUEL PERALTA, Erdisa S.A.
21. Uma abordagem progmatica para a construcao de bases de datos.
    MARCELO FERNANDES, PAULO GUILHERME LAZZAROTTO DE ALMEIDA, ROBERTO MARCIO SANT ANDRE y HELVECIO ALVIM, Telecomunicacoes de Minas Gerais S.A. (Brasil).
22. Un marco de referencia para la concepción y desarrollo de sistemas de apoyo a la toma de decisiones.
    HECTOR RICARDO ACEVEDO, Universidad Técnica Federico Santa María. (Chile).
23. Informática en la administración primaria de la salud.
    ALBERTO E. BARBIERI, Consultora Hispanoamericana S.A.
24. Un estudio de la carga de ejecución requerida para el manejo de espacio en memoria por un protocolo de transporte.
    JOSE RAMIRO MONTEALEGRE, Carleton University, (Canadá).
25. Software para el procesamiento de procedimientos de decisión lógica secuencial.
    J.J. PLUS, V. GONZALEZ y F.A. MAMIAS, Conicet (UNR).
26. Informática educativa: su planteo en la educación presencial y sus posibilidades en la educación a distancia.
    JULIO C. A. GOMEZ y GUILLERMO F. MARTIN.
27. Sistema provincial de informática jurídica legislativa.
    JULIA E. SCARPINO, HORACIO D. ROA y VICTOR KAUFFORD
28. Informática en la bioquímica. Un caso
    STELLA MARIS VALIENTE, Universidad Nacional de Mar del Plata, Facultad de Ingeniería
29. Pautas para una legislación sobre programas de ordenador.
    RODOLFO A. IRIBARNE y CRISTINA M. L. DE CARJUZAA
30. Bases para una política nacional de informática jurídica.
    CRISTINA M. L. DE CARJUZAA y RODOLFO IRIBARNE
31. Metodología de planejamiento de recursos de informacao.
    PAULO GUILHERME LAZZAROTTO DE ALMEIDA, ROBERTO MARCIO SANT ANDRE y FERNANDO FRANCA, Telecomunicacoes de Minas Gerais S.A. (Brasil).
32. Automatización en la central hidroeléctrica "Arroyito".
    ENRIQUE HUMBERTO DIAZ
33. La auditoría informática y su entorno de realización.
    MARTA D'AMORE, E.R.I.A. S.A (España).
234. La planificación de la industria y la formación del directivo.
    JAIME LAVIÑA, E.R.I.A. S.A. (España).
35. Un marco metodológico completo para los servicios informáticos.
    MARTA D'AMORE, E.R.I.A. S.A. (España).
36. Factibilidad, requisitos y pautas para la computarización de auditoría dica.
    MARIA INES SCIUSCO, Sociedad Informática Biomédica
37. El CAD/CAM y su aplicación a la exploración petrolera.
    RUBEN ENRIQUE ANDREANI, Data Proceso S.A.
38. Seguridad en informática.
    HECTOR SALAZAR y JORGE ETKIN, Organización Consultores Argentinos para el Desarrollo S.A.
39. Análisis y evaluación de la productividad administrativa.
    JORGE BUJAN, Pickenhayn y Asociados.
40. Las microcomputadoras en los hogares.
    ADRIANA FRANCO
41. Presupuesto público e informática.
    JORGE NARDACCIONE, LILIANA SANANES y MAURICIO RUBAJA, Organización Consultores Argentinos para el Desarrollo
42. Evolución de la Seguridad en el procesamiento eléctronico de datos.
    BEATRIZ SUBERBORDES, FERNANDO DOTTO y JORGE BUJAN, Pickenhayn y Asociados
43. Base teórica y especificaciones para una base de datos con manipulación semántica.
    MANUEL MOLINA y ELIDA GARCIA ROZADO, NCR Argentina
44. Base de datos jerárquica y base de datos multiplex. Comparación de dos experiencias.
    JORGE URBANITSCH y SILVIA DE LA RUA DE CATTANEO, Organización Consultores Argentinos para el Desarrollo S.A.
45. Desarrollo de un sistema CAD/CAM para P.C.
    BASILIO JEZIENIECKI, Techint S.A.
46. Introducción a los lenguajes para el diseño de software.
    RAUL MARTINEZ y EDUARDO MIRANDA.
47. Técnica de diagnóstico de los sistemas de bombeo mecánico.
    J. C. NAPOLITANO, J. C. ZAJACZKOWSKI y R.D. CASTELO, Yacimientos Petrolíferos Fiscales.
48. El planeamiento y la informática en un empresa petrólera estatal.
    RODOLFO PABLO FERNITZ, Yacimientos Petrolíferos Fiscales.
49. Utilización del computador en enseñanza personalizada.
    MARIA VICTORIA GHISALBRTI DE LEWIS, Instituto Juan XXIII
50. Las microcomputadoras en la oficina.
    ADRIANA FRANCO y PATRICIA IURCOVICH
51. Modelización global al planamiento estratégico de la gestión de un Centro de Computos para la decada del noventa. Modetran.
    ALBERTO GIAMPIANI y LUIS RICOTTI
52. Programación libre de estructura de datos.
    LILIANA FAVRE, OSCAR MAYORGA, LAURA RIVERO y HUGO RYCKEBOER, Universidad Nacional del Centro de la Provincia de Buenos Aires, Facultad de Ciencias Exactas, Departamento de Computación y Sistemas.
53. La auditoría en el desarrollo de aplicaciones: la metodológia.
    JORGE OMAR ALIJO, IBM Argentina S.A.
54. Las computadoras personales en la educación.
    VICTOR FONTANA, IBM Argentina S.A.
55. Sistema de apoyo de decisiones e información gerencial.
    JORGE TEODORO ARES, IBM Argentina S.A.
56. Un sistema bancario de bases de datos.
    GUSTAVO M. MURMAN, IBM Argentina S.A.
57. La informática en el ámbito científico y de ingeniería.
    GUSTAVO M. MURMAN, MARIO A. PAZOS y ALDO RAMOS, IBM Argentina S.A.
58. Educación y readaptación del personal de sistemas de información.
    ALEJANDRO PACECA, IBM Argentina S.A.
59. Sistemas expertos en educación. El tutor de estudios inteligente.
    DANIEL S. TKACK, IBM Argentina S.A.
60. Desarrollo de sistemas que utilicen técnicas de bases de datos y procesamiento distribuido. CECILIA VIDAL DIAZ, IBM Arg.
61. Un caso práctico de auditoría de sistemas: verificación de un sistema de presupuesto fiscal.
    JUAN ARTURO IACAPRARO
62. Seguridad de datos y procesamiento.
    JUAN CARLOS RUGILO, Price Waterhouse & Co.
63. El lenguaje didáctico BACKUS.
    FRANCISCO IBAÑEZ y HUGO RYCKEBOER
64. Posibilidades de los sistemas gráficos interactivos para diseño y fabricación.
    MARCELO SOL, IBM Argentina S.A.
65. Plan de informática región N.E.A., políticas, estratégias y metas.
    JOSE GIMENEZ, Centro de Estudios e Investigación en Matemática Aplicada.
66. Desarrollo de una aplicación para el control del cumplimiento de la producción terminada.
    AGUSTIN JORDAN PALMA, RUBEN SCHISTER y ALEJANDRO VILLEMUR, Alpargatas S.A.
67. Sistemas integrados de gestión empresaria en el área de refinación.
    FERNANDO REY (Coord) a) Modelo matemático, LUIS ALBERTO HEVIA y ARNALDO NESTOR MOURAS b) Balance de mesa. ANTONIO VIDELA y LUIS SCARPIN c) Control de gestión JOAQUIN MATIAS y JUAN CARLOS MARTINEZ, Yacimientos Petrolíferos Fiscales.
68. Control operativo de unidades catalíticas comerciales.
    HUGO IANTORNO, Yacimientos Petrolíferos Fiscales
69. Un enfoque sistemático del cerebro.
    WALTER FRITZ y RAMON GARCIA, Sociedad Argentina de Informática e Investigación Operativa S.A.
70. Una estrategia para el desarrollo de usuarios de microcomputadores.
    JOSE A. DO RIO, Price Waterhouse & Co.
71. Un enfoque metodológico para el desarrollo de modelos de planeamiento.
    CARLOS CAPELLA y PATRICIO SERRA, Price Waterhouse & Co.
72. Algoritno simple para la eliminación de las lineas ocultas de modelos para la computación gráfica.
    EDUARDO LAITANO y SERGIO IDELSOHN, Instituto de desarrollo tecnológico para la Industria Química.
73. Arquitectura de redes.
    CARLOS MARTINEZ, IBM Argentina S.A.
74. Arquitectura de Sistemas de oficina.
    HUGO MONDELO, IBM Argentina S.A.
75. Técnicas actuales en la ingeniería de software.
    EMIR GARZA, IBM Argentina S.A.
76. Estudio de la política ganadera con un modelo de simulación dinámica. ANDREW BREITER
77. La simulación dinámica de sistemas como enfoque para la solución de problemas complejos.
    ANDREW BREITER

Si algún o algunos de estos trabajos resultaran de su interés, escríbanos al SOFTI CLUB. Si hubiera marcada preferencia por algún artículo en especial, lo publicaremos en la revista. Si no, le prepararemos una fotocopia "al costo".

Experamos noticias.

El *Club Ferro Carril Oeste* inicia el dictado de cursos de computación para todos sus asociados

El ***Club Ferro Carril Oeste*** ha firmado recientemente un importante convenio por el cual la empresa ***EPI S.A.*** se hace cargo de la enseñanza de computación para los socios de dicho Club.

Para lograr este objetivo ***EPI S.A*** ha instalado un aula con 14 microcomputadoras ***CZ 1500*** de la firma ***Czerweny Electrónica***, 14 monitores ***Hitachi*** de la firma ***Radio Victoria Fueguina*** y el amoblamiento de ***Aguilar Equipamientos.***

Los cursos que se dictarán son de Lenguaje y Programación Basic, desarrollándose de lunes a viernes de 9 a 21 hrs. y los días sábados de 9 a 13 hrs.,

A los cursos podrán asistir niños desde los 10 años, adolescentes y adultos, contando cada alumno con un equipo.

Esta experiencia es inédita en dicha Institución y la ubica por sus características entre los centros más importantes nacionales e internacionales, donde la conjugación de las actividades físicas e intelectuales son sinónimo de la formación integral de las personas.

Informes e inscripción: Club Ferro Carril Oeste, Tesorería, Cucha Cucha 350, T.E. 431-0017/4787/6673.

Y dale, y dale, y dale Ferro... dale... (*Softi*)

Ya está en marcha el **Primer Concurso Nacional de Programas en microcomputadoras para Discapacitados** organizados por E.C.I. Editorial Consultora en Informática!!!...

El 20 de junio quedó cerrada la lista de premios de este novedoso aporte a la cultura iniciado por E.C.I. cuyo detalle damos aquí, así como las Bases del Concurso.

Softi estuvo presente en el acto llevado a cabo el 7 de junio en el Centro Cultural San Martín, donde se lanzó esta idea. Estos son sus comentarios:

Personalidades presentes: Licenciada Eva Sarka, de la Dirección de Enseñanza Superior, Licenciada Massari del Instituto Ward, Dr. Carlos Distilo del Instituto del C.AD. Y S., Licenciada Nora Leda de Distilo de la Asociación de Psicólogos de Bs. As., Directivos de varios colegios y sociedades de ayuda al discapacitado, pudiendo mencionar entre ellos a la escuela "Mi Camino" y la "Asociación Argentina de Padres de Sordos Hipoacúsicos".

Empresas de computación líderes en el mercado, y medios de prensa radiales, gráficos y televisivos.

Inicia el acto el señor Alfredo Tramonte, Direcfor administrativo del Centro Cultural, quien entre otras expresivas y cálidas palabras dijo: "dadas las características especialísimas del libro, éste ha sido incluido en la programación oficial de esta alta casa de cultura".

En un marco de gran expectativa disertaron los señores Pablo Solowiej, Daniel Sentinelli y Juan Carlos Zahrebelnyj.

El Señor Solowiej se refirió al temario del Tomo IV, destacando los mejores programas y presentó el lanzamiento del primer concurso a nivel nacional de programación para discapacitados; el primer aniversario de la Editorial y como elemento de integración, el primer libro musical del país.

El señor Sentinelli se refirió en su alocución al apasionante tema de Inteligencia Artificial y Robótica, ya en desarrollo en laboratorio por parte de los autores de la colección. Entre otras cosas djo: "intentamos en nuestros trabajos reproducir el funcionamento de la maquinaria más perfecta existente en la naturaleza, el cerebro humano".

Finalidades y objetivo:

Difundir por todo medio posible, la importancia de la informática en la cultura y educación, y las grandes posibilidades que brinda para el caso específico de los discapacitados.

Un programa creado por un discapacitado y otro realizado por alguien que no lo es, no tienen diferencia alguna para quien desconoce su origen, por lo tanto la computación es el medio mas completo que tenemos para recordar la igualdad del ser humano, resaltando la capacidad creativa, de elaboración y técnica que todos podemos tener.

Bases del Concurso

Podrá intervenir toda persona discapacitada y de cualquier edad o sexo.

No podrán intervenir, integrantes autores, colaboradores de cualquier naturaleza y/o familiares, de personas que pertenezcan a la ***"Editorial Consultora en Informática"*** y/o ***"Editorial Ediciones del Futuro" S.A.***

Los *programas* deben ser juegos (no bélicos), entretenimientos, de aplicación general o particular, de enseñanza o didácticos.

Se designaron como integrantes del *Jurado:* por Editorial Consultora en Informática, a los señores ***Daniel Sentinelli y Omar González***, por Ediciones del Futuro S.A. al Dr. ***Héctor J. Silva*** y al ***Sr. Enrique H. Ramírez***, por Q.S.P. S.A. al Sr. ***Ricardo Pankiewicz.***

La *resolución final* del jurado será documentada ante Escribano.

Se tendrá en cuenta para la *evaluación* los siguientes aspectos: Originalidad y creatividad de la idea. Presentación en pantalla. Interés y atractivo del juego, en su caso. Utilidad del programa, para programas de aplicaciones. Facilidades de uso. Documentación del programa, principalmente para aplicaciones.

Se admitirá *un solo programa* por cada concursante.

Los trabajos presentados, deben ser *Exclusivamente* de *Creación* del participante, pudiendo ser en *lenguajes* Basic, Basic extendido o Logo, posibles de interpretar por las microcomputadoras TI-99/4A o TK85. Se aceptarán en cassette o diskette para el primer caso y en cassette para el segundo.

Deberán ser *entregados* en todos los casos por el autor y un representante del mismo en caso de ser necesario, en las oficinas de la Editorial Consultora en Informática, Avda. Rivadavia 4161, piso 1ro. Of. 5, Ramos Mejía, o en su defecto podrán ser enviadas por correo certificado.

Con la entrega *se acompañará* el listado completo del programa (por impresora o escrito a mano), firmado por el autor y alguien que lo represente.

Se fija el siguiente ordenamiento

OFERTA COMMODORE

PRECIO POR UNIDAD U$S

Commodore 64 330
Commodore 16 ... consultar
Data Set 120
Disk Drive 430
Disketts x 10 38

PRECIO ESPECIAL POR MAS DE 5 UNIDADES

T.E. 52-4336 (dejar mensaje)
Soler 4031 P.B. "8"
de 21 a 23 hs.

de fechas: *Apertura del Concurso:* 7 de junio de 1985.

Cierre del Concurso: 30 de septiembre de 1985 a las 12 hs.

Veredicto del Jurado: 31 de octubre, se aceptarán impugnaciones por telegrama o carta documento hasta 30 días después.

Entrega de Premios: 16 de diciembre de 1985.

Las diligencias notariales están a cargo del Sr. Escribano Público Juan Carlos Lyhiardopoulos.

E.C.I. se reserva el derecho, para el caso de ser necesario, de solicitar a los ganadores un certificado médico extendido por autoridad competente.

Los *soportes físicos* (cassettes o diskettes) serán devueltos al concursante a los 30 días de declarar finalizada la evaluación final, y dictamen definitivo de los ganadores del concurso, quedando disponibles en las oficinas de E.C.I. para su retiro. Dicho retiro podrá efectuarse hasta 60 días después del dictamen. De no hacerlo la totalidad del material entregado podrá ser destruido y/o darle otro destino de acuerdo al criterio de E.C.I., sin derecho por parte del autor a reclamo alguno.

La entrega de los programas a la Editorial Consultora en Informática, no significa derecho a reclamo de pagos por autor. E.C.I., solo se compromete a hacer llegar a todos los medios de prensa presentes en este acto, y a otros que estén a su alcance, el listado completo de los programas premiados, mencionando su origen, con nombres y apellido completos del autor, y solicitándoles en calidad de colaboración se publiquen.

La revista ***"Programación Popular"*** de la editorial Ediciones del Futuro S.A., ha sido designada por la Editorial Consultora en Informática, para publicar los programas y resultados de ganadores, en la edición que siga a la fecha final del fallo.

Los *premios* serán entregados ante Escribano Público.

Para el supuesto caso de que alguno de ellos quedara vacante por circunstancias previstas o nó, éste será entregado a la Sub-Secretaría de Informática del Ministerio de Educación, con el objeto que lo curse a la entidad que específicamente trabaje con discapacitados y que, a su juicio considere es la más conveniente.

Se recibirán materiales hasta la fecha prevista como cierre del concurso, labrándose acta ante Escribano Público, consignando cantidad de trabajos, como así nombres y apellido de autores y sus respectivos representantes según el caso.

La Editorial Consultora en Informatica notificará por carta documento a todos aquellos participantes que resulten beneficiados, sin perjuicio de las publicaciones anteriormente mencionadas.

Lista de Premios

1º premio: Una Computadora TK85, con generador de sonido programable, Joystick y software, donado por Q.S.P.

2º. premio: Un servicio de Videofilmación profesional, donado por Remember Video Producciones

3º al 4º premios: Becas para el aprendizaje del Basic, donadas por Sanwa S.A.

5º premio: 3 (tres) pasajes ida y vuelta a Colonia, República Oriental del Uruguay, donado por Ferritur S.A.

6º al 15º premios: Una colección de libros de Editorial Consultora en Informática.

16º al 17º premios: Suscripciones por un año de la Revista Programación Popular, donados por Ediciones del Futuro S.A.

18º a 20º premios: Programas para microcomputadora TK2000, donado por Q.S.P.S.A.

21º y 22º premios: Suscripciones por seis meses a la Revista Programación Popular, donador por Ediciones del Futuro S.A.

23º al 24º premios: Libros especializados en computación de autores varios, donados por Editorial El Ateneo.

25º al 27º premios: Basic para Niños, donado por Editorial Yenni S.R.L.

28º al 42º premios: Libros especializados en computación de autores varios, donados por Editorial El Ateneo.

De último momento!!! Estarían disponibles a la brevedad los *Floppy-disk* para la *CZSpectrum* en la Argentina

Fuentes habitualmente bien informadas aseguraron a ***Softi*** que la empresa Czerweny Electrónica de Argentina, lanzará próximamente a nuestr mercado, los floppy - disk para la CZSpectrum.

Esta disketera, de una capacidad de 280 Kb y de reducido costo, estarían disponibles incluso antes que los Microdrives, también anunciados para una próxima aparición en la Argentina.

Dado que esta información llegó al cierre de este número de inmediato vamos a reportear a las autoridades de esta empresa para obtener más precisiones.

Los días 16, 17 y 18 de julio del corriente año se celebrarán las *Primeras Jornadas de Educación y Computación* en el *Salón Auditorio Bs. As.* de esta Capital.

Estas Jornadas forman parte del programa de difusión de la informática implementado por la empresa EPI S.A. a la que se le sumarán en el mes de agosto Microcomputación y Medicina, septiembre Comercio Exterior y Computación, octubre Agro y Computación y en noviembre microcomputación para actividades financieras.

Cada una de ellas estará dirigida al público directamente interesado y su objetivo es el de difundir las aplicaciones y experiencias realizadas en cada una de las áreas de que tratan.

Acompañará en todos los casos, una muestra de elementos de computación denominada centro de información permanente que funcionará en Florida 683 de esta Capital. En la misma se exhibirán productos referidos a las jornadas que se desarrollen. Los días y el horario de funcionamiento serán de martes a sábado de 14 a 21 hs. Allí los esperará ***Softi.***



las cabezas de lectura hacia el soporte (discos de cabeza móvil).

ciclo de máquina, *machine cycle:* Es la función elemental de *acceso a la memoria* o localización de una dirección en la memoria central. Es una operación elemental que forma parte de todos los Ciclos de Ejecución, por ello tiene gran importancia su rapidez de realización en la expresión de velocidad de proceso de una Unidad Central (UCP).
Sinónimo de: **ciclo base**

cilindro, *cylinder:* Una serie de pistas (una de cada cara) en una unidad de discos situadas en un mismo plano vertical, de forma que se puede tener acceso a todas ellas con un solo posicionamiento (desplazamiento de las cabezas lecto-grabadoras).
Utilizar el concepto de **cilindro** tiene por objeto minimizar el movimiento de las cabezas.
El conjunto de datos contenido en un cilindro y, por tanto, accesible en un solo posicionamiento, se denomina **Area de búsqueda**

codificación, *coding:* Expresión de uno o varios caracteres según las normas de un código determinado.
Programación de un proceso, esquematizado en un Diagrama de Flujo, según un lenguaje concreto y de acuerdo con el correspondiente juego de reglas.
Sinónimo de **Escritura de un programa** (usualmente).

codificación simbólica, *symbolic coding:* Utilización de nombres simbólicos en una codificación, representativo de un concepto más amplio p/el programador. generalmente nemónicos (nemotécnicos).

código, *code:* Conjunto de reglas que determinan la expresión de los dígitos de un sistema de numeración por medio de una serie de símbolos preestablecidos
Tabla de doble entrada para la codificación de caracteres numéricos, alfabéticos, de puntuación, y especiales, asignándoles un número de fila y un número de columna, los cuales se denominan, respectivamente, parte DIGIT y parte ZONA del Código del carácter.
Así: Código **EBCDIC.** Código **ISO—USASCII.**

código decimal codificado binario, *binary coded decimal (BCD):* Conjunto de reglas que determinan la expresión de un número del sistema de numeración decimal, por medio de un grupo de elementos binarios.
Existe gran diversidad de códigos de representación binaria de números decimales, que pueden presentar diferentes ventajas de simplicidad, de detección de errores, de corrección de errores, de realización electrónica, etc.
Podría darse como clasificación de los **códigos BCD:**

1. **Ponderados:** A cada bit se

mayor sea la dimensión en bits del campo operación, mayor podrá ser el set de instrucciones de una máquina.
Sinónimo de: **Campo Función.**

campo operando, *operand field:* Parte de una instrucción que contiene la información destinada a identificar (por los órganos lógicos de la Unidad Central de Proceso) con qué operando u operandos se debe ejecutar la función indicada en el **Campo Función.** Estrictamente lo que contiene el Campo Operando es la "dirección de memoria donde está el operando" u operandos.
El aumento de la dimensión en bits del **Campo Operando** permite por una parte indicar en la misma instrucción los dos operandos de una operación, y por tanto hacer operativas todas las zonas o registros de la memoria; por otra parte, alcanzar direcciones más y más elevadas, y por tanto ampliar la dimensión direccionable directamente de la memoria central.

canal, *channel:* Parte de la Unidad Central constituida por elementos de la Memoria Central, de la Unidad Lógica, y de la ROM de microprogramas, que a modo de unidad de proceso independiente comanda la unidad de control de cada periférico.

canal multiplexor, *multiplexor channel:* Canal que puede atender transferencias de información con varias unidades de control de varios periféricos simultáneamente (en apariencia). Se aprovecha esta posibilidad para conectar varios periféricos relativamente lentos, como impresoras, etc. a un mismo canal multiplexor. Ver **Multiplexor.**

canal selector, *selector channel:* Canal que puede atender, cada vez transferencias de información a/de una sola unidad de control, de/a uno o varios periféricos generalmente rápidos. Ello no implica que no pueda tener varias conexiones, pero en el momento de transferir con una de ellas las demás se mántienen "en espera"

capacidad de almacenamiento, *capacity:* Límite máximo de la cantidad de información contenida en un soporte determinado, entendiendo como tal, sea una memoria central, sea una memoria auxiliar, sea un soporte de dispositivo periférico, sea una zona, registro o bloque de una de ellas.
Se mide en número de bytes (octetos), o en número de palabras de **n** bits.
Para medidas grandes se utiliza el múltiplo:

$$\text{Kbytes} = \text{Kb} = 1.024 \text{ bytes} = 2^{10} \text{ bytes}$$

Así:
Una memoria de 16 K bytes = 16.384 bytes.
Una memoria de 4 K palabras de 4 bits = 4.096 números de 4 bits
Una cinta magnética de 256 K bytes = 262.144 bytes.

carácter, *character:* Cada uno de los números, letras o símbolos de

puntuación que se pueden representar en 0 bits, tomando 4 bits como fila y 3 como columna de una tabla de hasta 16 filas por 8 columnas, considerada como código Hay también códigos de 7 hasta 32 bits.
Por extensión a cada posición fila-columna no correspondiente a un número, letra o símbolo, se le asigna un significado y se le llama también **Carácter.**
Sinónimo de: **Byte.**

carácter alfanumérico, *alphanumeric character:* Cada uno de los números 0 a 9, letras A a Z, y símbolos de puntuación según la definición anterior de carácter.
Ver **Alfanumérico.**

carácter de relleno, *filling character:* Carácter que se destina a aparecer impreso en reemplazo de los espacios "en blanco" que podrían aparecer en la impresión de un dato

carácter hexadecimal, *hexadecimal character:* Cada una de las 16 configuraciones que pueden presentar 4 bits, se nombran: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
Cuando se habla de un carácter hexadecimal se quiere indicar un conjunto de 4 bits (un **semibyte**) con un significado numérico como el anterior.

carácter separador, *separator character* Entre los caracteres no alfanuméricos de un código, hay algunos a los que se ha convenido en dar un significado de separadores de información. Su efecto input se encuentra un carácter separador se da por finalizado el input y se reemprende la ejecución del programa.
Normalmente utilizado en soportes no magnéticos y en cintas magnéticas FS, RS, US, CS del código ISO.

característica, *characteristic:* Ver **aritmética de coma flotnate.**

cargador, *loader:* Uno de los programas o módulos del Sistema Operativo que efectúa y controla la carga del programa del usuario.
Interviene cada vez que se requiere cargar en Memoria Central un segmento (Overlay) de programa, al cual cederá después el control.
Mientras el Sistema Operativo no lo necesita, el cargador puede no residir en Memoria Central, pero siempre está presente en el sistema.

casete, *cassette:* Cartucho contenedor de cinta magnética, de características físicas universales muy utilizado como soporte secuencial de poca capacidad y velocidad, pero muy bajo costo y fácilmente intercambiable.
En general se utiliza como soporte de programas.
Su capacidad puede ser aproximadamente desde 100 Kbytes a 256 Kbytes.

catódico, *cathody ray tube:* Calificativo frecuentemente asociado a la expresión Tubo de rayos catódicos (CRT), que es un elemento de representación visual en pantalla.

celda, *cell:* Grupo o conjunto de posiciones de memoria destinadas a ser tratadas como una unidad física y lógica.

celda de datos, *storage cell:* Dispositivo periférico de almacenamiento a modo de tambor de sectores intercambiables y, por tanto, de acceso directo. Su principal característica es la gran capacidad de almacenamiento (del orden de 400 Mbytes) aunque su tiempo de acceso es relativamente lento (de hasta 600 mseg.)

cibernética, *cybernetics:* Ciencia del control y la comunicación en animales y máquinas.
Ha sido también difundido por la escuela francesa como la técnica de hacer eficaz la acción.
Sinónimo de **Informática** (usualmente)

ciclo, *cycle:* Ver **bucle.**

ciclo de ejecución, *execution cycle:* Proceso completo de tratamiento de un dato en la UCP comprendiendo las fases de 1) interpretación de la instrucción, 2) búsqueda en Memoria Central del dato a tratar, y 3) tratamiento del mismo en función de la instrucción respectiva.
También puede describirse como el proceso de ejecución que media entre el inicio de la interpretación por la Unidad de Control de una instrucción, y la llegada a la Unidad de Control de la siguiente instrucción a interpretar y ejecutar.

ciclo de instrucción, *instruction cycle:* Parte del ciclo de ejecución en que la Unidad Lógica pone a disposición de la ROM la instrucción que va a ser interpretada y ejecutada.

ciclo de grabación, *recording cycle:* Proceso ejecutado desde que la Unidad de Control del Periférico recibe la orden de grabación, hasta que verifica que la misma ha sido ejecutada correctamente.
Incluye una transferencia de información con la Unidad Central, transporte físico del soporte a la posición de grabación, grabación y frenado o expulsión del soporte. En algunos casos la grabación va acompañada de una lectura de comprobación. Como transporte puede entenderse también el movimiento de aproximación de las cabezas de grabación hacia el soporte (discos de cabeza móvil).

ciclo de lectura, *reading cycle:* Proceso ejecutado desde que la Unidad de Control del Periférico recibe la orden de lectura, hasta que la misma ha sido ejecutada correctamente.
Incluye un transporte físico del soporte a la posición de lectura, lectura del mismo, transferencia de la información y frenado o expulsión del soporte.
Como transporte puede entenderse también el movimiento de aproximación de

Los servicios de Epi

CURSOS:

Sólo Epi le da un computador para Ud. solo

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

Suipacha 946 - 1er. Piso - Capital TE.: 311-8618

CASSETTES PARA

TI-99/4A

MICRODIGITAL

COMMODORE 64

COMPILADOR para TI-99/4A

FORMATOS DISPONIBLES

CASSETTECAS

x 4: $a 6.250.-
x 6: $a 8.200.-
x 12: $a 12.500.-

SISTEMA SKINPACK

118 Títulos
a $a 1.600.- c/u

SISTEMAS

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

Disponemos:

- Stock
- Facturación
- Clientes
- Cuentas Corrientes
- Contabilidad
- Listas de precios

... y también sistemas a su medida.

SE ACEPTAN TARJETAS DE CREDITO

EMPRESA PARA INFORMATICA

INSTITUTO: Suipacha 946 1er. Piso (1008) Capital.
VENTAS: Viamonte 1479 8° "B" (1055) Capital.
Teléfonos: 311-8618 y 49-7985.
Florida 683
Av. Corrientes 2198

Bernini & Valentini

Sepa exactamente cómo una computadora CZ Spectrum puede prolongar su cerebro.





1. Si usted es médico puede llevar la agenda de turnos de sus pacientes. Guardar los datos de cada uno y obtener así las liquidaciones para las Obras Sociales.

2. Si tiene una Agencia de Prode, quiniela y loterías, puede registrar la venta total de la semana, cuáles son los números más pedidos, los atrasados y los ganadores del Prode.
Con una CZ SPECTRUM, tendrá la grande en su local.



3. Si tiene un autoservicio puede mantener actualizado su stock de mercaderías, la rotación del mismo y ordenar su facturación.

4. Si tiene una agencia de autos, podrá llevar un registro completo de cada unidad; color, modelo, kilometraje y hasta modificar el precio al instante.



5. Si es constructor podrá tener rápidamente un panorama completo de cada obra. La compra de materiales por rubro, cálculos estadísticos, cantidad de personal, índice de rentabilidad.
Con la CZ SPECTRUM, comienzan hoy los cimientos de sus obras del año 2000.



6. Si administra consorcios, ingrese los gastos de cada edificio y obtenga en instantes el cálculo de las expensas.

7. Si vende pasajes podrá tener un control absoluto de todos los vuelos. Teniendo en cuenta los distintos tramos. Compañías. Tarifas en dólares y la transformación automática quincenal y mensual.

8. Si administra clubes, lleve el registro de socios.

9. Si tiene un restaurante, registre fácilmente los cambios de la lista de precios, ingrese los consumos por mesa y obtenga las adiciones a pedido.

10. Si tiene una inmobiliaria, lleve el registro de las propiedades en venta. Hasta qué suma puede estudiar una oferta. Qué zona le falta cubrir para ofrecer. Dónde están las propiedades. Qué características tienen. Ubicación, acceso, comodidades, precio.

11. Si su actividad es el comercio exterior, con la CZ SPECTRUM tenga en segundo datos fundamentales para su negocio; cantidad de toneladas cosechadas, precio del grano en los distintos mercados, cantidad de cabezas ingresadas a Liniers en los últimos años, su disponibilidad de mercaderías exportables al instante. CZ SPECTRUM inclina la balanza a su favor.



12. Con la CZ SPECTRUM, un aventajado estudiante como usted, resuelve rápidamente cálculos, ecuaciones o fórmulas y tiene a su alcance cientos de datos de cualquier materia. Y luego, para despejar la mente, corra una carrera de Fórmula 1, rechace un ataque extraplanetario o simplemente dé un paseo en su avión. CZ SPECTRUM la simplificación del estudio.



13. Y cualquiera sea su actividad, con la CZ SPECTRUM, la computación empieza a formar parte de su vida de todos los días para hacerla más fácil.
Porque eso es la tecnología CZ: computación al alcance de todos.

CZ

Spectrum

La prolongación de su cerebro

Adquiérala en los Distribuidores de la

Red CZ

Distribuye en todo el país: CZERWENY ELECTRONICA S.A. Av. de Mayo 963 - 3° P (1084) Capital Federal