Precio ₳ 1.80

programación popular

Nº 10

REVISTA MENSUAL DE COMPUTACION

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

Galería de Máquinas

Características de las principales micros

Ganá vos también una CZ 2000 (Spectrum) de 48 K y otros importantes premios

Código de Máquina

Aprovechemos al máximo el Z-80

CONCESION Nº 1949
FRANQUEO PAGADO
CONCESION Nº 840
TARIFA REDUCIDA

VIDEO JUEGOS

en castellano
para microcomputadoras



CARGA GARANTIZADA

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

COMPATIBLE PARA

- SPECTRUM
- TK 90

PROXIMAMENTE SINCLAIR 1000 TK 83/85 CON SISTEMA CONCORD Y COMMODORE 64 CON TURBO PLUS

(*) SISTEMA CONCORD

Reduce tiempo de carga

Instrucciones con tono de ajuste para poner a punto el cabezal del grabador. Instrucciones para limpieza del cabezal.

PIDALOS EN NEGOCIOS DE COMPUTACION Y DISQUERIAS

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

Dirección General

Dr. Héctor J. Silva

Dirección Financiera
Enrique H. Ramírez

Dirección Comercial
Ricardo J. Baiele

Colaboran en este número:
Eduardo Grosclaude
Andrés Bitenc
Roberto S. Chervo
Horacio Domizzi
Pablo Ferrante
Hugo R. Ortiz
Diego Rodríguez
Ing. Ernesto O. Selzer
Christian Tottino
Ing. Raúl D. Vera
Daniel Weselka

Arte y Diagramación:
Daniel P. Unamuno
Francisco J. Fenili
Lucas A. Tischlangian

Número 10

El tiraje y circulación de la revista programación popular es de

21000 EJEMPLARES

Según verificación y certificación efectuadas por Escribano Público y Contador Público Nacional en el mes de enero de 1986

Fotocomposición y Películas
Photolettering SA
Fotocromía
Gráfica Boedo SA
Impresión
Fabril Financiera SA
Distribución Kioscos Capital y Gran Bs.As.
Prícolo SA
Distribución Kioscos Interior
Rvela SAIC y F

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

EDITORIAL

Se va el verano, y con él quedan atrás las vacaciones, el descanso, el "desenchufe" de una difícil realidad. Volvemos a conectarnos entonces con el mundanal ruido. Vemos problemas, pero también hay mucha gente buscando soluciones. Percibimos y vivimos descontentos, pero también aprendemos a disentir en democracia.

Pero en PP no hemos descansado demasiado, como lo prueban estas jugosas páginas de nuestro número 10, con 11 nuevos programas para las principales micros del país (faltaría la TK 2000, solamente). También les ofrecemos una "Galería de máquinas" con las principales características de todas las micros, Curso Basic 7, Lenguaje de Máquina 4, y muchas notas más (me olvidaba, hay una ***sorpresa*** *en la contratapa...).*

Marzo se nos antoja un buen punto de partida. Estamos algo más relajados, descansados y puede ser un mes propicio para buscar de reimpulsar la maquinaría dormida, cada uno desde nuestra humilde posición de trabajo. Seguimos trabajando para hacer crecer nuestra PP (esperamos agregar más páginas) y ampliar los servicios del Softi Club.

Un último comentario: no nos gustó (y a muchos amigos tampoco) el resultado de la Prueba que hicimos cambiando parte del papel interior del número 9 de PP, pretendiendo una mejor impresión y menor reflejos y transparencia a un costo menor. Son cosas de la búsqueda de mejorar, pero el que no hace, seguro que no se equivoca...

Hasta el número 11.

Dr. Héctor J. Silva

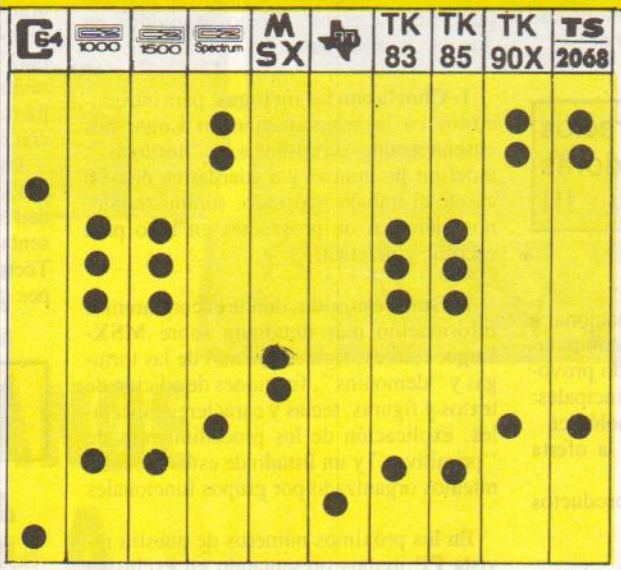
NOVEDADES

Noticias, anticipos y comentarios de nuestro mundo de la microcomputación 4

PROGRAMACION

Curso Basic (7)
En este número: Programación Estructurada: **REM, READ, DATA, RESTORE** 11
Lenguaje de Máquina (4)
En esta entrega: Sistema Binario: **INC, DEC** 40
Manejo de Archivos (4)
Veremos: Ordenamientos sucesivos 30

PROGRAMAS



Alpha 1 18
Bola Loca 33
Congelador de Listados 17
Costos de Fabricación (fin) ... 34
Golf 38
Grabador digital 16
Gráfico de barras 44
Gráfico de tortas 45
Nuevo juego de caracteres 28
Números Romanos 21
Paracaidistas 22
Sprites en C-64 42

EQUIPOS

Galería de Máquinas
Análisis de las principales micros del mercado 25

VARIOS

AviSofti: Bolsa de usados y empleos 6
Concurso Nacional de Programas
Ganadores, Premios y Bases para participar 10
SoftiClub: Club de lectores de **PP** 47
Softiteca: Comentarios de libros de computación y software 6

TK85 de 48 K TK2000 de 128 K. Dos nuevos modelos de Microdigital en el país

ref. 1

La firma **Microdigital** lanzó recientemente al mercado argentino dos nuevos productos.

- una versión de **48 K RAM** de la ya popular **TK85**, tan conocida y apreciada por nuestros lectores (expandible a **64 K**), a un precio muy interesante: ₳ 169.-

Sus principales características técnicas las encontramos en la página central de este número de **PP**

- un nuevo modelo de la **TK2000**, que dispne de **128 K RAM**, que potencia así las cualidades de su antecedente, que compite en precio y prestaciones con otros micros equivalentes, dado que se ofrece a un precio de ₳ 390.-

Ambos equipos son distribuidos y garantizados en nuestro país por la empresa **ARVOC SR**, que ofrece un completo asesoramiento y servicio técnico post-venta

Bajan los precios de las microcomputadoras !!!

ref. 29

Es conocido la tendencia internacional a la baja de los precios de las microcomputadoras y sus accesorios. Esto ha sido provocado por los siguientes factores principales:

- Acelerada obsolescencia tecnológica
- Sobre-expansión anterior de la oferta de equipos y accesorios
- Desarrollo tecnológico de productos más económicos

Alguno de estos factores, combinado con una economía mundial en receso, ha provocado no pocos sofocones en empresas otrora exitosas (**Texas**, **Sinclair**, **Apple**, y otras), que se han visto en figurillas para readaptarse a la situación, producto de la cual también han quedrado innumerable cantidad de firmas de distinto tamaño en todo el mundo (incluso Japón y Corea)

Lo novedoso de todo este proceso es que recién ahora comienza a verse en nuestro país, tan acostumbrado a la suba constante de precios en todos los órdenes, un atisbo importante de baja de precios (principalmente del hardware), como nuestros lectores habrán podido apreciar en enero y febrero. Es aún prematuro definir si la baja es producto de la estacionalidad (son meses de menores ventas, lo que puede ser la causa) o bien si responde a una tendencia con carácter más permanente.

En cualquier caso, en los próximos números de PP brindaremos los resultados de las encuestas e investigaciones que realizaremos sobre el particular, en el entendimiento de ser este tema de sumo interés para nuestros amigos lectores.

LOGO para Talent-MSX próximamente en la Argentina

ref. 30

La empresa **Telemática** está desarrollando un muy completo manual para promover la aplicación del **LOGO** en las micros **Talent-MSX**. Este poderoso lenguaje de programación está especialmente indicado para introducir a los niños en el fascinante mundo de la micro-computación, a través de un método ameno y didáctico, y a su vez es capaz de ser utilizado eficientemente en aplicaciones de inteligencia artificial

Podemos adelantar a nuestros lectores que este interesante trabajo se dividirá en dos partes principales:

1. Charla con las tortugas, para introducirnos en la programación en **Logo**: nos enseñará cómo dirigirnos a las ''tortugas'', a definir programas y a guardar en disco o casete el trabajo realizado, suministrándonos ejemplos de programas en logo para escribir y prácticar.

2. Guía resumida, donde encontraremos información más detallada sobre **MSX-Logo**: colores, figuras (forma) de las tortugas y ''demonios'', funciones de edición de textos y figuras, teclas y caracteres especiales, explicación de los procedimientos de ''primitivas'' y un listado de estos procedimientos organizado por grupos funcionales

En los próximos números de nuestra revista **PP** iremos presentando en exclusiva los distintos capítulos que componen este importante curso sobre **Logo-MSX**, que ha sido muy requerido por nuestros lectores. A esperar, entonces...

16 JAIIO 16a. Jornadas Argentinas de informática organizadas por la SADIO

ref. 34

Entre el 8 y 12 deseptiembre de 1986 se llevarán a cabo en esta Capital Federal las **16 JAIIO-16ª** Jornadas Argentinas de Informática, organizadas por la **Sociedad Argentina de Informáticas e Investigación Operativa** (**SADIO**), tal lo anunciado en conferencia de prensa por el presidente de dicha entidad, **Hugo Pedro Moruzzi** el presidente del comité ejecutivo de las jornadas, **Jorge Chapiro**, **Eloy Martínez** y **Armando Haeberer** integrantes también del comité.

Según los directivos, el objetivo de estas jornadas será el de facilitar un fecundo intercambio tecnológico entre ''los profesionales de nuestras empresas progresistas y los profesores de las universidades de todo el país con reconocidos expertos en informática procedentes de lugares que se encuentran en posiciones de avanzada tecnología''.

Por este motivo, según lo expresado, se ha designado un Comité de Programa compuesto por profesores y profesionales locales junto con académicos del exterior, los que se encuentran plenamente comprometidos con el desarrollo alcanzado en el país en materia de informática.

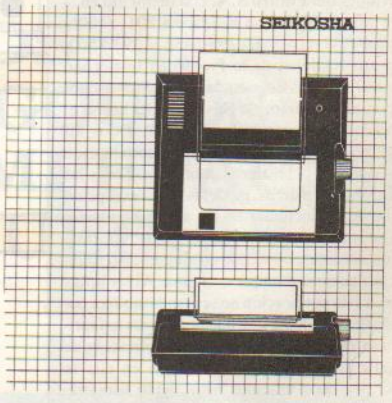
A tal fin han sido invitados, de acuerdo a lo informado, a participar en calidad de conferencistas, científicos de Alemania, Estados Unidos, Francia e Italia.

Además, con carácter de novedad y para facilitar la orientación de los profesionales argentinos y el estableicmiento de vínculos con centros de primer nivel tecnológico, para estas jornadas han sido previstas la realización de reuniones abiertas donde los participantes locales puedan comunicarse con los expertos invitados.

Por otra parte, y de acuerdo a lo sostenido pro los directivos de la **SADIO**, a la brevedad se publicará el ''Llamado para la Presentación de Contribuciones Científicas y Técnicas'' el que contendrá algunos anticipos de las actividades a desarrollar.

Nueva impresora gráfica Seikosha GP-50S

Ref. 35



Ha hecho la aparición en nuestro mercado esta pequeña impresora, que podremos co-

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



PIENSE EN GRANDE LLEGUE A BASIS.

Ref 29

BASIS

Un corazón argentino

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

nectar directamente con la **Spectrum** (CZ2000/**TK90X/TS2068**) y (a través de un sencillo dispositivo adicional) con las hermanas menores de la línea **Sinclair** (**CZ1000/1500** y **TK83/85**)
Seikosh GP-505, precio: ₳ 312

Trabaja por matriz de puntos e imprime 32 columnas por línea a una velocidad de 35 caracteres/segundo. Obviamente no resulta eficiente para aplicaciones comerciales pero tiene varios puntos a favor: no requiere soft adicional, reproduce todos los caracteres de la computadora (incluídos los caracteres gráficos y UDG) y es de un precio accesible sobre la impresora térmica (**Alphacom 32**, por ejemplo) ofrecería la ventaja de una mejor calidad de impresión, y la utilización de rollos de papel común (más económico y fácil de conseguir que el papel térmico)

Próximamente probaremos intensivamente este nuevo accesorio e informaremos los resultados a nuestros amigos de **PP**...

Segundas Jornadas de Informática al Servicio del Derecho a realizarse en Mar del Plata

Ref. 36

El ministro de gobierno de la provincia de Buenos Aires, Juan Antonio Portesi, anunció la realización de las "Segundas Jornadas de Informática al Servicio del Derecho", que se concretarán en la ciudad de Mar del Plata los días 30 y 31 de octubre y 1° de noviembre próximos, organizadas por el Ministerio de Gobierno bonaerense y la Suprema Corte de Justicia de la provincia.

Portesi formuló el anuncio durante una conferencia, donde señaló que "dada la velocidad que tiene el tema de la Informática, hemos dispuesto realizar en la provincia de Buenos Aires Jornadas nacionales e internacionales intercaladas anualmente".

Anunció también que "este año vamos a comenzar a incorporar la red de telemática al registro civil bonaerense y además realizaremos informática en el penal de Batan, para lo cual tenemos donados tres televisores y otros equipos".

Expresó el ministro que "es nuestra intencion lograr que los internados se especialicen en el tema, y y que al dejar la cárcel tengan una salida laboral". Portesi además inauguró, en el edificio de tribunales, el servicio de conexión de informática entre Mar del Plata y el Banco de Datos de la policía bonaerense y servicios penitenciarios.

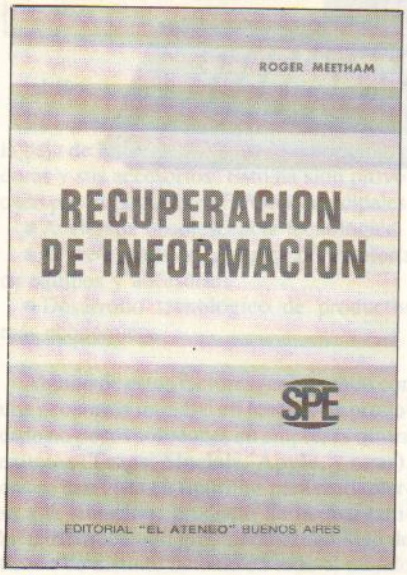
"Esta conexión –sostuvo Portesi– se realizó sin costo alguno por parte del gobierno provincial, usando para ello las instalaciones ya existentes. Con este sistema, hemos afianzado la seguridad del tráfico de la información".

SOFTITECA

Siguen produciéndose muchas novedades en materia de libros en esta plaza. Las distintas editoriales, importadores y librerías se muestran particularmente activos para poner a disposición de todos los interesados las mejores obras que se producen en el país y en todas partes del mundo. Por nuestra parte, para facilitar la elección, continuamos señalando aquellos libros que consideramos más interesantes y útiles para nuestros lectores. Todos sabemos que el estudio es el medio más idóneo para obtener los conocimientos que nos permitirán sacar mejor y a veces insospechado provecho de de nuestra computadora.

Libros editados por **EDITORIAL "EL ATENEO" PEDRO GARCIA S.A.** (Ref. N° 24):

- **RECUPERACION DE INFORMACION** - de **Roger Meetham.**



Esta obra trata en detalle el empleo de la información dentro de las actividades cotidianas. A partir de la naturaleza esencial de la comunicación avanza a través de la escritura y los libros hasta desembocar en el dominio de los centros de computación informativa. Con un específico enfoque orientado hacia las actividades bibliotecarias, analiza en especial el sistema de la recuperación de la información, o sea, el rendimiento de la memoria de una computadora apropiadamente programada. Temas como el almacenamiento de la información, los subsistemas, las clasificaciones lineal y múltiple, los equipos de entrada y salida, la red de datos, etc., resultan accesibles aun para el que no sea especialista en la materia. Precio: ₳ 2,30 (158 páginas).

- **METODOS DE COMPUTACION** - de **Peter Naur.**

Se trata de un manual que se ofrece como "segundo libro del usuario de computadoras", ya que está dirigido al lector ya familiarizado con computadoras y programaciones por lo menos elementales y que utilizan Fortran y Algol 60. La obra incluye un estudio de los conceptos fundamentales acerca de los datos, del procesamiento de datos, de los computadores y de la programación. Expone la metodología para la construcción y verificación de programas; contiene, además, la descripción de diversos métodos de computación de considerable utilidad, los cuales comprenden el análisis de cadenas de caracteres, procesos de selección de acción, técnicas aritméticas, técnicas de búsqueda, de clasificación; se incluye, asimismo el manejo de grandes archivos y el almacenamiento virtual. Precio: ₳ 2,70 (250 páginas).

- **BASES DE DATOS** - de **John K. Lyon.**

Trata sobre la creación de las bases de datos, su mantenimiento, protección y crecimiento, así como las condiciones óptimas en que deben funcionar. Se examinan, además, las cualidades fundamentales de un buen administrador de bases de datos, su ro esencial, funciones y responsabilidades específicas y sus relaciones con otros sectores de la empresa. Incluye un muy práctico glosario. Los conceptos y técnicas expuestos son independientes del hardware, software y del tipo de actividad o empresa en que se emplea la base. Precio: ₳ 3,50 (178 páginas).

- **TEORIA DE LOS SISTEMAS DE INFORMACION** - de **Börje Langefors.**

Esta obra se ocupa de diferentes problemas básicos relativos al análisis y al diseño de sistemas de información. Se tratan en profundidad el análisis de la información, la estructuración de datos, del sistema y la especificación de las propiedades externas de los procesos. La metateoría resultante es, pues, una descripción formal de los símbolos y de las estructuras de lqs sistemas. El autor elige para ellos como lenguaje lo que llama "Algebra de Sistemas", basada en gráficos y matrices. Con este matalenguaje preciso describe luego las bases de datos, los archivos y sus relaciones con el porocesamiento, el diseño de sistemas y la recuperación de información. Precio: ₳ 4,45 (312 páginas).

- **INFORMACION Y DATOS EN LOS SISTEMAS** - de **Börje Langefors y Kjell Samuelson.**

Proporciona los fundamentos básicos imprescindibles para el diseño de sistema de información y analiza las áreas que presuponen mayores problemas para la realizacipón de dicho diseño. Su lectura capacita para conocer y comprender la nueva tecnología y su uso más adecuado; trabajar con datos y sistemas de información y utilizarlos; evitar los errores más comunes en la instalación de sistemas de procesamiento de datos y mejorar la comunicación entre los usuarios y los diseñadores de los sistemas de datos. Precio: ₳ 3,90 (127 páginas).

Libros importados y distribuidos por **DISTRIBUIDORA CUSPIDE S.A.** (Ref. N° 22). Editados por

latindata
MPF-3

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



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

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

latindata MPF-3 TOTALMENTE COMPATIBLE CON apple II

Consulte a nuestros agentes de todo el país.

latindata s.a.

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

Key Advertising

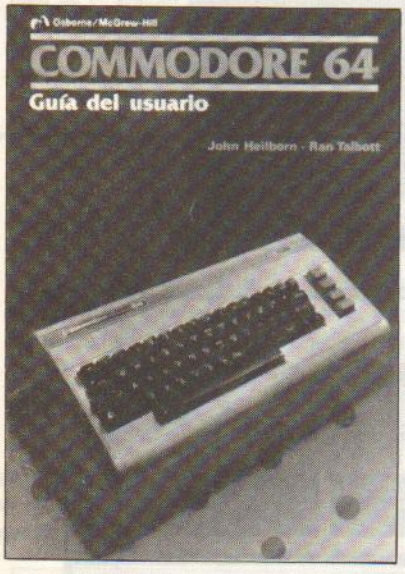
ref. 17

OSBORNE/Mc Graw-Hill, México (México).

● **"MSX". GUIA DEL USUARIO** - de **Paul Hoffman.**

Se trata de un libro recientemente editado que consideramos de gran utilidad tanto para quienes ya disponen de la computadora, o para quienes estén estimando la posibilidad de adquirirla. Contiene informnación detallada de cómo utilizarla, cómo programar en MSX-Basic y cómo emplear el sistema MSX-Dos con una unidad de disco. Está especialmente concebido para que los usuarios principiantes puedan com prenderlo con facilidad, pero contiene también gran cantidad de información útil, para usuarios expertos. Los apéndices señalan las tablas de todas las órdenes, así como de los caracteres que pueden visualizarse en una computadora MSX. Precio: ✠ 24,30 (310 páginas).

● **COMMODORE 64 - GUIA DEL USUARIO** - de **John Heilborn** y **Ran Talbott.**



Esta obra es una guía y fuente de consulta permanente para el usuario. La general aceptación obtenida entre los fanáticos de esta máquina, en todas partes donde fue presentado, dan inequívoca fe de ello. La simple enunciación de su contenido brinda una clara idea de su importancia: Introducción a la computadora, Manejo y Programación avanzada en Basic, Controles para juegos, Gráficos, Sonido, Dispositivos periféricos, Arquitectura del sistema, Utilización de la memoria, Terminales de E/S, Funciones trigonométricas, Caracteres y códigos, Mensajes de error, Sentencias y Funciones Basic. Precio: ✠ 25,20 (464 páginas).

Editado por **FERRE MORET S.A.**, Barcelona (España):

● **TODO SOBRE EL NUEVO COMMODORE 128** - de **Klaus Gerits** y **Frank Kampow.**

Este libro de primicias del **Commodore 128** no ofrece solamente un resumen completo de todas las caracteristicas y rendimiento, lo cual significa una importante ayuda para su adquisición. Muestra, además, todas las posibilidades del nuevo equipo en función de sus tres modos de operación. Entre otros, se describen el hardware, los modos de operación: modo 64, modo 128 y modo CP/M, las configuraciones de memoria, la disposición de la página cero, trabajos con dos pantallas, modo de 80 caracteres, Basic V7.0, comandos de gráficos y sonido, comandados de control, periféricos rápidos (1571), etc. Precio: ✠ 23,10 (276 páginas).

Libros importados y distribuidos por **DISTRIBUIDORA YENNY S.R.L.** (Ref. Nº 4). Editados por **EDITORIAL PARANINFO S.A.**, Madrid (España).

● **LENGUAJE FORTH PARA MICROS** - de **Steve Oakey.**

El forth es un lenguaje relativamente nuevo, que está teniendo una gran acogida por parte de los usuarios de los microordenadores: es más fácil de usar y más rápido que el Basic. Con la ayuda de los test de autoevaluación, todo lector podrá aprender a programar en Forth y a comprender el uso y las ventajas del lenguaje, que está disponible para una gama muy amplia de micros y que cada vez tiene mayor popularidad. Precio: ✠ 13,56 (176 páginas).

● **APPLE II Y APPLE II/e** - de **Jean-Yves Astier.**

En ella se expone en detalle el funcionamiento del sistema gráfico, que es uno de los mejores logros de la Apple II. Sirve para aprender a hacer coexistir y comunicarse programas Basic y Ensamblador. Los principales subprogramas del Monitor y del Apllesoft se explican detalladamente y permiten programar eficazmente en Ensamblador. Precio: ✠ 6,78 (113 páginas).

● **BASIC DEL APPLE II LY APPLE II/e** - de **Jean-Yves Astier.**

Está dedicada al Basic del App'le II, indicando el funcionamiento de todos sus comandos, y acompañando el texto con numerosos ejemplos. También se dan, en apéndices, programas completos para ilustrar las posibilidades del Basic del Apple II y Apple II/e y otras ideas muy interesantes. Precio: ✠ 6,78 (107 páginas).

Editado por **EDICIONES GENERALES ANAYA**, de madrid (España):

● **MI PRIMERA BIBLIOTECA INFORMATICA** - de **Gaby Waters.**



He aquí una nueva serie, divertida y llena de colorido, para que los más pequeños se inicien en el mundo apasionante de las computadoras, de la mano de una banda de fantasmas y mosntruos muy simpáticos que van mostrando lo que pueden hacer los ordenadores, cómo funcionan y cómo programarlos. Los libros contienen programas elementales, cortos y graciosos que se pueden utilizar en cualquier máquina. Los títulos que forman esta colección son: **"Todo sobre ordenadores"**, que es la amenísima introducción que explica en términos sencillos qué es un rodenador, cómo funciona y todo lo qué puede hacer. **"Basic Fácil"**, un libro divertido y sencillo para que cualquiera aprenda con toda facilidad a programar en Basic, ayudándose de los abundantes programas para realizar, y de los rompecabezas para resolver qué incluye en sus páginas, profusamente ilustradas. **"El Ordenador Divertido"**, que enseña que un ordenador puede utilizarse como si fuera un jueguete, pero, en tal caso, resulta tan apasionante que es difícil dejarlo. En este libro se demuestra que, sin necesidad de conocimientos previos, se puede usar la computadora como un fascinante baúl de juegos. Precio: ✠ 4,90 (48 páginas).

Por último, en materia de libros, estamos en condiciones de anticipar que, en los próximos días (principio de marzo) aparecerá, editado por **JAVIER VERGARA EDITOR S.A.** (Ref. Nº 25), el esperado libro ● **EL COMPUTADOR DE LOS NEGOCIOS** - de **Peter A. Mc Williams,** que, juntamente con **El procesador de Palabras** y **El libro del Computador Personal,** forma una trilogía de ste mismo autor que, empleando el estilo desenfadado y certero que lo caracteriza, nos informa ahora acerca de qué son los computadoresp ersonales, qué pueden hacer por una empresa y cómo comprar uno. Contiene ejemplos de uso, respuesta a las dudas más comunes y hasta un capítulo sobre el miedo a las computadoras. Saldrá a la venta con un precio de ✠ 5 (296 páginas).

En esta sección de **SOFTITECA** también quermos que tengan presente la existencia en el mercado de numeroso y muy calificado software que, en forma de casetes o disketes, se encuentran al alcance de todo entusiasta de la computación. Los mismos cubren una amplísima gama de aplicaciones utilitarias o sirven para jugar, desarrollando la habilidad del usuario y agudizando su destreza. Pedimos, una vez más, a editores, importadores, distribuidores y, en general, empresas o profesionales productores de software, que por intermedio de estas columnas, pongan en conocimiento de nuestros lectores sus productos, en especial cuando se trate de novedades.

En esa ocasión, vamos a hacer una rápida enunciación de CASETES de buen nivel técnico que se encuentran a la venta en casa de computación, en disquerías y **también por nuestro intermedio** (talón o Softi Club):

PROGRAMAS "CZ" (1)

Para el Hogar

● **Iniciación - CZ 1000 TK83** - 2K memoria.

Contiene instrucciones básicas para operar el teclado y cuatro programas de distintos rubros: Matemática, Recetas, Adivinanzas y Gráficos. Precio: ✠ 3,54.

● **Gastos - CZ 1500 TK85** - 16K memoria.

Permite llevar el presupuesto de su hogar. Precio: ✠ 3,24.

Educativos

● **Simulador de Vuelo - CZ 2000** - 48K memoria.

Integra además el rubro juegos. Se complementa con el programa **Piloto de Caza.** Pilotee un avión. Practique despegue, vuelo y aterrizaje. Precio: ✠ 4,33.

● **Gráficos - CZ 2000** - 48K memoria.

Permite dibujar en tres dimensiones, en base a sus cortes transversales, dando color y movimiento en el espacio. Puede sombrear, rotar, acercar o alejar el cuerpo creado. Ideal para estudiantes de arquitectura. Precio: ✠ 4,33.

● **Provincias y Capitales (Arg.) - CZ 1500 TK85** -

16K memoria.

Test sobre geografía argentina, se complementa con geografía universal. También para **CZ 2000.** Precio: ₳ 3,54.

- **Cuenta Bancaria - CZ 1500 TK 85** - 16K memoria.

Permite tener al día la chequera personal, de la oficina o del comercio (hasta 250 operaciones de depósito emisiones.

- **Estadística - CZ 1000 TK83** - 2K memoria.

Contiene 3 programas para calcular valores de desviación estandar, media, ordenada al origen y pendiente. Precio: ₳ 3,24.

(1) Producidos por Czerwen y Electrónica S.A. (*).
(*) La mayoría de los programas desarrollados por Sepctrum son compatibles con la TX 90X y muchas de llas (las programadas con Basic) también con la TS2068. Conviene probar antes de comprar.

- **Archivo - CZ 2000** - 16K memoria.

Permite guardar hasta 500 fichas en base a un modelo dado. Poderosos comandos le permiten clasificar, modificar, y seleccionar los datos archivados. Incluye "**Gaceta**", un ejemplo de un catálogo de países. Precio: ₳ 3,73.

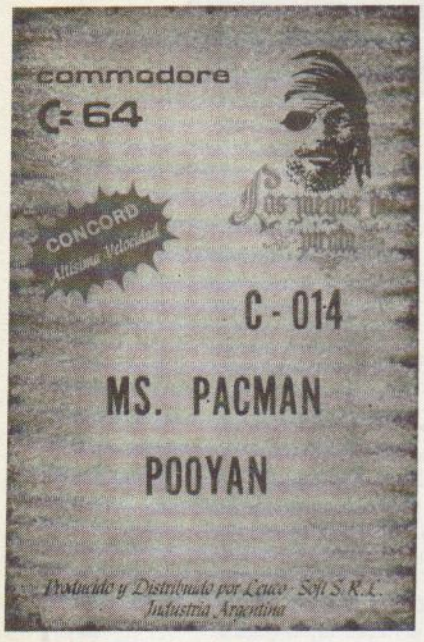
Juegos

- **Piojos Locos - CZ 2000** - 48K memoria.

El objetivo es recorrer un laberinto, recogiendo los huevos antes de que se lo coman los piojos, que lo perseguirán. Precio: ₳ 4,33.

- **Galaxians - CZ 2000** - 16K memoria.

El objetivo es destruir a los invasores. Dispone de tres naves y pueden jugar dos personas. Precio: ₳ 3,73.



- **Ms. Pacman** - Conduzca a un simpático comecocos por un intrincado laberinto, intentado comer todos los puntos que encuentre en su camino; pero tendrá que cuidarse de los terribles fantasmas que intentan apresarlo.
- **Pooyan** (en el reverso) - El objetivo es impedir que los malvados lobos se coman a sus ovejas. Precio: ₳ 4,50.
- **Jumping Jack 2** - Permita que la rana Jack peuda cruzar la transitada avenida, luego un río, buscando un lugar seguro donde esconderse.
- **Cuddly Cuburt** (en el reverso) - Conduzca al simpático perrito a través de todos los escalones de la pirámide donde se encuentra. Evite todos los obstáculos que se le crucen. Precio: ₳ 4,50.

Para Spectrum 48K - TK 90X-48K y CZ 2000

- **AVALON** - Se encuentra en una habitación tridimensional y lo primero que debe hacer es recorrer el primer nivel, recogiendo objetos mágicos. El objetivo es desltruir al Amo del Caos, que está en el octavo nivel de Avalon. Juego bastante complejo.
- **Alchemist** (en el reverso) - El objetivo inicial es encontrar las cuatro piezas del conjuro de la destrucción, luego hay que buscar al amo del castillo y, utilizando el conjuro, se lo podrá destruir. Precio: ₳ 4,50.
- **Stop the Express** - El expreso Sur está en movimiento. La misión es detenerlo a toda costa. Para ello, hay que llegar a la locomotora, pero para su desgracia alguien se equivocó y lo subieron en el último vagón del tren.
- **Taladro** (en el reverso) - Ha conseguido trabajo como chofer del taladro del reino y por lo tanto tiene una misión para cumplir. Debe limpiar los subsuelos del palacio de bichos y mamuts, mediante el empleo de un rayo congelante que mata al bicho y congela momentáneamente al mamut. Precio: ₳ 4,50.

Les recordamos que todos los libros y casetes comentados en esta revista y en las anteriores pueden conseguirse en las buenas librerías, negocios especializados en computación y... ¡A NO OLVIDARLO! por nuestro intermedio con toda facilidad, con el sencillo y muy práctico sistema de pedidos mediante el uso del talón que a ese efecto parece en la revista, o personalmetne en el Softi Club (Maipú 464, piso 8, Oficina 801, 1006-Capital).



Libros de computación

Los best-sellers del trimestre

Todo sobre el nuevo Commodore 128,
276 páginas, por Klaus Gerits,
(Ed. Ferré Moret, 1985) ..... ₳ 23,10

Commodore 64. Guía del Usuario,
464 páginas, por John Heilborn,
(Ed. McGraw-Hill, 1985) ..... ₳ 25,20

MSX. Guía del Usuario,
310 páginas, por Paul Hoffman,
(Ed. McGraw-Hill, 1985) ..... ₳ 24,30

El Lenguaje de Programación C,
248 páginas, por Brian Kernighan,
(Ed. Prentice-Hall, 1985) ..... ₳ 17,90

Commodore 128. Consejos y Trucos,
328 páginas, por H. Horning,
(Ed. Ferré Moret, 1985) ..... ₳ 29,40

nº 33

CUSPIDE computación/libros

Suipacha 1045, Tel. 313-0486/9362, 1008 - Buenos Aires.

TU VOTO DECIDE Y PARTICIPA EN EL SORTEO;
"Cubo Mágico" de **Héctor R. Molena** ganó una **CZ2000 (Spectrum) de 48 K** donada por **Czerweny Electrónica SA** por resultar el programa más votado al 15/2/86.
El 4/4/86 cierra la siguiente selección de los programas que ganarán también una **CZ2000 (Spectrum) de 48 K** y otros importantes premios...

PARTICIPA Y GANA !!!...

CONCURSO NACIONAL DE PROGRAMAS

MEJORES PROGRAMAS PUBLICADOS EN P.P. NUMEROS 1 AL 9
Resultados según constancia en acta labrada ante Escribano Público

1er. Premio. **CZ2000 (Spectrum)**	*Héctor R. Molena* San Martín, PBA	**Cubo Mágico** T/S2068
2do. Premio un **lápiz óptico DEC**	**Ing Raúl D Vera** Sgo del Estero	Costos de Fabricación TK85
un **Curso de Basic** en **EPI**	**Daniel Weselha** Ezeiza PBA **Marcelo D. Tropia** Olavarria, PBA	Organo y efectos CZ1000 + EM Air Battle CZ 1500

GANADORES DEL CONCURSO "VOTO LECTOR"
Entre todos los votos recibidos al 15/2/86, han resultado favorecidos por sorteo los siguientes lectores:

Ganaron un Curso de Basic en **EPI**	**Jorge Catena** Pto. Madryn (Cbt) **Karina G. Sani** Haedo PBA	**Ariel Edgardo Arias** Salta (Salta) **Carlos Alberto Quiroga** Capital
Ganaron un Casete de los **Juegos del Pirata**	**Santiago Amsé** Santa Rosa (LP) **Sergio Villalón** San Rafael (Mza)	**Jorge Daniel Comerce** Bahía Blanca (PBA **Francisco Montero** Tucumán (Tucumán)

NUEVOS PROGRAMAS GANADORES PUBLICADOS EN P.P. NUMERO 10

PREMIO	GANADOR	PROGRAMA	PREMIO	GANADOR	PROGRAMA
₳ 25 en efvo.	**Andrés Bitenc** Ituzaingó PBA	Números Romanos CZ 1500	₳ 25 en efvo.	**Diego Rodríguez** Longchamps PBA	Nuevo juego de caracteres T/S 2068
₳ 25 en efvo.	**Pablo D. Ferrante** Capital Federal	Paracaidistas TI-99/4A + EB	₳ 25 en efvo.	**Christian Tottino** La Plata PBA	Bola Loca CZ 2000
₳ 25 en efvo.	**Hugo R. Ortiz** Mendoza	Golf T/S 1000 + EM	₳ 25 en efvo.	**Daniel Weselka** Ezeiza PBA	Grabador Digital T/S 1000 + EM



Premios

- **₳ 25 (veinticinco australes) en efectivo** por cada programa publicado en P.P.
- Premio a los mejores programas todos los meses. Para el próximo mes 1er. premio: Una computadora **CZ2000 (SPECTRUM) de 48 K.** donada por Czerweny Electrónica S.A. para ser entregada al autor del mejor programa de los publicados en los últimos 12 (doce) números de nuestra revista **PROGRAMACION POPULAR**. La selección del mejor programa del mes se hará con la participación de los **lectores** (ver **Bases** y **Talon** de voto adjunto) quienes al votar intervendrán en un sorteo especial de cada mes (cursos de computación y otros importantes premios)

Todos los meses: más premios sorpresa.

- Participación en el concurso semestral anual, con importantes premios para todos los programas del ranking.
- Inscripción del autor en la lista de **Colaboradores** de tu revista **Programación Popular** y en el **Cuadro de honor**.
- Si el programa fuera "comercializable" su autor percibirá un mínimo del 30% de los beneficios (regalías) por derecho de autor.

Bases

- Programas **INEDITOS** en cassette o diskete (se admite más de un por autor). El cassette/diskete se devolverá al autor a partir de los 30 días de recibido en la Editorial.
- Máquinas: **COMMODORE 16/64/128, CZ1000/1500/SPECTRUM, Talent MSX, TI99/4A, TK83/85/90X 2000 y TS2068** (lista de microcomputadores más corrientes en el país, no excluyente).
- Acompañar, además del cassette o diskete, listado del programa (emitido por impresora, a máquina o manuscrito con letra de imprenta) y texto explicativo, todo FIRMADO POR SU AUTOR. Es aconsejable (no imprescindible) remitir también la estructura del programa o diagrama de lógica y la tabla o lista de las variables. También deben acompañarse los datos personales, domicilio completo, edad y computadora en la que fue realizado el programa y grabado el soporte magnético enviado (cassette o diskette) como asimismo los K de memoria de la máquina.
- Remitir material por correo certificado a:

Ediciones del Futuro S.A. Revista **Programación Popular**
Softi Club
Maipú 464, 8° piso of. 801 (1006) Capital Federal

- **Jurado**: La selección la realizará personal técnico altamente calificado de la Editorial, cuyo fallo será inapelable.

Asimismo para elegir el mejor Programa del mes, cuyo autor ganará una **CZ2000 (Spectrum) de 48 K**, el jurado lo formarán: **un voto** de los **lectores** de nuestra revista **P.P., un voto de Czerweny Electrónica S.A. y un voto de Programación Popular**. Resultará ganador el programa que tenga mayoría de votos. Si los 3 votos difieren, ganará el seleccionado por los lectores. Se labrará un acta ante **Escribano Público** de los resultados.

Para el próximo mes, la selección se realizará el día 4 de abril de 1986, informándose al ganador en nuestro próximo ejemplar de **P.P.** (N° 10).

- El programa ganador de un primer premio no seguirá participando de los concursos siguientes. Los votos que reciba erróneamente no serán computados.
- Los autores de programas ganadores ceden expresamente a **Ediciones del Futuro S.A.** los derechos de exclusividad del uso del programa y el restante material, a título gratuito, con fines de divulgación del conocimiento en todo tipo de medios de difusión (gráfico, radial, televisivo, casetes, etc.). Si la Editorial hiciera un uso comercial del programa, abonará un mínimo del 30% de los beneficios resultantes (regalías) por derechos de autor.
- En lo aquí no indicado, regirán las restantes bases publicadas en el N°5 de P.P. (no para los premios).

CURSO BASIC VII

Programación Estructurada, REM, READ, DATA, RESTORE

RESPUESTAS AL TEST DE LA LECCION VI

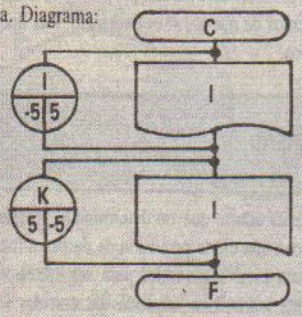
1. Verdadero o Falso:

a. Verdadero.
b, Falso.
c. Falso.

En caso de no responder correctamente este punto, repasar el Capítulo 6, punto 1.

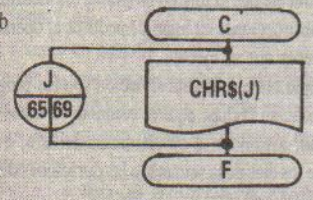
2. Diagramar y programar:

a. Diagrama:



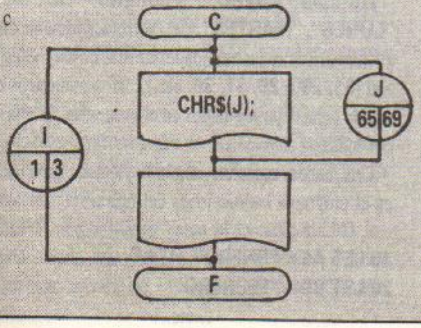

```
10 FOR I = -5 TO 5
20 PRINT I
30 NEXT I
40 FOR K = 5 TO -5 STEP -1
50 PRINT K
60 NEXT K
```

b



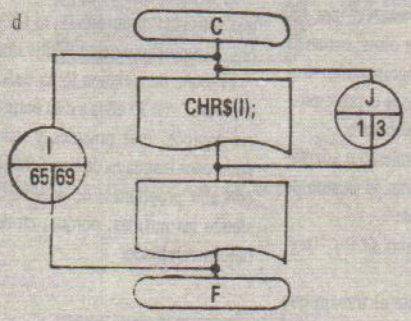

```
10 FOR J = 65 TO 69
20 PRINT CHR$(J)
30 NEXT J
```

c



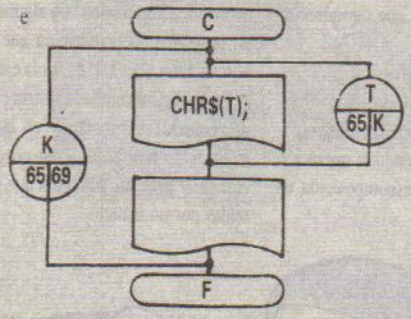

```
10 FOR I = 1 TO 3
20 FOR J = 65 TO 69
30 PRINT CHR$(J);
40 NEXT J
50 PRINT
60 NEXT I
```

d




```
10 FOR I = 65 TO 69
20 FOR J = 1 TO 3
30 PRINT CHR$(I);
40 NEXT J
50 PRINT
60 NEXT I
```

e




```
10 FOR K = 65 TO 69
20 FOR T = 65 TO K
30 PRINT CHR$(T);
40 NEXT T
50 PRINT
60 NEXT K
```

3. Las fallas son:

a. Falta la línea **NEXT A.**
b. Falta completar la línea **10** con **TO 10.**
c. Falta la sentencia **FOR** en la línea **10.**
d. La variable de control es **T,** y se imprime **A.**
e. La línea **30** cierra un lazo cuya variable de control es **T** y no **A,** como se abrió en la línea **10.**

En caso de contestar mal alguna de las preguntas del punto 3. repasar 6.2.2.

En caso de error en el punto 2.a. revisar 6.2
En caso de error en el punto 2.b. revisar 6.5
En caso de error en el punto 2.c. revisar 6.4 y 6.5
En caso de error en el punto 2.d. revisar 6.4 y 6.5
En caso de error en el punto 2.e. revisar 6.4 y 6.5

7.1 La programación estructurada

Habíamos hablado anteriormente sobre el concepto de **programación estructurada (PE).**

PE es un *conjunto de técnicas y principios de programación*, destinado no tanto a facilitar la práctica de la programación (por el contrario, PE exige tanta o más comprensión del asunto por parte del programador que si éste emplea sus técnicas habituales), sino más bien *a aumentar la funcionalidad de un programa y a asegurar su supervivencia*, facilitando su *mantenimiento*.

La **PE** persigue objetivos que son *económicos* en más de un sentido.

Por un lado, busca la *optimización del empleo de los recursos técnicos* (tiempo de programación, de ejecución y de memoria utilizada por un programa) y, por otro, tiende a promover el *ahorro en esfuerzo de interpretación que deban hacer eventuales usuarios* que vayan a utilizar o modificar de alguna manera el programa.

Esto asegura que el trabajo hecho una vez por un programador pueda ser aprovechado por otros, evitando que vuelvan a cometerse errores de programación –frecuentes cuando se ataca un problema desde el principio– y ahorrando tiempo de desarrollo del software.

Para facilitar ese trabajo de mantenimiento o adaptación de los programas conviene observar, al tiempo de escribirlos, algunos **principios básicos:**

1.- Delimitar perfectamente cada uno de los "segmentos" del programa.

Es decir, "dividir" conceptualmente el trabajo que debe hacer el programa en tareas independientes entre sí y escribir un segmento de programa que cumpla con cada una de ellas.

2.- Aplicar el refinamiento progresivo.

(También llamado Método de Aproximación *"top-down"*, de arriba hacia abajo), reduciendo cada problema a sus partes componentes y repitiendo la operación con esas partes, hasta obtener una descomposición en subproblemas triviales.

3.- "Estructurar" el programa resultante.

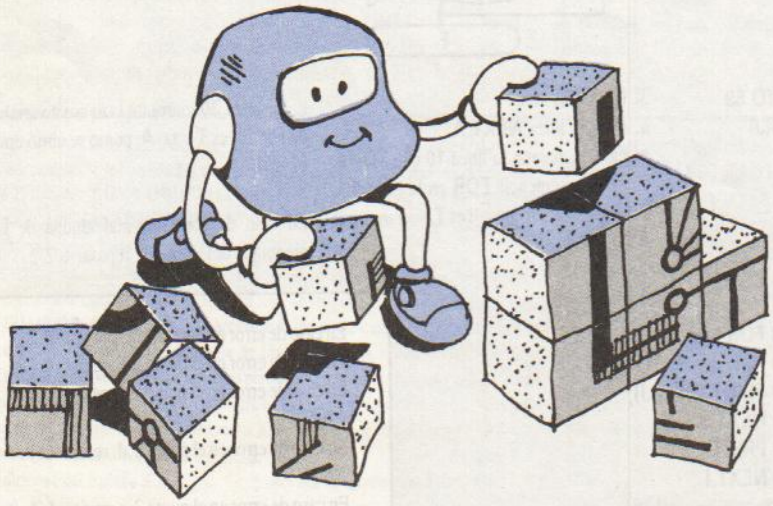
O sea, escribir los segmentos –y el conjunto de ellos– en términos de las tres estructuras fundamentales descriptas en el capítulo anterior: *composición secuencial, decisión, iteración.*

4.- En lo posible, el programa debe ser de curso secuencial.

O sea, no conviene complicar el recorrido del control con **GOTO**s evitables. A veces, un reordenamiento de los bloques en que se ha escrito un programa permite eliminar algunas de estas instrucciones.

5.- El programa debe ser lo más corto posible.

Esto favorece el que se ciña a ejecutar una tarea específica en forma estricta, constituyendo un módulo que luego pueda agregarse a otros para construir un nuevo programa.



El método de aproximación "top-down" es aplicable a muchos órdenes de cosas, y siempre permite generar estructuras comprensibles a partir del caos.

7.2 Los ayudamemorias REM.

El tener en cuenta el hecho de que la informática es una actividad desarrollada, al fin y al cabo, por humanos, es una perspectiva relativamente reciente.

El toque de atención lo ha dado la actual "crisis del software": de 35 años a esta parte disponemos de mejores equipos, más potentes y veloces, mejores y más poderosos lenguajes de computación, pero las formas de concebir programas, de desarrollar algoritmos, siguen siendo las mismas, porque, en definitiva, nosotros no hemos cambiado.

Sin embargo, todos los lenguajes de programación evolucionados cuentan con alguna manera de incluir en un programa documentación que sea de exclusivo uso humano, sin relevancia para la ejecución del programa.

En BASIC tenemos la instrucción **REM**, abreviatura de **remark** ("comentario" en inglés), que *permite introducir en el listado de un programa anotaciones de cualquier tipo*, tan sólo para que estén allí y puedan ser leídas por un usuario.

Un sencillo ejemplo mostrará su uso:

```
10 REM EJEMPLO DE USO DE LA
20 REM SENTENCIA REM
30 REM ESTE PROGRAMA NO ES MAS
40 REM QUE UN SALUDO
50 REM A LOS LECTORES DE P.P.
60 PRINT "HOLA"
70 REM ESO ERA TODO

RUN

 HOLA
```

El efecto que tiene **REM** al momento de ejecución es sencillamente nulo: *las líneas de programa que comienzan con la palabra* **REM** *son simplemente ignoradas* (su contenido no se imprime en pantalla) y se pasa a la línea siguiente. Conviene acostumbrarse a usarlas como ayudamemoria, con algunas salvedades:

- Ocupan lugar en la memoria de la computadora.
- Disminuyen (aunque mínimamente) la velocidad de ejecución del programa.

7.2.1. Un uso menos evidente de **REM:** en la *depuración de programas*. Cuando se desea saber *qué ocurriría si la línea* **n** *no estuviera*, porque quizá sea la causa del problema, podemos borrarla y ejecutar el programa, pero así la perderemos para siempre; *corrijamos, en cambio, esa línea insertando una instrucción* **REM** *inmediatamente después del número de línea* **n.** Todo lo que sigue en esa línea será pasado por alto, y borrando el **REM** tendremos de nuevo, si es necesario, a la línea **n** entre nosotros.

7.3 DATA, READ

Muchas veces ocurre que un determinado programa debe contar con una cierta provisión de datos en forma fija, que deben estar disponibles para ser usados –no necesariamente todos ellos– en todas las corridas.

Es conveniente que el mismo programa los incluya en su lista de instrucciones en lugar de pedirlos al operador todas las veces que se ejecute.

Un ejemplo clásico de esta situación son los programas que manejan fechas; aquí el conjunto de datos es, por ejemplo, los nombres de los meses del año, o los nombres de los días de la semana, o las duraciones de los meses, en días.

Los datos **"ENERO", "FEBRERO",** etc., o bien **"LUNES", "MARTES",** etc., son efectivamente datos alfanuméricos, y las cantidades de días de los meses del año (**31, 28** o **29, 31, 30,** etc.), datos numéricos que vale la pena dejar asentados en el programa, de una vez por todas.

Una manera accesible, según lo ya visto, sería incluir en el programa instrucciones del tipo:

```
10 LET A$ = "ENERO"
20 LET B$ = "FEBRERO"
                    ...etc...
```

...o bien

```
10 LET A$ = "LUNES"
20 LET B$ = "MARTES"
```

...etc...

El inconveniente de esto es que este conjunto de instrucciones ocupa mucha memoria en la computadora, y mantiene ocupadas muchas variables con datos que quizá no lleguen a utilizarse.

Una alternativa elegante sería tomar ese conjunto de datos e indicarle al programa:

"Esta es una lista de datos. Guárdense en una línea de programa y úsense, uno a la vez, cuando sea necesario."

Toda esta oración se resume en la palabra BASIC:

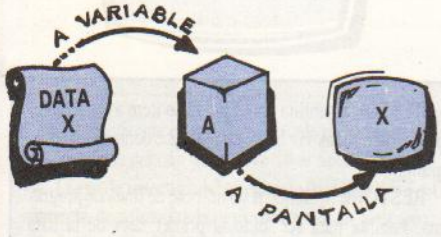
DATA.

De modo que nuestra lista de datos quedará convenientemente almacenada si tipeamos:

```
10 DATA ENERO, FEBRERO, MARZO, ABRIL, MAYO, JUNIO, JULIO, AGOSTO, SEPTIEMBRE, OCTUBRE, NOVIEMBRE, DICIEMBRE
```

...Separando los datos entre comas; a pesar de que son datos alfanuméricos, no siempre se necesita ponerlos entre comillas (depende de la computadora); en algunas computadoras ni siquiera existe la sentencia **DATA**.

En el momento en que necesitemos usar esta información –por ejemplo, para imprimirla en pantalla– deberemos tomarla delicadamente de esta lista **DATA** y depositarla en una variable. Una vez allí el dato, podemos hacer con él lo que deseemos.



De modo que, si el dato **"ENERO"** ha sido recogido en la variable **A$** (respetar el tipo de variable en la asignación: ver **2.8**), bastará un

PRINT A$

para ponerlo en pantalla.

Lo que no está claro hasta aquí es el pasaje del dato, desde la lista **DATA** hasta la variable en cuestión.

Para efectuar este pasaje necesitamos decirle a la computadora:

"Por favor, fíjese en la lista **DATA.** *Encontrará allí unos cuantos datos. ¿Sería tan amable de tomar uno de ellos y colocarlo aquí en esta variable* **A$**"

Ceremoniosa oración que en BASIC suena casi prepotente:

READ A$

Es oportuno recordar que en **4.2** usamos una de esas tontas frases, como la de más arriba, para describir la instrucción **INPUT**.

(**INPUT** iba a buscar datos al teclado; **READ** los buscará en una lista **DATA**. Ambos necesitan que les indiquemos la variable "destino" de la información solicitada.)

En este ejemplo, entonces:

```
10 DATA ALFA, BETA, GAMMA
20 READ A$
30 PRINT A$
```

¿Qué dato leerá **READ**? Simplemente el primero de la lista **DATA**. Un comando **RUN** mostrará en pantalla

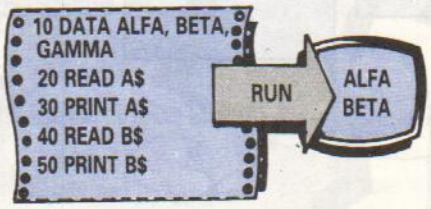
```
ALFA
```

7.4 Más precisiones...

...y el resto de los datos?

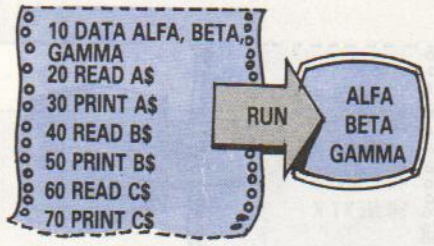
...Habrá que usar una instrucción **READ** por cada uno a rescatar. Ejemplos:

1)



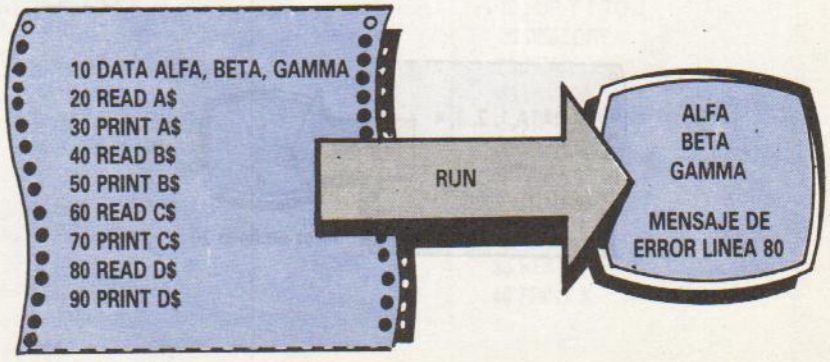
Obsérvese que el dato leído por el **READ** de línea **40** es el siguiente al último leído.

2)



Aquí, al ejecutarse la línea **60**, los datos **ALFA** y **BETA** ya fueron leídos. **READ C$** coloca en **C$** el dato siguiente:

3)



Aquí el problema consiste en intentar leer cuatro datos –a cargar en **A$, B$, C$** y **D$**– donde sólo hay tres.

Moraleja: cada dato que **READ** *lee es –por decirlo así– "tachado", para que no vuelva a ser leído.* **READ** *siempre lee el primer dato disponible en la lista* **DATA**.

Notas: 1) Los datos manejados hasta aquí por estos sencillos programas de ejemplo, han sido siempre alfanuméricos, pero lo dicho se aplica idénticamente a datos numéricos, siempre que respetemos el tipo de la variable que vamos a asignar.

2) Cuando los datos a colocar en una lista **DATA** son alfanuméricos, algunas computadoras exigen que se coloquen entre comillas. Siempre deben separarse por comas, y no colocar coma al final del último dato de la lista.

3) Puede colocarse más de una sentencia **DATA** en un programa: agotada la primera lista, el programa pasará automáticamente a leer la siguiente.

Según esto, es lo mismo

```
10 DATA UNO, DOS, TRES
```

que

```
10 DATA UNO
20 DATA DOS
30 DATA TRES
```

o que

```
10 DATA UNO
   20 DATA DOS, TRES
```

4) La o las listas **DATA** pueden ubicarse en cualquier lugar del listado del programa. **READ** irá allí a buscar los datos, estén donde estén.

7.5 RESTORE

Es muy común ver el empleo de **READ/DATA** acompañado de **FOR/NEXT** (recordar capítulo 6):

```
10 DATA 1, 2, 3, 4
20 FOR K = 1 TO 4
30 READ F
40 PRINT F
50 NEXT K
```

RUN

```
1
2
3
4
```

La implementación de un lazo permite la lectura de todos los elementos en la lista en forma secuencial. Por otro lado, como **FOR/NEXT** repite un conjunto de líneas una cantidad determinada de veces, no existe riesgo de "quedarse sin datos que leer", interrumpiendo así el curso del programa, como ocurriría en el caso siguiente:

```
10 DATA 1, 2, 3, 4
20 READ F
30 PRINT F
40 GOTO 20
```

RUN

```
1
2
3
4
```

error en línea 20

Este programa debe imprimir una frase muy curiosa:

```
10 DATA DONDE, TODO, SIRVE, PARA,
NADA
20 READ A$
30 READ B$
40 READ C$
50 READ D$
60 READ E$
70 PRINT A$; B$; C$; D$; B$; E$; C$; D$; E$
```

(...descúbrala el lector diligente.)

Supongamos querer repetir la impresión de la frase continuamente: agreguemos

```
80 GOTO 10
```

Que nos devuelve el control al principio del programa.

El problema es que la línea **10** no es ejecutable; es decir, por más que el control pase por la línea **10** una y otra vez, los datos "tachados" por una primera lectura de **READ** no se rehabilitan. El **READ** de línea **20** detendrá el programa con un mensaje de error.

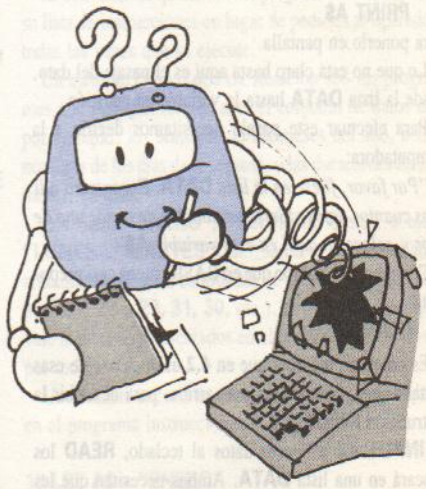
Lo que queremos es que ese **READ** pueda disponer del dato alfanumérico **"DONDE"** cada vez que devolvamos el control al principio del programa.

Lo conseguiremos *"borrando"* las tachaduras con una instrucción:

```
75 RESTORE
```

RESTORE habilita como próximo ítem a ser leído el primero de todas las listas **DATA** que contenga el programa.

RESTORE n, siendo **n** un número de línea de programa, habilita para ser leído al primer ítem de la lista **DATA** que se halle en la línea **n**.



Un programa con sus propios datos por dos preguntones.

Un ejemplo: *construir un programa que tome un dato numérico representando el número de orden en la semana de un día determinado, y responda con el nombre de ese día.*

```
10 DATA LUNES, MARTES, MIERCOLES,
JUEVES, VIERNES, SABADO, DOMINGO
20 INPUT DIA
30 FOR L = 1 TO DIA
40 READ Q$
50 NEXT L
60 PRINT Q$
```

Este programa toma del operador (línea **20**) un número –digamos **3**– y lo conserva en la variable **DIA**.

A continuación repite **DIA** veces –tres, en este caso– la lectura de un dato alfanumérico de la lista **DATA**, depositándolo en **Q$**. Cada dato borra el anterior contenido de **Q$**.

Al concluir las **DIA** vueltas, el contenido de **Q$** es el tercer (en general, el **DIA** ésimo) dato de la lista. Recién entonces se lo imprime.



Si queremos que esta tarea se efectúe en forma continua, no alcanzará con un **GOTO 10** al final del programa. Luego de la lectura del tercer dato **MIERCOLES**, el siguiente dato habilitado será **JUEVES**. Si deseamos saber el nombre del día de la semana número tres, el programa nos responderá



Y si pedimos el nombre del quinto día de la semana, ocurrirá un error en la línea **40** que interrumpirá el programa.

Nuevamente la solución es tener, al volver al principio, otra vez la lista habilitada en el primero de los datos. La respuesta es, una vez más,

```
70 RESTORE
80 GOTO 10
```

El siguiente **TEST** será divertido...





Las nuevas generaciones de hardware se suceden en oleadas, mientras las técnicas de desarrollo de software no varían fundamentalmente, produciéndose un "retraso" relativo de este último. A esto se ha dado en llamarlo "crisis del software".

TEST DE LA LECCION VII

Indicar lo que imprimirá la computadora en cada caso.

a)
```
10 DATA 8, 56, 7
20 READ R,T
30 PRINT T
40 STOP
```

b)
```
10 FOR X = 1 TO 5
20 READ XX
30 PRINT XX
40 NEXT X
```

c)
```
10 DATA 56, 84, 49
20 DATA 48, 77, 53
30 DATA 19, 16, 51
40 RESTORE 30
50 READ X
60 RESTORE
70 READ R
80 PRINT R
```

d)
```
10 FOR T = 1 TO 10
20 DATA 56, 48, 2
30 RESTORE
40 PRINT T
50 NEXT T
```

e)
```
10 FOR T 1 TO 6
20 RESTORE
30 READ F
40 PRINT F
50 NEXT T
60 DATA 28, 87, 33, 23, 11, 91, 4
```

f)
```
10 DATA 5, 43
20 READ X
30 RESTORE
40 PRINT X
```

Grabador Digital



Este es otro insólito programa de **Daniel Weselka**, el autor de **"Organo y efectos"**, donde había conseguido (P.P. Nº 4) sonorizar estas computadoras, cosa que nadie se esperaba...

Ahora **Daniel** nos vuelve a sorprender mostrándonos cómo digitalizar la información constituida por cualquier clase de sonidos audibles... y de los otros.

Nos dice **Daniel**:

"Este programa es para los que gusten de experimentar con algo fuera de lo común. Se trata de grabar sonidos en forma digital en la memoria de la máquina (**TS 1000** y compatibles **con una configuración de al menos 16 Kb**).

Lamentablemente, el programa, variables y archivo de pantalla ocupan unos **6 Kb** de memoria, o sea que nos quedan unos **10 Kb** libres para depositar allí la información recogida del exterior... que se consumen en unos cuantos segundos de grabación.

Pero este programa dispone de un **graficador de sonido** que nos muestra la amplitud del sonido registrado bit a bit, lo cual es suficiente para pasarse un buen rato entusiasmado observando la diferencia entre la voz de un hombre y la de una mujer, o entre sonidos provenientes de una fauta o una cuerda de guitarra... etc.

Paso a detallar cómo se usa el programa:

Antes de cargarlo, ya sea desde el cassette o desde teclado, **es muy importante que hagamos POKE 16389,88** y luego **NEW**.

Con esto fijamos el **RAMTOP** de modo de hacer un **BASIC** de tan solo 6 Kb, y el resto de la **RAM** por encima –10 Kb– nos quedan libres para grabar información.

Les recomiendo poner la máquina en modo **FAST** para no sulfurarse al ingresar la línea **1 REM**, que debe tener **como mínimo 250 caracteres**.

En las líneas **130** a **170**, que pueden resultar complicadas de entender, he puesto un mecanismo de control de errores ara prevenir cualquier equivocación en los largos strings de líneas **30** a **60**.

Una vez cargado el programa y hecha una copia de seguridad podemos digitar **RUN** y empezar a investigar el sonido que deseamos.

La manera de hacerlo es conectar la entrada EAR de la computadora a la ficha EAR del grabador, y poner en modo PLAY RECORD el aparato al mismo tiempo que pulsamos la opción **'G'** del menú (para grabar).

La ficha MIC del grabador debe estar desconectada, o bien podemos usarla para conectar un micrófono exterior. En cualquiera de los dos casos, hablando frente al micrófono **observaremos líneas horizontales en la pantalla**, señal de que la información está llegando a la computadora.

(sigue en pág. 39)

Tabla de variables

A$	Contiene el LM
C	Contiene la suma de cada instrucción
X	La dirección a **POKE**ar.
F	Para explorar **A$** en el bucle **FOR/NEXT**.
P	El valor extraído de **A$**.
S	Variable no definida para detener el programa en caso de error de tipeo.

–Todo esto hasta la línea **210**, donde un **CLEAR** borra todas las variables–

A$	Ahora contiene el mensaje a imprimir en la pantalla.
X	Varias variables de control de **FOR/NEXT**.
X$	El valor de **INKEY$**.
A	Utilizada en los **INPUT**s.

```
  1 REM .....
.....
.....
.....
.....
.....
.....
.....
.....
.....
  10 REM
  - GRABADOR DIGITAL -
  DANIEL WESELKA-EZEIZA,P.B.A.
  PROGRAMACION POPULAR NO. 10
  CZ1000/1500-TK83/85  3.5 KB
  20 FAST
  30 LET A$="210058237CFE80C8160
00608CB0ACD9E40C87F2802CBFA10F37
218E7DBFECB7F2802D3FF0E050D20FDD
BFEC9210058237CFE80C8060866CB0AC
B7A20070E0A0D"
  40 LET A$=A$+"20FD1803CDCD4010
EE15E4DBFE0E0A0D20FDD3FF0E050D20
FDDEFEC9E5C5D5CDBB02144D3E002C23
04CDBD077ED1C1E1C91608CB0BCB4328
0804CD0A4104CD0A"
  50 LET A$=A$+"411520EFCD0A41C9
E5D5C52A3040E5213E0C2230400CDB20B
E12230400C1D1E13E00C92A0C400600023
237EFE76280528772318F52B3600232323
0478FE13C816E921"
  60 LET A$=A$+"0058CDDE40FE76C8
FE2420170E3F060A7CFE8028EDE5C5CD
2441C1E1235ECDF34018DFFE3F2007F5
CD6908E118D4FE00C8303444DC916C9"
  70 LET C=0
  80 LET X=16514
  90 FOR F=1 TO 494 STEP 2
  110 LET P=CODE A$(F)*16+CODE A$
(F+1)-476
  120 POKE X,P
  130 LET C=C+P
  140 IF F=187 THEN IF C<>5869 TH
EN PRINT "ERROR EN LINEA 30";S
  150 IF F=255 THEN IF C<>14161 T
HEN PRINT "ERROR EN LINEA 40";S
  160 IF F=383 THEN IF C<>20186 T
HEN PRINT "ERROR EN LINEA 50";S
  170 IF F=493 THEN IF C<>26330 T
HEN PRINT "ERROR EN LINEA 60";S
  190 LET X=X+1
  200 NEXT F
  210 CLEAR
  220 SLOW
  230 REM ----------------------
  240 LET A$="..... "ENTER = MENU"
.....
.....
  250 GOSUB 930
  260 FOR X=1 TO LEN A$-20
  270 LET X$=INKEY$
  280 PRINT AT VAL "21",VAL "5";A
$(X TO X+20)
  290 IF X$<>"" THEN GOTO 320
  300 NEXT X
  310 GOTO 260
  320 PRINT AT 21,5;A$(19 TO 39)
  330 IF X$=CHR$ 118 THEN GOTO 25
0
  340 IF X$<>"G" THEN GOTO 390
  350 FAST
  360 RAND USR 16514
  370 SLOW
  380 GOTO 250
  390 IF X$<>"P" THEN GOTO 460
  400 REM "REPRODUCIR"
  410 FAST
  420 RAND USR 16558
  430 IF INKEY$="" THEN GOTO 430
  440 IF INKEY$="R" THEN GOTO 420
  450 SLOW
  460 IF X$<>"S" THEN GOTO 580
  470 LET D=USR 16705
  480 IF INKEY$="D" THEN PRINT AT
18,5;D
  490 IF INKEY$<>"" THEN GOTO 490
  500 LET X$=INKEY$
  510 IF INKEY$="" THEN GOTO 500
  520 IF X$<>"S" THEN GOTO 560
  530 POKE 16706,D-256*INT (D/256
)
  540 POKE 16707,INT (D/256)
  550 GOTO 470
  560 POKE 16705,88
  570 POKE 16707,88
  580 IF X$<>"V" THEN GOTO 750
  590 REM
  600 POKE 16701,22
  610 POKE 16695,155
  620 FOR X=1 TO 33
  630 RAND USR 16676
  640 NEXT X
  650 PRINT AT 5,1;"INGRESE VELOC
IDAD DE GRABACION"
  660 PRINT AT 7,3;"( DE 1 HASTA
30 )"
  670 PRINT AT 9,3;" NORMAL  5 "
  680 PRINT AT 11,3;"AHORA:";PEEK
16551
  690 INPUT A
  700 IF A<1 OR A>20 THEN GOTO 69
0
  710 POKE 16551,A
  720 POKE 16701,19
  730 POKE 16695,0
  740 GOTO 260
  750 IF X$<>"B" THEN GOTO 920
  760 REM
  770 POKE 16701,22
  780 POKE 16695,155
  790 FOR X=1 TO 33
  800 RAND USR 16676
  810 NEXT X
  820 PRINT AT 5,1;"INGRESE VELOC
IDAD DE "
  830 PRINT AT 6,1;"REPRODUCCION"
  840 PRINT AT 7,3;"( DE 1 HASTA
30 )"
  850 PRINT AT 9,3;" NORMAL 10 "
  860 PRINT AT 11,3;"AHORA:";PEEK
16576
  870 INPUT A
  880 IF A<1 OR A>20 THEN GOTO 87
0
  890 POKE 16576,A
  900 POKE 16701,19
  910 POKE 16695,0
  920 GOTO 260
  930 REM "MENU"
  940 CLS
  950 PRINT "PULSE ""8"" PARA GRA
FICO"
  960 PRINT ,,,,"Y ""D"" PARA VER
LA DIRECCION"
  970 PRINT "QUE SE EXPLORA"
  980 PRINT AT 10,0;"  ""G"" PARA
GRABAR"
  990 PRINT ,," ,""R"" PARA REPRO
DUCIR"
 1000 PRINT ,,"  ""Z"" = COPY"
 1010 PRINT ,,"  ""U"" VELOCIDAD
DE GRABACION"
 1020 PRINT ,,"  ""B"" VELOCIDAD
DE REPRODUCCION"
 1030 RETURN
```

Congelador de listado

Softi nos demuestra la diferencia existente entre una C-64 y una heladera!?!?....

Esta vez tenemos un programaa muy práctico, más que programa es una rutinita que está ubicada entre las líneas **20** y **70** de dicho programita.
¿Para qué sirve? Con el programa ejecutado (y autodestruido) se puede hacer o cargar cualquier programa en basic.
Pidamos el listado de cualquier programa, al presionar la tecla **SHIFT**, veremos que el listado se detiene o congela en la pantalla, si soltamos dicha tecla, el listado continúa, y si presionamos la tecla **ALPHA LOCK** el listado se congela y continuará cuando presionemos nuevamente dicha tecla.

Atención. Peligro. Una vez cargado el programa **no ejecutar hasta tenerlo almacenado en cinta o disco ya que al ejecutarse, automáticamente se destruye.**

```
READY.

 1 REM  ******************************
 2 REM  **                          **
 3 REM  **  I N T E R R U P        **
 4 REM  **                          **
 5 REM  **   PROGRAMACION POPULAR   **
 6 REM  **                          **
 7 REM  **  C O M M O D O R E  6 4  **
 8 REM  **                          **
 9 REM  ******************************
10 FOR T=12736 TO 12798:READ A
11 POKE T,A:NEXT
20 V=53248:POKE2047,199:POKEV+21,128
21 POKE V+46,01:POKEV+14,0:POKEV+15,120
30 POKE V+29,128
40 POKE 251,5:POKE 252,3
54 REM
55 REM
56 REM  DEFINICION DEL SPRITE
57 REM
100 DATA 128,0,8,60,0,0,34,0,0,34,0,0
101 DATA 60,0,0,32,0,0,33,0,0,34,128,0
110 DATA 0,0,0,0,0,0,0,0
111 DATA 60,0,0,034,0,0,34,0,0,60,0
120 DATA 0,32,0,0,33,0,0,34
121 DATA 128,0,1,0,0,0,0,0,0,0,0,128,0,8
400 REM
500 REM  ***  ESTA ES LA RUTINA  ***
501 REM
510 FOR T=49152 TO 49229:READ A
511 POKE T,A:NEXT:SYS 49152
520 DATA 120,169,13,141,20,3,169,192
521 DATA 141,21,3,88,96,198
522 DATA 251,165,251,240,3,76
530 DATA 74,192,173,16,208,41,128,240
531 DATA 23,173,14,208,201
532 DATA 101,208,31,173,16,208
540 DATA 37,127,141,16,208,169,0,141,14
541 DATA 208,76,67,192,173
542 DATA 14,208,201,255,208
550 DATA 8,173,16,208,5,128,141,16,208
551 DATA 238,14,208,165,252,133
552 DATA 251,76,49,234,0

READY.
```


Baudio s.a.
COMPUTACION E INFORMATICA

Organización Vivas S. A.

Lo que nos distingue no es el precio...
Tenemos una modalidad distinta:
brindar servicios y ganar amigos.

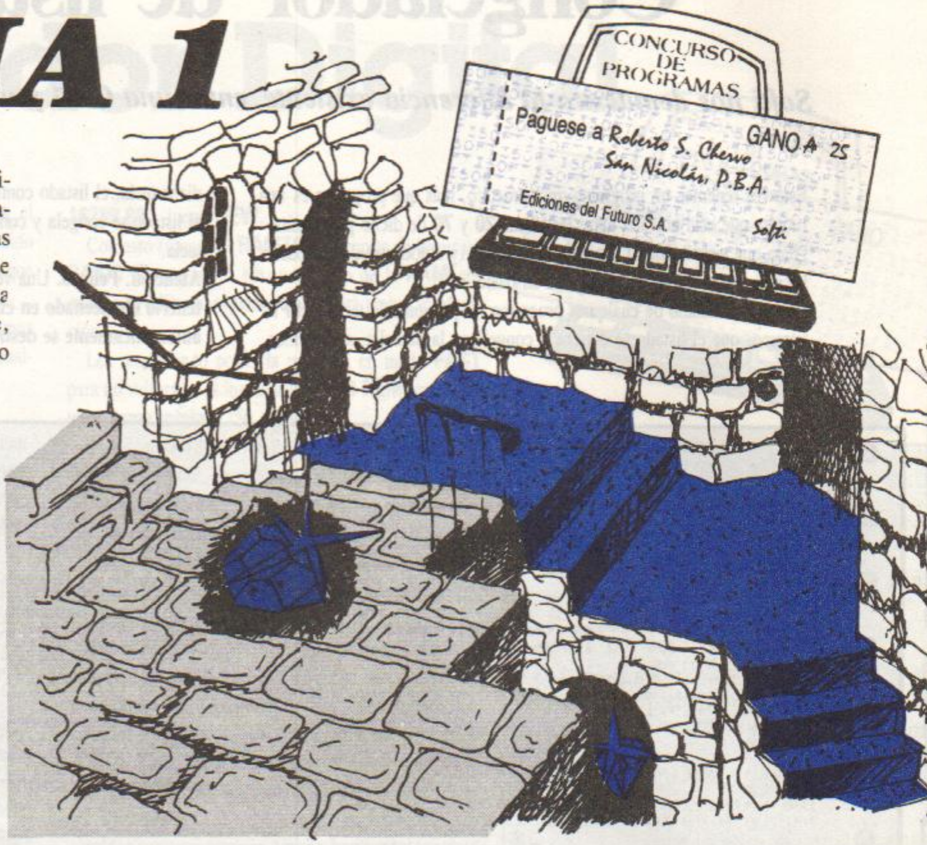
Ref. 26

- Venta de equipos y accesorios para Commodore 16/64/128
- Cursos técnicos especiales para Commodore:
  Nivel 1 (principiantes): 3 meses Nivel 2 (intermedios): 3 meses Nivel 3 (avanzados): 2 meses
  Terminando los cursos como experto en Commodore
  Cupos limitados - Grupos por nivel de conocimientos y edades
- Centro de Demostración y Venta de los Juegos del Pirata para todas las marcas.
- Programas especiales y "a pedido" para Commodore

Te esperamos en J.E. Uriburu 629
(1027) - Tel.: 48-5441/8829

IMPORTANTE:
Sólo inscribiéndote antes del 10/3/86 en los Cursos para Commodore, tendrás un lugar asegurado.

ALPHA 1



Desde tu cabina de comando deberás dirigir al robot **Alpha 1** por un laberinto plagado de peligros, para que pueda recoger las piezas de su nave y la mayor cantidad de diamantes posibles con el fin de regresar a la nave, una vez que esté armada totalmente, antes que se acabe el tiempo o consuma todo el oxígeno disponible.




```
  10 REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           A L P H A   1         @
@ Roberto S Chervo-San Nicolas @
@ Rev.Programacion Popular-Nº9 @
@ CZSpectrum-TK90X-TS2068  13K @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  50 BEEP 1,10: CLS : GO SUB 188
0
  60 REM graficos
  70 FOR v=USR "a" TO USR "u"+7:
 READ c: POKE v,c: NEXT v
  80 DATA 24,60,66,126,189,165,1
65,165: REM a
  90 DATA 24,60,126,231,231,126,
60,24: REM b
 100 DATA 255,1,253,133,165,189,
129,255: REM c
 110 DATA 1,3,7,14,28,41,115,231
: REM d
 120 DATA 231,255,255,0,0,255,0,
255: REM e
 130 DATA 128,192,224,112,56,156
,206,231: REM f
 140 DATA 239,239,239,108,40,40,
124,124: REM g
 150 DATA 247,247,247,54,20,20,6
2,62: REM h
 160 DATA 165,165,165,189,126,10
2,60,24: REM i
 170 DATA 15,48,127,216,216,127,
48,15: REM j
 180 DATA 240,12,254,27,27,254,1
2,240: REM k
 190 DATA 24,60,24,255,189,153,6
0,102: REM l
 200 DATA 0,126,66,90,90,66,126,
0: REM m
 210 DATA 0,0,24,60,60,24,0,0: R
EM n
 220 DATA 24,60,126,126,231,219,
189,165: REM o
 230 DATA 219,219,219,219,219,21
9,219,219: REM p
 240 DATA 255,255,0,255,255,0,25
5,255: REM q
 250 DATA 127,255,192,223,223,21
6,219,219: REM r
 260 DATA 219,219,216,223,223,19
2,255,127: REM s
 270 DATA 219,219,27,251,251,3,2
55,254: REM t
 280 DATA 254,255,3,251,251,27,2
19,219: REM u
 290 LET rec=0: LET n$="SPECTRUM
"
 300 GO SUB 1680
 310 REM laberinto
 320 GO SUB 1520
 330 LET pun=0: LET ox=3: LET ti
=0: LET a$="A"
 340 BORDER 3: PAPER 6: CLS
 350 FOR k=0 TO 21: PRINT INK 3;
AT k,31;"■": NEXT k
 360 LET b$="...................
..............": FOR b=1 TO 21: PR
INT AT b,0;b$: NEXT b
 370 FOR b=2 TO 20 STEP 2: FOR a
=1 TO 29 STEP 2: PRINT INK 2;AT
b,a;"▣": NEXT a: NEXT b
 380 PRINT AT 0,0;"OXIGENO:"; IN
K 2;ox; INK 0;AT 0,11;"TIEMPO:";
 INK 2;ti; INK 0;AT 0,21;"PUNTOS
:"; INK 2;pun
 390 PRINT #0; INK 9;"TIEMPO LIM
ITE...100"
 400 REM obstaculos
 410 FOR v=1 TO 30
 420 LET o=INT (RND*21)+1
 400 LET p=INT (RND*16)*2
 440 PRINT PAPER 1; INK 5; BRIGH
T 1; FLASH 1;AT o,p;"▣": FLASH 0
 450 NEXT v
 460 FOR w=1 TO 20
 470 LET x=INT (RND*21)+1
 480 LET y=INT (RND*16)*2
 490 PRINT FLASH 1;AT x,y;"♣": F
LASH 0
 500 NEXT w
 510 FOR f=1 TO 35
 520 LET m=INT (RND*21)+1
 530 LET l=INT (RND*16)*2
 540 PRINT INK 3;AT m,l;"◆"
 550 NEXT f
 560 FOR r=1 TO 5
 570 LET j=INT (RND*21)+1
 580 LET k=INT (RND*16)*2
 590 PRINT INK 1; FLASH 1;;AT j,
k;"•": FLASH 0
 600 NEXT r
 610 INK 1: PRINT AT 8,13;"▄▄▄
": FOR h=9 TO 11: PRINT AT h,13;
"▌ ▐": NEXT h: PRINT AT 12,13;
"▙#▟": INK 0
 620 PRINT INK 1; BRIGHT 0;AT 3,
3;"▀"; BRIGHT 0; INK 2;AT 5,15;"
Y"; INK 3; BRIGHT 0;AT 3,26;"A"
 630 PRINT INK 4; BRIGHT 0;AT 19
,3;"/"; INK 5; BRIGHT 0;AT 17,15
;"\"; INK 0; BRIGHT 0;AT 19,26;"
Y"
 640 REM musica comienzo
 650 RESTORE 660
 660 DATA 28,24,28,33,28,33,28,3
3,29,31,29,31,29,34,29,30,32,30,
29,30,33,30
 670 FOR f=1 TO 22
 680 READ a: BEEP .1,a-3: NEXT f
 690 BEEP .2,15
 700 REM posicion robot y tiempo
 710 POKE 23673,255: POKE 23672,
255
 720 LET c=29: LET d=19
 730 LET dd=d: LET cc=c
 740 PRINT INK 0; FLASH 1;AT d,c
a$: BEEP .01,15
 750 DEF FN t()=INT ((PEEK 23674
+256*PEEK 23673+PEEK 23672)/50):
 LET ti=FN t()
 760 REM movimiento
 770 PRINT AT d,c;" "
 780 IF INKEY$=c$ THEN LET a$="€
"
 790 IF INKEY$=d$ THEN LET a$="∋
"
 800 IF INKEY$=g$ THEN LET a$="W
"
 810 IF INKEY$=f$ THEN LET a$="A
"
 820 IF a$="€" THEN LET c=c-1: I
F c<0 THEN LET c=30
 830 IF a$="∋" THEN LET c=c+1: I
F c>30 THEN LET c=0
 840 IF a$="W" THEN LET d=d+1: I
F d>21 THEN LET d=1
 850 IF a$="A" THEN LET d=d-1: I
F d<1 THEN LET d=21
 860 PRINT AT 0,18; INK 2;ti;" "
: IF ti=100 THEN PRINT AT 0,18;
FLASH 1;"100": FOR w=0 TO 29: BE
EP .1,10: NEXT w: GO SUB 1590: G
O TO 1370
```

Para disfrutar este estupendo programa ganador de **Roberto** sólo tienes que tipearlo en tu máquina, sin olvidar hacer copias de seguridad parciales, por las dudas... Una vez cargado y depurado de "bugs", podrás **SAVE**'arlo y, **RUN** mediante, iniciarás tu misión con **Alpha 1**...

El programa es un poco largo pero realmente vale la pena. Si queremos ahorrar tiempo, podríamos no ingresar las líneas **1450/1510** y **1880/1940** –ver estructura del programa– modificando en consecuencia las líneas **1430** y **50**.



Si bien las instrucciones saldrán por pantalla, te damos algunas aclaraciones y sugerencias del amigo **Roberto**:

- Cada ez que **Alpha 1** salga por un extremo del laberinto, ingresará por el extremo opuesto.
- Si se arma rápidamente la nave y todavía falta bastante para que el contador de tiempo llegue a 100, convendrá seguir tratando de obtener puntos (sin alejarse mucho del hangar), ya que cuando se entra en el hangar la misión termina.

ESTRUCTURA DEL PROGRAMA

50	Sonido de aviso de finalización de carga y presentación en pantalla
60/280	Creación de GDU
290	Nombre de la computadora: cambiar si no es Spectrum
300	Salto a S/r instrucciones
310/390	Imprime pantalla del laberinto
400/630	Imprime obstáculos y hangar
640/690	Música de comienzo de juego
700/740	Posiciona a **Alpha 1**
750	Controla el tiempo
760/860	Fija posición de robot según la dirección que lleva y controla movimiento
870/1070	Desarrollo del juego
1080/1260	Música y pantalla de nave armada
1270/1360	S/r de caída en trampa
1370/1410	Final del juego y récord
1420/1440	Elección a otro juego
1450/1510	S/r de autodestrucción
1520/1580	S/r de elección de teclas
1590/1670	S/r de música de tiempo
1680/1860	S/r de instrucciones
1870	S/r de "teletipo"
1880/1940	S/r de presentación

```
870 REM juego
880 IF ATTR (d,c)=240 THEN FOR f=1 TO 5: BEEP .01,14: NEXT f: LET ox=ox-1
890 IF ATTR (d,c)=241 THEN LET ox=ox+1: BEEP .5,15: IF ox>=3 THEN LET ox=3
900 IF ATTR (d,c)=114 THEN LET d=dd: LET c=cc
910 IF ATTR (d,c)=115 THEN FOR f=0 TO 30: BEEP .01,50: OUT 246,z: NEXT z: LET pun=pun+10: PRINT AT 0,28; INK 2;pun
920 IF ATTR (d,c)=113 THEN LET d=dd: LET c=cc
930 IF ATTR (d,c)=205 THEN GO TO 1270
940 IF ATTR (d,c)=246 THEN PRINT INK 0;AT 11,15;"A": GO TO 1080
950 IF SCREEN$ (d,c)="." THEN LET pun=pun+1: PRINT INK 2;AT 0,28;pun
960 PRINT AT 0,8; INK 2;ox
970 IF ox<=0 THEN PRINT FLASH 1;AT 0,8,"0": FOR i=1 TO 6: PRINT AT d,c; INK i;a$: BEEP .11,50: NEXT i: FOR i=-30 TO 60: BEEP .001,i: NEXT i: FOR i=60 TO -30 STEP -1: BEEP .001,i: NEXT i: FLASH 0: GO TO 1370
980 IF ATTR (d,c)=49 THEN BEEP .1,25: PRINT AT 10,15;"=": LET pun=pun+50
990 IF ATTR (d,c)=50 THEN BEEP .1,25: PRINT AT 11,14;"T": LET pun=pun+50
1000 IF ATTR (d,c)=51 THEN BEEP .1,25: PRINT AT 9,15;"A": LET pun=pun+50
1010 IF ATTR (d,c)=52 THEN BEEP .1,25: PRINT AT 10,14;"/": LET pun=pun+50
1020 IF ATTR (d,c)=53 THEN BEEP .1,25: PRINT AT 10,16;"\": LET pun=pun+50
1030 IF ATTR (d,c)=48 THEN BEEP .1,25: PRINT AT 11,16;"Y": LET pun=pun+50
1040 PRINT INK 2;AT 0,28;pun
1050 IF (ATTR (9,15)=112 AND ATTR (10,14)=112 AND ATTR (10,15)=112 AND ATTR (10,16)=112 AND ATTR (11,14)=112 AND ATTR (11,16)=112 THEN PRINT PAPER 6;AT 12,15;" "; INK 6; FLASH 1;AT 11,15;" "
1060 LET dd=d: LET cc=c
1070 GO TO 740
1080 REM Musica final
1090 BEEP .1,10: BEEP .1,15: BEEP 1,20: BEEP .1,15: BEEP .4,10
1100 CLS : BORDER 2: PAPER 6: INK 0: CLS
1110 PRINT AT 2,8;"MISION CUMPLIDA";AT 3,8;"          "
1120 PRINT AT 6,9;"ALPHA 1 LOGRO";AT 8,6;"RECONSTRUIR SU NAVE"
1130 PRINT AT 12,7;"LA MISION TERMINO";AT 14,8;"CON TODO EXITO"
1140 RESTORE 1150
1150 DATA 3,4,5,3,4,5,3,4,5,3,4,5,3,4,5,3
1160 FOR b=1 TO 16
1170 READ a: BEEP .09,a: NEXT b
1180 BEEP .5,6: BEEP .5,5: BEEP .5,4: BEEP .5,3
1190 RESTORE 1200
1200 DATA 6,7,8,6,7,8,6,7,8,6,7,8,6,7,8,6
1210 FOR c=1 TO 16
1220 READ d: BEEP .09,d: NEXT c
1230 BEEP .5,9: BEEP .5,8: BEEP .5,7: BEEP .5,6
1240 FOR e=9 TO 24: BEEP .09,e: NEXT e
1250 BEEP .3,25
1260 PRINT FLASH 1;AT 18,8;"Pulsa una tecla": FLASH 0: PAUSE 0: GO TO 1370
1270 REM Caidas en trampas
1280 FOR a=10 TO 60: BEEP .01,a: NEXT a: FOR b=60 TO -10 STEP -1: BEEP .01,b: NEXT b
1290 CLS : BORDER 0: INK 0: CLS : FOR m=1 TO 704: PRINT INK 2;"■";: NEXT m
1300 PRINT AT 8,8;"MISION ABORTADA";AT 11,3;"ALPHA 1 CAYO EN UNA TRAMPA"
1310 RESTORE 1320
1320 DATA 8,9,10,5,9,3,7,9
1330 FOR b=1 TO 8
1340 READ d: BEEP .5,d: NEXT b
1350 BEEP .7,6
1360 PRINT AT 18,8; FLASH 1;"Pulsa una tecla"; FLASH 0: PAUSE 0: GO TO 1370
1370 REM Final
1380 CLS : BORDER 0: PAPER 0: INK 6: CLS : PRINT AT 5,11; FLASH 1;"GAME OVER": FOR g=0 TO 60: BEEP .01,56: OUT 246,g: NEXT g
1390 IF pun=rec THEN LET n$=n$
1400 IF pun>rec THEN LET rec=pun : PRINT AT 9,6;"FELICITACIONES,.!!";AT 12,6;"Superaste el Record": POKE 23609,50: INPUT "Nombre?(Max. 10 letras):"; LINE n$: IF LEN n$>10 THEN LET n$=n$( TO 10)
1410 POKE 23609,0: CLS : PRINT AT 5,9;"Tiempo: ";ti;AT 8,9;"Puntos: ";pun;AT 11,9;"El Record es ";rec;AT 14,9;"Lo hizo ";n$
1420 PRINT INK 4;AT 19,9;"Otro partido`?": IF INKEY$="s" OR INKEY$="S" THEN GO TO 320
1430 IF INKEY$<>"n" OR INKEY$="N" THEN GO TO 1450
1440 IF INKEY$<>"n" OR INKEY$<>"N" THEN GO TO 1420
1450 REM Autodestruccion
1460 CLS : BORDER 0: PAPER 0: INK 5: LET f$=" ESTE PROGRAMA SE AUTODESTRUIRA EN 9 SEGUNDOS": FOR a=1 TO LEN f$: BEEP .05,30: PRINT f$(a);: NEXT a
1470 FLASH 1: PRINT INK 7;AT 9,14;"┌─┐";AT 10,14;"│ │";AT 11,14;"└─┘": FLASH 0
1480 PRINT AT 10,15; INK 4;"9"
1490 LET cuenta=10
1500 LET cuenta=cuenta-1: PRINT INK 4;AT 10,15;cuenta: BEEP .01,45: PAUSE 60: IF cuenta=1 THEN FOR h=50 TO -20 STEP -1: BEEP .01,h: NEXT h: NEW
1510 GO TO 1500
1520 REM Eleccion teclas
1530 CLS : BORDER 6: PAPER 6: INK 1: BRIGHT 1: CLS
1540 LET k$=" Eleccion de Teclas de Direccion"
1550 FOR g=1 TO LEN k$: BEEP .00
```

● Algunos objetos no son muy visibles (principalmente algunas partes de la nave) pero sucede que los objetos son muchos y la cantidad de colores disponibles es reducida (el programa se desarrolla principalmente con **ATTR**).



● Para grabar el programa es aconsejable hacerlo con **SAVE "ALPHA-1" LINE 1**, para que se autoejecute al cargarlo.

● Las *''teclas de movimiento''* son definidas por el jugador antes de comenzar cada juego, para que no se arruine el teclado al usar siempre las mismas teclas (**Softinota**: correcto el criterio, pero bastaría con hacerlo al comienzo del programa, con las ''instrucciones'' para que no sea ''pesado'').

● Para poder regular el sonido (salvo en la **TK90X**, que ya sale por el parlante de TV) podemos conectar la salida MIC de la computadora a la entrada MIC del grabador y presionar PLAY (sin casete colocado).

TABLA DE VARIABLES

a$	Figura de **Alpha 1**
c	Columna de impresión del *robot*
cc	Ultima posición de **Alpha 1**(columna)
cuenta	Cuenta regresiva para autodestrucción
c$	Tecla hacia la izquierda
d	Fila de impresión de **Alpha 1**
d$	Tecla hacia la derecha
dd	Ultima posición de **Alpha 1** (fila)
FN (t)	Control del tiempo
f$	Tecla hacia arriba
g$	Tecla hacia abajo
j	Coordenada horizontal de *pastillas*
k	Coordenada vertical de *pastillas*
l	Coordenada vertical de *diamantes*
m	Coordenada horizontal de *diamantes*
n$	Nombre del jugador que posee el récord
o	Coordenada horizontal de *trampas*
ox	*Oxígeno*
p	Coordenada vertical de *trampas*
pun	Puntos
rec	Récord
ti	Tiempo
x	Coordenada horizontal de *invasores*
y	Coordenada vertical de *invasores*

```
7,40: PRINT INK 1; PAPER 7;k$(g)
;: NEXT g
1560 INK 2: PRINT AT 7,9;"Arriba
 =====> ";AT 9,9;"Abajo ======>
";AT 11,9;"Izquierda ==> ";AT 13
,9;"Derecha ====> "
1570 INK 0: INPUT "Tecla Arriba?
 ";f$: PRINT AT 7,23;f$: INPUT "
Tecla Abajo? ";g$: PRINT AT 9,23
;g$: INPUT "Tecla Izquierda? ";c
$: PRINT AT 11,23;c$: INPUT "Tec
la Derecha? ";d$: PRINT AT 13,23
;d$
1580 PRINT INK 3;AT 19,9;"Pulsa
una tecla": PAUSE 0: RETURN
1590 REM MUESTRA TIEMPO
1600 FOR g=1 TO 2
1610 RESTORE 1620
1620 DATA 18,18,20,18,20,22,20,1
8,20,22,20,18,20,22
1630 FOR a=1 TO 14
1640 READ b: BEEP .1,b: NEXT a
1650 BEEP .2,18
1660 NEXT g
1670 RETURN
1680 REM INSTRUCCIONES
1690 BORDER 0: PAPER 0: INK 6: C
LS
1700 PRINT AT 11,5;"NECESITAS IN
STRUCCIONES ?";AT 13,14;"(s/n)":
 PAUSE 0
1710 IF INKEY$="s" OR INKEY$="S"
 THEN GO TO 1740
1720 IF INKEY$="n" OR INKEY$="N"
 THEN GO TO 320
1730 IF INKEY$<>"s" OR INKEY$<>"
S" OR INKEY$<>"n" OR INKEY$<>"N"
 THEN GO TO 1700
1740 CLS : PRINT AT 0,14;"ALPHA
1"
1750 PRINT '': LET a$="   Tu misi
on es dirigir al robotALPHA 1 po
r un laberinto, para  recolectar
 la mayor cantidad de diamantes
posibles,armar su navey poder ll
egar a ella antes de  que se aca
be  el tiempo.": GO SUB 1870
1760 PRINT ''': LET a$="   Cuida
do con las trampas; si  ALPHA 1
cae en una de ellas.....la misio
n termina.": GO SUB 1870
1770 PRINT ''': LET a$="   La na
ve se arma en el centro del labe
rinto cada vez que ALPHA1 pase s
obre los componentes queencuentr
e en su camino.": GO SUB 1870
1780 FLASH 1: PRINT ''': LET a$=
"Pulsa una tecla": GO SUB 1870:
PAUSE 0: FLASH 0: CLS
1790 LET a$="   Cuando se arme c
ompletamente la nave, la puerta
del hangar seabrira automaticame
nte; para queALPHA 1 pueda entra
r y completarsu mision.": GO SUB
 1870
1800 PRINT ''': LET a$="   Para
llevar a cabo su mision,el robot
 dispone de 3 unidades  de oxige
no.": GO SUB 1870
1810 PRINT ''': LET a$="   Si ch
oca con algun invasor,  perdera
una unidad, pero puede  recupera
rla ingiriendo pastillasde oxige
no.": GO SUB 1870
1820 PRINT ''': LET a$="   Si pi
erde las 3 unidades, la mision q
ueda inconclusa.": GO SUB 1870
1830 FLASH 1: PRINT ''': LET a$="
Pulsa una tecla": GO SUB 1870: P
AUSE 0: FLASH 0
1840 CLS : PRINT '''';TAB 11;"PUN
TUACION";AT 4,11;"          "
1850 PRINT AT 7,4;"Puntos      (.
)... 1 punto";AT 9,4;"Diamantes
 (◆)...10 puntos";AT 11,4;"Parte
s de nave...50 puntos";AT 13,4;"
Invasores  (♣)...-1 oxigeno";AT
15,4;"Pastillas  (•)...+1 oxigen
o";AT 17,4;"Trampas    (▣)...Mue
rte"
1860 FLASH 1: PRINT AT 21,8;"Pul
sa una tecla": PAUSE 0: FLASH 0:
 CLS : GO TO 320
1870 FOR n=1 TO LEN a$: PRINT a$
(n);: BEEP .01,20: NEXT n: PAUSE
 10: RETURN
1880 REM CREA CARACTERES
1890 BORDER 5: PAPER 7: INK 1: C
LS
1900 LET t=175: FOR c=0 TO t: PL
OT 0,c: DRAW OVER 1;255,t: LET t
=t-2: NEXT c
1910 PLOT 0,0: DRAW 255,0: PLOT
0,175: DRAW 255,0: PLOT 0,1: DRA
W 255,0: PLOT 0,174: DRAW 255,0
1920 PLOT 0,0: DRAW 255,175: PLO
T 0,175: DRAW 255,-175: PLOT 0,1
: DRAW 254,174: PLOT 0,174: DRAW
 254,-174
1930 INK 2: PRINT AT 4,13;"ALPHA
";AT 6,15;"1";AT 15,13;"SOFTI";A
T 17,15;"y";AT 19,7;"Roberto S.
Chervo"
1940 PAUSE 200: RETURN
```

CGDU >>>>>> GRAPHICS + Letras

A ◆ ▣ ◢ ☰ ◣ ▼ ▼ ▼ ∈ ∋ ✱ ⊞ • ▲ ▮▮

A B C D E F G H I J K L M N O P

▤ ▤ ▤ ▤ ▤

Q R S T U



Números romanos



Este interesante programa de Andrés Bitenc, de Ituzaingó, tiene un enfoque netamente didáctico desde su planteo. Realiza una evaluación de un grupo de alumnos sobre un tema de matemática elemental.

Por un lado el programa da una pauta importante para los docentes interesados en el fenómeno y aplicaciones de la computadora en el aula, ejemplificando correctamente una posible "veta" de la informática educativa, a la que tanta falta le hace un buen desarrollo.

Por otro lado el programa en sí es digno de estudio serio por la complejidad de su diagramación, debida a que efectivamente realiza una "traducción" de una a otra sintaxis, la de los números arábidos y la de los romanos. Esta traducción –que en principio se encarga al alumno evaluado– es necesaria a los efectos de verificar la respuesta.

Veamos la explicación, muy bien documentada, que nos da el amigo Andrés:

(sigue en pág. 33)

TABLA DE VARIABLES

Alfanuméricas

A$** = Unidades de mil expresadas con números romanos. **B$ = Centenas expresadas con números romanos. **C$** = Decenas expresadas con números romanos. **D$** = Unidades expresadas con números romanos. **G$** = Número romano respondido por el alumno. **O$** = Opción. **R$** = Número romano. **X$(N)** = Nombre del alumno **N**. **Numéricas NA** = Número arábigo. **UM** = Unidades de mil. **C** = Centenas. **D** = Decenas. **U** = Unidades. **C(N)** = Calificación del alumno **X$(N)**. **E** = Número total de ejercicios para cada alumno. **G** = Número arábigo respondido por el alumno. **J** = Contador de ejercicios. **K** = Número de alumnos evaluados. **N** = Número de orden de los alumnos. **P** = Contador de puntos. **T** = Variable temporizadora.

ESTRUCTURA DEL PROGRAMA

10 = Título. **12-14** = Dimensionamientos. **16-30** = Presentación del título, instrucciones, opciones y solicitud a la maestra. **32-52** = La maestra indica la opción y el número de ejercicios que deberá resolver cada alumno. **54-68** = Informe a los alumnos. **70-74** = Determinación de las condiciones iniciales. **86** = Imprime el número del ejercicio. **88-90** = Generación al azar del número. **92-98** = Descomposición del número en unidades, decenas, centenas y unidades de mil. **100-168** = Conversión del número arábigo en número romano. **170-204** = Presentación del ejercicio, respuesta del alumno, corrección y acumulación del puntaje. **206-210** = Controla la cantidad de ejercicios de cada alumno. **212-216** = Cálculo de la calificación con informe al alumno. **218-250** = Generación del listado de alumnos evaluados con sus respectivas calificaciones y su presentación en pantalla. **252-266** = Subrutinas de temporización y borrado de la pantalla. **268** = Stop.

Nota 1: Con la sentencia **90 LET NA = INT (RND * 3999) +1**, se generan números desde **1** hasta **3999** porque de allí en adelante se colocan sobre los numeros romanos rayas que indican la multiplicación del número por 1000. Ejemplo: 4000 = IV.

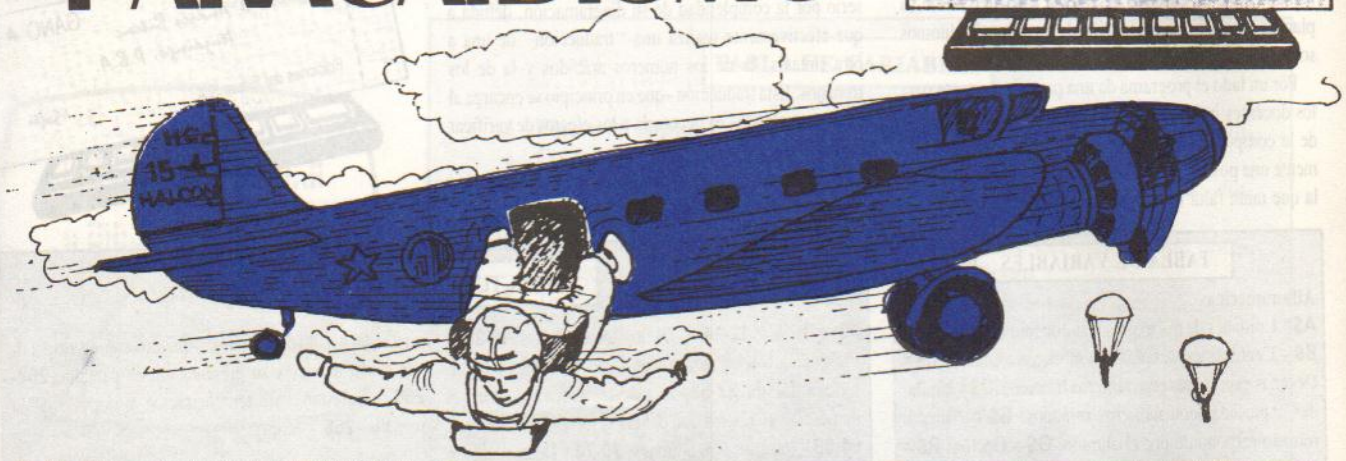
Nota 2: En compensación a la corrección rigurosa, la calificación se redondea, en los casos en que contiene decimales, a la unidad inmediata superior.

Sentencia **212 LET C(N) = INT (10 * P / E + 0.9)**

```
  10 REM
  #NUMEROS ROMANOS#
  ANDRES BITENC - ITUZAINGO
  PROGRAMACION POPULAR  NO.10
  CC1000-1500-TK83/85   4,0 KB
  12 DIM X$(45,23)
  14 DIM C(45)
  16 PRINT AT 5,10;"T E M A",AT
10,6;"NUMEROS ROMANOS"
  18 GOSUB 264
  20 PRINT AT 1,7;"INSTRUCCIONES
",AT 7,1;"CUANDO HAYAN SIDO EVAL
UADOS";AT 9,1;"TODOS LOS ALUMNOS
 Y APAREZCA";AT 11,1;"EN PANTALL
A.  AT 15,1;"NOMBRE DEL ALUMNO P
ULSE ENTER"
  22 GOSUB 260
  24 PRINT AT 0,8;"OPCIONES",AT
2,1;"LOS ALUMNOS DEBERAN ESCRIBI
R":
  26 PRINT AT 4,1;"A) CON NUMERO
S ROMANOS, LOS";AT 6,4;"NUMEROS
ARABIGOS DADOS"
  28 PRINT AT 8,1;"B) CON NUMERO
S ARABIGOS, LOS";AT 10,4;"NUMERO
S ROMANOS DADOS"
  30 PRINT AT 13,3;"-";AT 14,1;"
SEÑORITA MAESTRA ";AT 16,1;"INDI
QUE LA OPCION A-B      ";
  32 INPUT O$
  34 IF O$="A" THEN GOTO 40
  36 IF O$="B" THEN GOTO 44
  38 GOTO 32
  40 PRINT "A"
  42 GOTO 46
  44 PRINT "B"
  46 PRINT AT 19,1;"CUANTOS EJER
CICIOS QUIERE";AT 21,1;"PARA CAD
A ALUMNO?";
  48 INPUT E
  50 PRINT AT 21,26,E
  52 GOSUB 264
  54 PRINT AT 2,1;"ALUMNOS ";AT
5,1;"ESCRIBAN CON NOTACION ";
  56 IF O$="A" THEN GOTO 62
  58 PRINT "ARABIGA"
  60 GOTO 64
  62 PRINT "ROMANA"
  64 PRINT AT 7,1;"LOS NUMEROS P
RESENTADOS EN";AT 9,1;"PANTALLA"
  66 PRINT AT 14,1;"PRESTEN MUCH
A ATENCION EN LA";AT 16,1;"ESCRI
TURA, PORQUE EN LA CORREC-";AT 1
8,1;"CION NO PERDONO NINGUN ERRO
R"
  68 PRINT AT 21,1;"PULSE ENTER"
  70 INPUT A$
  72 CLS
  74 LET P=0
  76 LET J=1
  78 PRINT AT 3,2;"NOMBRE DEL AL
UMNO:"
  80 INPUT X$(N)
  82 IF X$(N,1)=" " THEN GOTO 24
2
  84 PRINT AT 9,2;X$(N)
  86 GOSUB 264
  88 PRINT AT 2,1;"EJERCICIO NUM
ERO: ";J
  90 RAND 0
  92 LET NA=INT (RND*3999)+1
  94 LET UM=INT (NA/1000)
  96 LET C=INT ((NA-1000*UM)/100
)
  98 LET D=INT ((NA-1000*UM-100*
C)/10)
 100 LET U=NA-1000*UM-100*C-10*D
 102 IF UM=3 THEN LET A$="MMM"
 104 IF UM=2 THEN LET A$="MM"
 106 IF UM=1 THEN LET A$="M"
 108 IF UM=0 THEN LET A$=""
 110 IF C=9 THEN LET B$="CM"
 112 IF C=8 THEN LET B$="DCCC"
 114 IF C=7 THEN LET B$="DCC"
 116 IF C=6 THEN LET B$="DC"
 118 IF C=5 THEN LET B$="D"
 120 IF C=4 THEN LET B$="CD"
 122 IF C=3 THEN LET B$="CCC"
 124 IF C=2 THEN LET B$="CC"
 126 IF C=1 THEN LET B$="C"
 128 IF C=0 THEN LET B$=""
 130 IF D=9 THEN LET C$="XC"
 132 IF D=8 THEN LET C$="LXXX"
 134 IF D=7 THEN LET C$="LXX"
 136 IF D=6 THEN LET C$="LX"
 138 IF D=5 THEN LET C$="L"
 140 IF D=4 THEN LET C$="XL"
 142 IF D=3 THEN LET C$="XXX"
 144 IF D=2 THEN LET C$="XX"
 146 IF D=1 THEN LET C$="X"
 148 IF D=0 THEN LET C$=""
 150 IF U=9 THEN LET D$="IX"
 152 IF U=8 THEN LET D$="VIII"
 154 IF U=7 THEN LET D$="VII"
 156 IF U=6 THEN LET D$="VI"
 158 IF U=5 THEN LET D$="V"
 160 IF U=4 THEN LET D$="IV"
 162 IF U=3 THEN LET D$="III"
 164 IF U=2 THEN LET D$="II"
 166 IF U=1 THEN LET D$="I"
 168 IF U=0 THEN LET D$=""
 168 LET R$=A$+B$+C$+D$
 170 IF O$="A" THEN GOTO 186
 172 PRINT AT 8,1;"ROMANO",AT 8,
13;"ARABIGO"
 174 PRINT AT 10,1,R$
 176 INPUT G
 178 PRINT AT 10,20;G
 180 IF G=NA THEN GOTO 202
 182 PRINT AT 13,1;"LA RESPUESTA
";AT 15,1;"CORRECTA ES  ";AT 15,
20;NA
 184 GOTO 198
 186 PRINT AT 8,1;"ARABIGO";AT 8
 17;"ROMANO"
 188 PRINT AT 10,2;NA
 190 INPUT G$
 192 PRINT AT 10,17;G$
 194 IF G$=R$ THEN GOTO 202
 196 PRINT AT 13,1;"LA RESPUESTA
";AT 15,1;"CORRECTA ES  ";AT 15,
17;R$
 198 GOSUB 260
 200 GOTO 206
 202 LET P=P+1
 204 GOSUB 264
 206 IF J=E THEN GOTO 212
 208 LET J=J+1
 210 GOTO 86
 212 LET C(N)=INT (10*P/E+.9)
 214 PRINT AT 3,1,X$(N);AT 9,1,"
TU CALIFICACION ES: ";C(N)
 216 GOSUB 264
 218 IF N=45 THEN GOTO 224
 220 LET N=N+1
 222 GOTO 72
 224 LET K=45
 226 LET N=1
 228 PRINT "NOMBRE";AT 0,23;"CAL
IF."
 230 PRINT
 232 PRINT ,,X$(N);"  ";C(N)
 234 IF N=10 OR N=20 OR N=30 OR
N=40 THEN GOTO 248
 236 IF N=K THEN GOTO 268
 238 LET N=N+1
 240 GOTO 232
 242 CLS
 244 LET K=N-1
 246 GOTO 226
 248 LET N=N+1
 250 GOTO 228
 252 FOR T=1 TO 200
 254 NEXT T
 256 CLS
 258 RETURN
 260 FOR T=1 TO 100
 262 GOTO 254
 264 FOR T=1 TO 25
 266 GOTO 254
 268 STOP
```

TORNEO DE PARACAIDISMO





Es éste un juego que consiste en hacer aterrizar un paracaidista que se arroja desde un avión sobre una base en tierra. El avión pasa periódicamente por la pantalla y a cada salto la posición de la base varía aleatoriamente.

A modo de presentación de este magnífico programa basten las palabras del propio **Pablo**:

TABLA DE VARIABLES

CANT: Cantidad total de jugadores
CAR: Contiene un número de caracter elegido al azar que selecciona el gráfico de banderas a imprimir
JUG: Número del jugador que participa en cada momento
NOM$( ): Tabla que contiene los nombres de todos los jugadores
PUN ( ): Tabla con puntajes de los jugadores
SU: Puntaje del jugador que participa en cada momento
SAL: Cantidad de saltos que falta jugar
VEL: Determina la velocidad del viento. Depende de **CAR**
XPOS: Posición horizontal de la base, elegida al azar

ADVERTENCIA

Este es un juego que, por lo menos entre mis amigos, provoca muchísima adicción. Así que para procurar una larga vida a las computadoras de los jóvenes usuarios (aunque adultos... también!) les aconsejo que hagan un uso moderado de este programa. De otro modo, *mi conciencia no quedaría tranquila...*

Puede ser corrido en la **TI-99/4A** *con el módulo de BASIC Extendido únicamente*. La capacidad de memoria necesaria es de *16 K*, o sea, *no se necesita* expansión de memoria.

El juego puede ser usado por una sola persona o bien se pueden hacer competencias de hasta 9 participantes. La acción se puede controlar desde el teclado o desde la palanca de mando N° 1.

ESTRUCTURA DEL PROGRAMA

El programa está concebido como un bloque principal y varios bloques-subrutinas usados por éste según se requiera.

Programa principal **10-950**
Rutina definición de caracteres **960-1200**
Rutina dibujo pantalla **1210-1480**
Rutina felicita puntaje mayor 90 puntos **1490-1650**
Rutina suma sonora de puntajes **1660-1710**
Rutina tabla de posiciones y menú de control principal **1720-2140**
Rutina de puesta de puntajes a cero **2150-2170**
Rutina de presentación **2180-2360**

```
10 !@@@@@@@@@@@@@@@@@@@@@@@@@@@@
20 !@ ****PARACAIDISTAS*** @
30 !@ P.D.FERRANTE-CAPITAL @
40 !@ PROGRAMACION POPULAR @
50 !@@@@@@@@@@@@@@@@@@@@@@@@@@@@
55 !
60 CALL CLEAR
70 CALL SCREEN(6)
80 CALL MAGNIFY(4)
90 ON WARNING NEXT
100 RANDOMIZE
110 GOSUB 960 ! DEF DE CARAC
120 GOSUB 2180 !RUT PRESENT.
130 GOSUB 1830 !RUTINA CTRL
140 GOSUB 1210 !DIBUJA PANT
150 REM **********************
    *AQUI EMPIEZA JUEGO*
    **********************
    ****
160 SAL=9
170 IF CANT>1 THEN DISPLAY A
T(1,1):"EMPIEZA EL GRAN TORN
EO !!!" :: GOTO 870
180 DISPLAY AT(1,1):"PUNTAJE
=00"
190 DISPLAY AT(1,20):"SALTOS
=9"
200 CALL HCHAR(2,3,95,28)::
IF CANT>1 THEN DISPLAY AT(23
,1):"JUEGA:";NOM$(JUG);R
PT$(CHR$(127),28-LEN(NOM$(JU
G))-6)
210 XPOS=INT(RND*210)+10
220 CALL SPRITE(#3,104,12,14
0,XPOS)! IMPRIME BASE
230 IF CANT>1 THEN N$=NOM$(J
UG)&" " ELSE N$=""
240 IF SAL=0 THEN DISPLAY AT
(1,1):TAB(-7*(N$=""));N$;"OB
TUVO";SU;"PUNTOS" :: CAL
L MOTION(#2,0,0):: GOTO 860
250 CAR=INT(RND*7)+39 ! ELIG
E VIENTO
260 FOR I=3 TO 7 STEP 2 !IMP
RIME BANDERAS
270 CALL VCHAR(15,I,45,7)::
CALL HCHAR(15,I-1,32)
280 CALL HCHAR(15,I+1*(CAR>4
1 AND CAR<45),CAR)
290 NEXT I
300 VEL=6*(CAR=42 OR CAR=39)
+4*(CAR=43 OR CAR=40)+2*(CAR
```

```
=44 OR CAR=41)
310 IF CAR<42 THEN VEL=-VEL
320 CALL SPRITE(#2,100,12,1,
1,0,15)!IMPRIME AVION
330 CALL KEY(1,K,S)
340 CALL SOUND(-100,110,30,1
10,30,1000,30,-4,0)
350 CALL POSITION(#2,YO,XO)
: IF XO>230 THEN CALL DELSPR
ITE(#2):: FHEL=1 :: SAL=
SAL-1 :: GOTO 400 !PERDIO SA
LTO
360 IF S=0 THEN 330
370 IF K<>18 THEN 330
380 IF K=18 THEN CALL SPRITE
(#1,96,16,YO+10,XO,10,12)::
V=1 !LANZA AL HOMBRE
390 SAL=SAL-1
400 DISPLAY AT(1,27):STR$(SA
L):: IF FHEL THEN FHEL=0 ::
GOTO 830
410 FOR FREC=2000 TO 1000 ST
EP -40
420 CALL SOUND(-200,FREC,2)
430 CALL JOYST(1,X,Y):: CALL
 KEY(1,K,S)
440 CALL POSITION(#1,YO,XO)
450 CALL PATTERN(#1,96)
460 IF YO>106 THEN 640 ! SE
ESTRELLA
470 V=V+1 :: CALL MOTION(#1,
V*2+15,15-V*2.5)
480 CALL PATTERN(#1,108)
490 CALL PATTERN(#1,96)
500 IF Y=0 AND X=0 AND S=0 T
HEN 510 ELSE 520
510 NEXT FREC
520 IF Y<>-4 AND K<>0 THEN 5
10
530 IF Y=-4 OR S THEN CALL P
ATTERN(#1,116):: ALT=YO !ABR
E PARACAIDAS
540 CALL MOTION(#1,5,VEL)
550 CALL SOUND(200,110,30,11
0,30,880,30,-8,2)
560 CALL JOYST(1,X,Y):: CALL
 KEY(1,K,S)
570 IF X=-4 XOR K=2 THEN CAL
L MOTION(#1,5,-6+VEL)! MUEVE
 IZQ
580 IF X=4 XOR K=3 THEN CALL
 MOTION(#1,5,6+VEL)! MUEVE D
ER
590 IF X=0 AND S=0 THEN CALL
 MOTION(#1,5,VEL)
600 CALL POSITION(#1,YO,XO)
610 IF YO>137 THEN CALL MOTI
ON(#1,0,0):: CALL PATTERN(#1
,96):: GOTO 770 !ATERRIZ
A
620 GOTO 560


630 REM ********************
    ****BAJA A ESTRELLARSE*
    ********************
***
640 CALL POSITION(#1,YO,XO)
650 CALL PATTERN(#1,108)
660 CALL PATTERN(#1,96)
670 CALL POSITION(#1,YO,XO)
680 IF YO>125 THEN CALL MOTI
ON(#1,0,0):: CALL PATTERN(#1
,112):: CALL LOCATE(#1,1
52,XO):: GOTO 700 !LLEGO ???
!!
690 GOTO 640
700 CALL SOUND(100,110,30,11
0,30,1000,30,-8,2)! PUUMM!!
710 SU=SU-4 :: SU=-SU*(SU>0)
:: T$="0"&STR$(SU):: DISPLAY
 AT(1,9)SIZE(3):SEG$(T$,
-2*(SU>9)-1*(SU<10),LEN(T$))
720 FOR DEL=1 TO 500
730 NEXT DEL
740 IF SU<0 THEN SU=0 :: DIS
PLAY AT(1,13)SIZE(3):"0"
750 CALL DELSPRITE(#1,#2)::
GOTO 210 ! RECOMIENZA ESTREL
LADO


760 REM ********************
********ATERRIZO ..PERO,EN L
A BASE???!!**************
**
770 CALL LOCATE(#1,138,XO)
780 CALL COINC(#1,138,XPOS,6
,K)
790 CALL POSITION(#1,YO,XO)
800 IF XO>XPOS THEN 820 ! ER
RO !!!
810 IF K THEN CALL SOUND(100
,700,0):: GOSUB 1670 :: GOTO
 830 ! ATERRIZO BIEN
820 CALL SOUND(90,110,30,110
,30,2500,30,-8,2)
830 FOR DEL=1 TO 500 :: NEXT
 DEL
840 CALL DELSPRITE(#1,#2)
850 GOTO 210 !RECOMIENZA
860 IF SU>=90 THEN GOSUB 150
0 !FELICITA
870 IF CANT>1 THEN PUN(JUG)=
SU :: JUG=JUG-1 !CAMBIA JUGA
DOR
880 IF JUG=0 THEN GOSUB 1730
 :: IF CP=1 THEN JUG=CANT+1
:: GOSUB 2160 :: GOTO 14
0 ELSE GOSUB 2160 :: GOTO 14
0
890 IF CANT=1 THEN DISPLAY A
T(2,1):" PARA JUGAR OPRIMA Z
 SINO N" ELSE DISPLAY AT
(2,1):"JUGAS VOS ";NOM$(JUG)
;"@PULSA Z"
900 CALL SOUND(150,-1,0)
910 CALL KEY(0,K,S)
920 IF S=0 THEN 910
930 IF K=90 THEN SAL=9 :: SU
=0 :: DISPLAY AT(6,12):"" ::
 GOTO 180 ! RESET
940 IF K=78 AND CANT=1 THEN
CALL CLEAR :: CALL DELSPRITE
(ALL):: FLAG,SU=0 :: GOT
0 130
950 GOTO 910


960 !********************
    ****DEF CARACTERES*****
    ********************
***
970 CALL CHAR(39,"E0F0FCFFFC
FCE080")!FUERTE DER
980 CALL CHAR(42,"03071F7F1F
0703")!FUERTE IZQ
990 CALL CHAR(40,"80E0F0F8FC
9E8280")!MEDIO DER
1000 CALL CHAR(43,"0003070F1
F3C2000")!MEDIO IZQ
1010 CALL CHAR(41,"80C0E0F0B
0B09090")!BAJO DER
1020 CALL CHAR(44,"000103070
6060404")!BAJO IZQ
1030 CALL CHAR(45,"808080808
0808080")!NO HAY VIENTO
1040 CALL CHAR(64,"000000000
0301020")
1050 CALL CHAR(87,"3C3C3C3C3
C3C3C3C")!TRONCO ARBOL
1060 CALL CHAR(94,"000000007
C")
1070 CALL CHAR(95,"00FF")!LI
NEA SUBRAYADO
1080 CALL CHAR(96,"000000000
0000000003050301010204000000
0000000000000C0A0C080804
020")!TIPO PARADO BRAZOS ARR
IBA
1090 CALL CHAR(100,"00000000
0000030040417970304000000000
00000000FC60F0C8C8F8F8F0
3008")! DEFINE AVION
1100 CALL CHAR(104,"00000000
0000000000000000001F000000
0000000000000000000000000
C0")!DEFINE BASE
1110 CALL CHAR(108,"00000000
0000000000030103050102040000
0000000000000C080C0A080
2020")!DEFINE TIPO BRAZOS AB
AJO
1120 CALL CHAR(112,"00000000
030D0301090E0000000000000000
0000C0B0C080907000")!TIP
O EN EL SUELO
1130 CALL CHAR(116,"00000001
07070F0F080B0503010102040000
0080E0E0F0F010D0A0C08080
4020")!TIPO PARACAIDAS ABIER
TO
1140 CALL CHAR(127,"FFFFFFFF
FFFFFFFF")!DEL SUELO
1150 CALL CHAR(128,"14413688
55495298")
1160 CALL CHAR(136,"0103070F
1F3F7FFF")!PUNTA PINO IZQ
1170 CALL CHAR(137,"80C0E0F0
F8FCFEFF")!PUNTA PINO DER
1180 CALL CHAR(138,"FFFFFFFF
FFFFFFFF")!MEDIO PINO
1190 CALL CHAR(139,"183C3C7E
7E7EFFFF")!PUNTA PINO ARRIBA
1200 RETURN


1210 REM********************
     *DIBUJA PANTALLA JUEGO*
     ********************
****
1220 FOR I=7 TO 14 :: CALL C
OLOR(I,6,6):: NEXT I
1230 FOR I=22 TO 24 ! PISO
1240 CALL HCHAR(I,1,127,32)
1250 NEXT I
1260 REM NUBES
1270 CALL HCHAR(8,4,128,7)
1280 CALL HCHAR(10,6,128,3)
1290 CALL HCHAR(9,5,128,5)
1300 CALL HCHAR(7,5,128,5)
1310 CALL HCHAR(6,6,128,3)
1320 CALL HCHAR(10,12,128,2)
1330 CALL HCHAR(11,11,128,4)
1340 CALL HCHAR(12,12,128,2)
1350 REM ARBOLES
1360 FOR I=L1 TO L2 STEP 4
1370 CALL VCHAR(15,I+1,138,4
)
1380 CALL VCHAR(15,I,136,4)
1390 CALL VCHAR(15,I+2,137,4
)
1400 CALL HCHAR(14,I+1,139)
1410 CALL VCHAR(19,I+1,87,3)
1420 NEXT I
1430 REM ASIGNA COLORES
1440 FOR I=2 TO 9 :: CALL CO
```

```
LOR(I,2,6):: NEXT I
1450 CALL COLOR(12,4,2)
1460 CALL COLOR(13,15,6)
1470 CALL COLOR(14,13,6)
1480 RETURN


1490 REM ********************
        *S/R FELIC. PTS>90*
        ***************
****
1500 FOR I=1 TO LEN("FELICIT
ACIONES")
1510 CALL SOUND(150,262,8)
1520 CALL SOUND(150,523,8,65
9,8,262,8)
1530 CALL SOUND(1,110,30)
1540 CALL SOUND(150,523,8,65
9,8,262,8)
1550 A$=SEG$("FELICITACIONES
",I,1)
1560 DISPLAY AT(6,I+12):A$
1570 I=I+1
1580 CALL SOUND(150,196,8)
1590 CALL SOUND(150,196,8,78
4,8,494,8)
1600 CALL SOUND(1,110,30)
1610 CALL SOUND(150,196,8,78
4,8,494,8)
1620 A$=SEG$("FELICITACIONES
",I,1)
1630 DISPLAY AT(6,I+12):A$
1640 NEXT I
1650 RETURN

1660 REM******S/R SUMA*******
     **SONORA DE PUNTAJES***
     ********************


1670 IF ALT>106 THEN ALT=106
1680 FOR I=1 TO INT(13.5/88*
(ALT-18))
1690 SU=SU+1 :: DISPLAY AT(1
,9-1*(SU<10))SIZE(3):STR$(SU
):: CALL SOUND(100,150,0
)
1700 NEXT I
1710 RETURN


1720 REM*******S/R***********
     ***TABLA POSICIONES****
     ********************
****
1730 CALL DELSPRITE(ALL):: D
ISPLAY AT(2,1):">Z PARA TABL
A DE POSICIONES"
1740 CALL KEY(0,K,S):: IF S=
0 OR K<>90 THEN 1740
1750 CALL CLEAR :: PRINT TAB
(3);"TORNEO DE PARACAIDISMO"
:RPT$("_",28):TAB(5);"TA
BLA DE POSICIONES": : : : :
: : : : : : : : : : : : : : :
: :
1760 FOR I=1 TO CANT-1 :: FO
R J=I+1 TO CANT
1770 IF PUN(I)>PUN(J)THEN T=
PUN(I):: PUN(I)=PUN(J):: PUN
(J)=T :: T$=NOM$(I):: NO
M$(I)=NOM$(J):: NOM$(J)=T$
1780 NEXT J :: NEXT I
1790 FOR I=CANT TO 1 STEP -1
 :: DISPLAY AT(4+2*(CANT-I+1
),2):STR$(CANT-I+1)&"> "
;NOM$(I);" CON";PUN(I);"PUNT
OS" :: NEXT I
1800 DISPLAY AT(23,1):">OPRI
MAN CUALQUIER TECLA"
1810 CALL KEY(0,K,S):: IF S=
0 THEN 1810


1820 REM******RUTINA DE*****
     ****CONTROL PRINC***
     ***************
****
1830 DISPLAY ERASE ALL AT(2,
1):"OPRIMAN :":"_______": :"
1^PARA CONTINUAR EL JUEG
O": :"2^PARA CAMBIAR JUGADOR
ES": :"3^PARA INSTRUCCIONES"
1840 DISPLAY AT(11,1):"4^PAR
A TERMINAR"
1850 CALL KEY(0,CP,S):: IF S
=0 OR CP-48<1 OR CP-48>4 THE
N 1850
1860 CALL SOUND(80,500,0)
1870 CP=CP-48
1880 IF CP=4 THEN CALL CLEAR
 :: STOP
1890 IF CP=3 THEN CALL CLEAR
 :: GOTO 1920
1900 IF CP=1 AND FLAG THEN 2
140 ELSE 2070
1910 IF CP=2 THEN 2080
1920 PRINT "GRAN TORNEO DE P
ARACAIDISMO":RPT$("_",27): :
"ESTE JUEGO CONSISTE EN"
:"ATERRIZAR UN PARACAIDISTA"
:"EN UNA BASE SOBRE EL SUELO
"
1930 PRINT "DEFINIDA POR UNA
 LINEA":"AMARILLA. EL PARACA
IDISTA SE":"ARROJA DESDE
 UN HELICOPTERO.":"ESTE HELI
COPTERO PASA UNA":"SOLA VEZ
POR SALTO."
1940 PRINT "POR LO TANTO SI
NO SE ARROJA":"AL PARACAIDIS
TA Y EL HELICOP":"TERO P
ASA EL BORDE DERECHO":"DE LA
 PANTALLA@ UD. ENTONCES"
1950 PRINT "PERDERA DICHO SA
LTO."
1960 PRINT "PARA ARROJAR AL
PARACAIDISTA":"HAGALO OPRIMI
ENDO ""Q"" O EL":"BOTON
DEL CONTROL REMOTO 1.":"PARA
 ABRIR EL PARACAIDAS"
1970 PRINT "OPRIMA ""X"" O I
MPULSE HACIA":"ABAJO LA PALA
NCA DEL CONTROL"
1980 CALL KEY(0,K,S):: IF S=
0 THEN 1980
1990 PRINT :"EL DESCENSO ES
AFECTADO":"POR EL VIENTO QUE
 CAMBIA":"EN CADA SALTO
Y PUEDE SER":"FUERTE@MEDIANO
@BAJO O NULO."
2000 PRINT "PARA HACER FRENT
E AL VIENTO":"EL EQUIPO POSE
E COHETES QUE":"CONTRARR
ESTAN LA ACCION DEL":"MISMO.
 PARA ACCIONARLOS A LA"
2010 PRINT "DERECHA@OPRIMA "
"D"" O IMPULSE":"LA PALANCA
DEL CONTROL":"A LA DEREC
HA.PARA HACERLO":"A LA IZQUI
ERDA OPRIMA ""S"" O IMPULSE"
;
2020 PRINT " LA PALANCA DEL
CON^":"TROL HACIA LA IZQUIER
DA.":"EL PUNTAJE OBTENID
O DEPENDE":"DE LA ALTURA A Q
UE SE ABRIO":"EL PARACAIDAS.
¡MAX=11 PTS.¿"
2030 PRINT "CUANTO MAS ARRIB
A MENOS PUN^":"TAJE.EL PARAC
AIDAS SE ABRE  HASTA CIE
RTA ALTURA@SI NO LO":"ABRE E
NTONCES EL PARACAIDIS^"
2040 PRINT "TA SE ESTRELLA Y
 SE RESTAN":"CUATRO PUNTOS.B
UENA SUERTE!"
2050 CALL KEY(0,K,S):: IF S=
0 THEN 2050
2060 GOTO 1830
2070 GOSUB 2160
2080 CALL CLEAR
2090 DISPLAY AT(1,1):"CUANTO
S JUEGAN?¡1 A 9¿" :: ACCEPT
AT(1,24)VALIDATE("123456
789")SIZE(1):JUG :: CANT=JUG
2100 IF JUG=1 THEN 2120
2110 FOR I=JUG TO 1 STEP -1
:: DISPLAY AT((CANT-I+2)*2,2
)BEEP:"JUGADOR NUM "&STR
$(CANT-I+1)&":" :: ACCEPT AT
((CANT-I+2)*2,16)SIZE(9):NOM
$(I):: NEXT I
2120 JUG=JUG+1
2130 FLAG=-1
2140 CALL CLEAR :: RETURN


2150 REM *****S/R***********
     ****PUESTA A CERO PUN**
     ******************
****
2160 FOR I=1 TO 9 :: PUN(I)=
0 :: NEXT I :: SU=0
2170 RETURN


2180 REM*******************
********RUTINA PRESENTACION*
***********************
****
2190 FOR I=7 TO 14 :: CALL C
OLOR(I,6,6):: NEXT I
2200 FOR I=22 TO 24 :: CALL
HCHAR(I,4,127,26):: NEXT I
2210 L1=5 :: L2=10 :: GOSUB
1350
2220 DISPLAY AT(1,2):"P A R
A C A I D I S T A S":" "&RPT
$("_",25)
2230 DISPLAY AT(6,3)BEEP:"ES
 UN PROGRAMA DE:"
2240 DISPLAY AT(7,3)BEEP:RPT
$("_",17)
2250 DISPLAY AT(8,5):"PABLO
DANIEL FERRANTE"
2255 DISPLAY AT(10,3):"PARA
PROGRAMACION POPULAR"
2260 CALL SPRITE(#1,116,16,8
0,140,10,0)
2270 CALL SPRITE(#2,100,12,1
,120)
2280 FOR I=1 TO 330 :: NEXT
I
2290 CALL MOTION(#1,0,0)
2300 CALL PATTERN(#1,96)
2310 FOR I=1 TO 11 :: READ K
,S,X,Y :: CALL SOUND(K,S,3,X
,3,Y,0):: NEXT I
2320 DATA 150,523,659,262,15
0,523,659,262,150,523,659,26
2,500,523,659,262,170,44
000,44000,196,150,523,659,26
2
2330 DATA 150,523,659,262,15
0,587,698,196,150,523,659,19
6,150,587,698,196,500,52
3,784,262
2340 CALL DELSPRITE(ALL)
2350 L1=17 :: L2=29
2360 RETURN
```



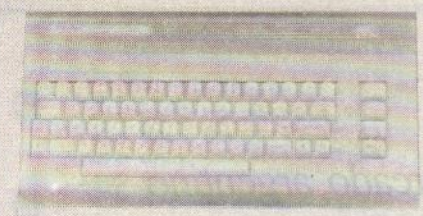
Galería de retratos

...de las principales microcomputadoras del mercado argentino

Intentamos en estas páginas responder a múltiples pedidos de los amigos de Programación Popular: Están los que quieren saber cuáles y cuántas computadoras existen en el mercado argentino, y qué características tienen...
Están los que saben todas esas cosas pero aún no se deciden por una marca o modelo en especial...
Y están los que ya se habían decidido pero de todos modos piensan que una comparación siempre es ilustrativa.
Pensando en todos ellos nos ocupamos en esta nota de los que –aquí y ahora– son los principales microcomputadores en plaza. Esperamos que este conjunto de mini-fichas técnicas les sea de utilidad.

Referencias:

- *Todas las capacidades de memoria se dan en Kb (Kilobytes).*
Recordar que un byte es capaz de almacenar un carácter de información, y que 1 Kb = 1024 bytes.
- *La "memoria usuario" equivale a la memoria libre efectivamente disponible (versión base) para programas en BASIC y datos, una vez "desplegado" en la RAM el Sistema Operativo / Basic.*
- *Las velocidades de grabación de programas y datos en cassette se dan en bauds (bits/segundo).*
- *Al referirnos al teclado de que vienen provistas las consolas consignamos el material de fabricación (de membrana, de goma o de plástico) teniendo en cuenta que la facilidad de tipeo varía mucho desde el teclado "de membrana" (poco manuable) hasta el de plástico rígido (el más cómodo).*
Por "normalizado" entendemos el teclado de tipo "Qwerty", con barra espaciadora independiente.
- *Los colores, obviamente, serán tonalidades de gris en un TV blanco y negro. Todas las computadoras analizadas se conectan directamente a un televisor casero, b/n o color. En algunos casos (T/S 2068 importada) es necesario un TV bajo norma NTSC o binorma para apreciar los colores "reales" con que se trabaja.*
- *La especificación "Controlador de ADSR" indica qué computadoras permiten programar la forma de envolvente del sonido generado, dando lugar a efectos sonoros.*
- *Sprites, como puede no saberlo todo el mundo, son elementos gráficos de forma y movimiento programables, en alta resolución, de aplicación –sobre todo– en juegos. Ejemplo de un sprite famoso: Pacman.*
- *Los periféricos mencionados no siempre cubren la totalidad de los existentes para cada computadora. Además se trata de un rubro en expansión permanente.*
Ningún periférico de los nombrados aquí se incluye en el precio de venta de la computadora.
Todas las computadoras analizadas son capaces de grabar y leer información en cassettes comunes, con grabadores comunes.
- *Compatibilidad entre computadoras:*
(1) TK-83, TK-85, CZ 1000 y CZ 1500 son totalmente compatibles en BASIC, con la salvedad de que CZ 1000 y TK-83 tienen menor capacidad de memoria RAM. Luego, algunos programas para CZ 1500 o TK-85 no correrán en CZ 1000 ó TK-83 sin expansión de memoria.
La TK-85 es ligeramente distinta de las demás recién mencionadas. Algunos programas escritos en las otras, que hagan uso de rutinas en la ROM pueden no correr en TK-85, y viceversa.
La diferencia consiste en la implementación en ROM de rutinas de transferencia rápida de dastos a y desde cassette. Las demás máquinas nombradas no recuperarán datos grabados en este modo.
(2) CZ 2000 Spectrum y TK-90 X (48 K) son máquinas muy similares, compatibles un 100% en BASIC (algunas rutinas en assembler pueden no ser intercambiables, especialmente si utilizan rutinas de la ROM).
La T/S 2068 presenta diferencias algo más notables con las nombradas. En principio las tres aceptan información que haya sido grabada por las otras, aunque excepcionalmente será necesaria una cierta (y generalmente simple) adaptación del programa BASIC a correr, desde o hacia la T/S 2068. El mapa de memoria es distinto en la T/S 2068, por lo cual los programas en assembler para la una no correrán, como regla general, en las otras.
- *Los precios indicados para cada computadora son los de lista, proporcionados por sus fabricantes y correspondientes a fines de enero de 1986.*



Ordenador: Commodore 16
Fabricante: Drean S.A.
Memoria ROM: 32 Kb
Memoria RAM: 16 Kb
Memoria Usuario: 12 Kb
Basic que utiliza: Commodore Basic versión 3.5
CPU: MOS7501 (de 8 bits)
Frecuencia de operación: 1,76 MHz
Velocidad cassette: 600 bauds
Teclado: de plástico
Normalizado: Sí
Teclas: 66 **Caracteres gráficos:** 64
Funciones: 26 **Colores:** 16
Comandos: 75
Caracteres alfanuméricos: 64

Pantalla modo texto: 40 x 25 caracteres
Pantalla modo gráfico: 320 x 160 pixels
Sonido: Sí, tres voces
Controlador de ADSR: Sí
Maneja Sprites: No
Periféricos: Diskette 5 ¼, dos joysticks, modem, impresora térmica de conexión directa.
Compatible con: ninguna otra computadora del mercado argentino
Otros lenguajes disponibles: Assembler.
Precio: ≠ 240



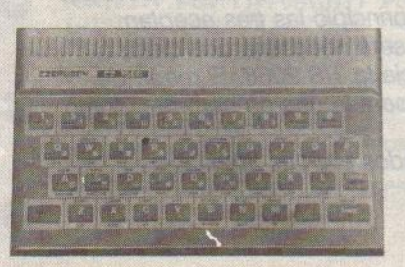
Ordenador: Commodore 64
Fabricante: Drean S.A.
Memoria ROM: 20 Kb
Memoria RAM: 64 Kb
Expandible a: 128 Kb
Memoria Usuario: 39 Kb
Basic que utiliza: Commodore Basic versión 2.0
CPU: 6510 (de 8 bits)
Frecuencia de operación: 1 MHz
Velocidad cassette: 600 bauds
Teclado: de plástico rígido **Comandos:** 38
Normalizado: Sí **Caracteres alfanuméricos:** 64
Teclas: 66 **Caracteres gráficos:** 64
Funciones: 26 **Colores:** 16

Pantalla modo texto: 40 x 25 caracteres
Pantalla modo gráfico: 320 x 200 pixels
Sonido: Sí, tres voces
Controlador de ADSR: Sí
Maneja Sprites: Sí
Periféricos: Diskette 5 ¼, impresora térmica de conexión directa, lápiz óptico, dos joysticks, modem, sintetizador de voz, conexión a interface serie, cartridges.
Compatible con: ninguna otra computadora del mercado argentino
Otros lenguajes disponibles: Logo, Assembler, Forth, Pilot
Precio: ≠ 385

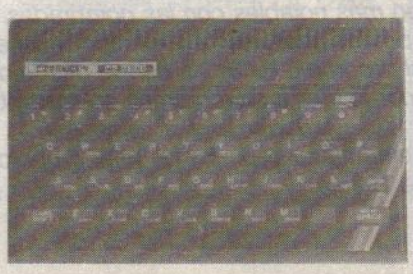


Ordenador: CZ 1000
Fabricante: CZERWENY S.A.
Memoria ROM: 8 Kb
Memoria RAM: 2 Kb
Expandible a: 16 Kb
Memoria Usuario: 2 Kb
Basic que utiliza: Sinclair Basic
CPU: Z80A (de 8 bits)
Frecuencia de operación: 3.25 MHz
Velocidad cassette: 300 bauds
Teclado: de goma
Normalizado: No
Teclas: 40
Funciones: 37
Comandos: 34
Caracteres alfanuméricos: 32
Caracteres gráficos: 22
Colores: blanco y negro
Pantalla modo texto: 32 x 22 caracteres
Pantalla modo gráfico: 64 x 44 pixels
Sonido: No
Maneja Sprites: No
Periféricos: Impresora térmica de conexión directa, expansión de memoria RAM a 16 Kb
Compatible con: CZ 1500, TK-83, TK-85 (1)
Otros lenguajes disponibles: Mini-Logo, Assembler
Precio: ≠ 69

Ordenador: CZ 1500
Fabricante: CZERWENY S.A.
Memoria ROM: 8 Kb
Memoria RAM: 16 Kb
Expandible a: 48 Kb
Memoria Usuario: 16 Kb
Basic que utiliza: Sinclair Basic
CPU: Z80A (de 8 bits)
Frecuencia de operación: 3.25 MHz
Velocidad cassette: 300 bauds
Teclado: de goma
Normalizado: No
Teclas: 40
Funciones: 37
Comandos: 34
Caracteres alfanuméricos: 32
Caracteres gráficos: 22
Colores: Blanco y negro
Pantalla modo texto: 32 x 22 caracteres
Pantalla modo gráfico: 64 x 44 pixels
Sonido: No
Maneja Sprites: No
Periféricos: Impresora térmica de conexión directa, expansión de memoria RAM, programador de EPROM.
Compatible con: CZ1000, TK-83, TK-85 (1)
Otros lenguajes disponibles: Mini-Logo, Assembler
Precio: ≠ 138

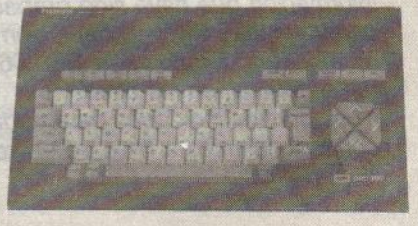


Ordenador: CZ 2000 Spectrum
Fabricante: CZERWENY S.A.
Memoria ROM: 16 Kb
Memoria RAM: 48 Kb
Memoria Usuario: 41 Kb
Basic que utiliza: Sinclair Basic Extendido
CPU: Z80A (de 8 bits)
Frecuencia de operación: 3.5 MHz
Velocidad cassette: 1500 bauds
Teclado: de goma
Normalizado: No **Funciones:** 35
Teclas: 40 **Comandos:** 48
Caracteres alfanuméricos: 32
Caracteres gráficos: 16
Colores: 8
Pantalla modo texto: 32 x 22 caracteres
Pantalla modo gráfico: 256 x 192 pixels
Sonido: Sí
Controlador de ADSR: No
Maneja Sprites: No
Periféricos: Impresora térmica de conexión directa. Microdrive Sinclair, diskette de 3 ½", lápiz óptico, joystick, sintetizador de voz.
Compatible con: TK-90 X y TS 2068 (2)
Otros lenguajes disponibles: Logo, Forth, Assembler, Pascal, Lisp
Precio: ≠ 249



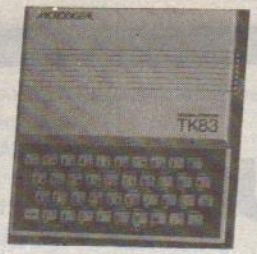
Ordenador: Talent-MSX DPC 200
Fabricante: Telemática S.A.
Memoria ROM: 32 Kb
Memoria RAM: 64 Kb
Expandible a: 576 Kb
Memoria Usuario: 29 Kb
Basic que utiliza: MSX Microsoft Basic
CPU: Z80A (de 8 bits)
Frecuencia de operación: 3.6 MHz
Velocidad cassette: 1200 bauds o 2400 bauds
Teclado: de plástico **Funciones:** 62
Normalizado: No **Comandos:** 94
Teclas: 77 **Colores:** 16
Caracteres alfanuméricos: 160
Caracteres gráficos: 64

Pantalla modo texto: 40 x 24 caracteres
Pantalla modo gráfico: 256 x 192 pixels
Sonido: Sí, tres voces **Maneja Sprites:** Sí
Controlador de ADSR: Sí
Periféricos: Diskette 5 ¼", QuickDisk 2.8", impresora tipo Centronics, interface serie, modem, lápiz óptico, sintetizador de voz, dos joysticks, cartridges de expansión de memoria y de software, tableta gráfica.
Compatible con: Todas otras bajo el standard MSX y S.O. MS/DOS (IBM, etc.)
Otros lenguajes disponibles: Logo, Assembler, Cobol, Pascal, Fortran, C
Precio: ≠ 495





Ordenador: TI-99/4A
Fabricante: Texas Instruments
Memoria ROM: 28 Kb
Memoria RAM: 16 Kb
Expandible a: 128 Kb
Memoria Usuario: 14 Kb
Basic que utiliza: TI Basic
CPU: TMS9900 (de 16 bits)
Frecuencia de operación: 1 MHz
Velocidad cassette: 600 bauds
Teclado: de plástico **Comandos:** 49
Normalizado: Sí **Caracteres alfanuméricos:** 94
Teclas: 48 **Caracteres gráficos:** 32
Funciones: 19 **Colores:** 16
Pantalla modo texto: 28 x 24 caracteres
Pantalla modo gráfico: 256 x 192 pixels
Sonido: Sí, tres voces
Controlador de ADSR: No
Maneja Sprites: Sí, con módulo de Basic Extendido
Periféricos: TI-Extended Basic y otros cartridges de software, diskette 5 ¼", disco 5 Mb y 10 Mb, tableta gráfica, dos joysticks, interfaces serie y paralelo, expansión de memoria, caja de periféricos.
Compatible con: Ninguna otra computadora del mercado argentino
Otros lenguajes disponibles: TI-Logo, TI-Pilot, tI-Pascal, Assembler
Precio: ≠ 350



Ordenador: TK-83
Fabricante: Microdigital Arvoc S.A.
Memoria ROM: 8 Kb
Memoria RAM: 2 Kb
Expandible a: 16 Kb
Memoria Usuario: 2 Kb
Basic que utiliza: Sinclair Basic
CPU: Z80 (de 8 bits)
Frecuencia de operación: 3.25 MHz
Velocidad cassette: 300 bauds
Teclado: de membrana **Normalizado:** No
Teclas: 40
Funciones: 37
Comandos: 34
Caracteres alfanuméricos: 32
Caracteres gráficos: 22
Colores: blanco y negro
Pantalla modo texto: 32 x 22 caracteres
Pantalla modo gráfico: 64 x 44 pixels
Sonido: No
Maneja Sprites: No
Periféricos: Impresora térmica de conexión directa, un joystick, admite conexión a interfaces serie y paralelo, expansión de memoria RAM a 16 Kb.
Compatible con: CZ-1000, CZ 1500, TK-85 (1)
Otros lenguajes disponibles: Mini-Logo, Forth, Assembler
Precio: ≠ 90



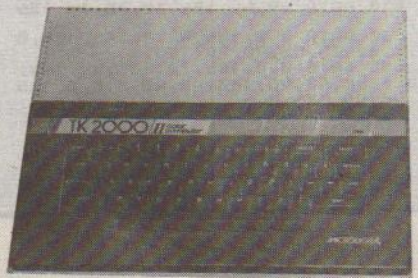
Ordenador: TK-85 (modelos de 16 y 48 Kb)
Fabricante: Microdigital Arvoc
Memoria ROM: 10 Kb
Memoria RAM: 16 Kb o 48 Kb
Expandible a: 48 Kb (a 64 Kb el modelo de 48 Kb)
Memoria Usuario: 16 Kb o 48 Kb según modelo
Basic que utiliza: Sinclair Basic
CPU: Z80A (de 8 bits)
Frecuencia de operación: 3.25 MHz
Velocidad cassette: 300 bauds y 4200 bauds
Teclado: de goma
Normalizado: No
Teclas: 40
Funciones: 37
Comandos: 34
Caracteres alfanuméricos: 32
Caracteres gráficos: 22
Colores: blanco y negro
Pantalla modo texto: 32 x 22 caracteres
Pantalla modo gráfico: 64 x 44 pixels
Sonido: No
Maneja Sprites: No
Periféricos: Expansor de memoria RAM. Impresora térmica de conexión directa, un joystick, interfaces serie y paralelo, programador EPROM
Compatible con: TK-83, CZ 1000, CZ 1500 (1)
Otros lenguajes disponibles: Mini-Logo, Assembler
Precio: ≠ 120 (modelo 16 Kb) y ≠ 169 (modelo 48 Kb)

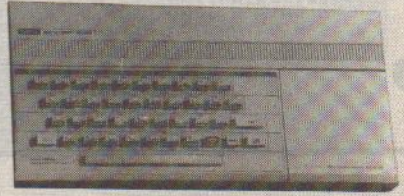
Ordenador: TK-90 X (modelos de 16 y 48 Kb)
Fabricante: Microdigital Arvoc S.A.
Memoria ROM: 16 Kb
Memoria RAM: 16 Kb o 48 Kb
Expandible a: 48 Kb (el modelo de 16 Kb RAM)
Memoria Usuario: 12 Kb o 41 Kb
Basic que utiliza: Sinclair Basic Extendido
CPU: Z80A (de 8 bits)
Frecuencia de operación: 3.5 MHz
Velocidad cassette: 1500 bauds
Teclado: de goma **Funciones:** 35
Normalizado: No **Comandos:** 60
Teclas: 40 **Colores:** 8
Caracteres alfanuméricos: 32
Caracteres gráficos: 16
Pantalla modo texto: 32 x 22 caracteres
Pantalla modo gráfico: 256 x 192 pixels
Sonido: Sí, por TV
Controlador de ADSR: No
Maneja Sprites: No
Periféricos: Impresora térmica de conexión directa, dos joysticks, lápiz óptico. Microdrive Sinclair, diskette de 3", 3 ½", 5 ¼", sintetizador de voz, interfaces serie y paralelo, expansión de 32 Kb para el modelo de 16 Kb.
Compatible con: CZ 2000 Spectrum, TS 2068 (2)
Otros lenguajes disponibles: Logo, Assembler, Pascal, Prolog
Precio: ≠ 190 (modelo de 16 Kb) y ≠ 230 (modelo de 48 Kb)

Ordenador: TK-2000 (modelos de 64 y 128 Kb)
Fabricante: Microdigital Arvoc S.A.
Memoria ROM: 16 Kb
Memoria RAM: 64 Kb o 128 Kb
Expandible a: 128 Kb (el modelo de 64 Kb)
Basic que utiliza: Applesoft Basic
CPU: 6502 (de 8 bits)
Frecuencia de operación: 1.02 MHz
Teclado: de plástico
Normalizado: Sí
Teclas: 54
Funciones: 46
Comandos: 65
Caracteres alfanuméricos: 59
Caracteres gráficos: 49
Colores: 6
Pantalla modo texto: 40 x 24 caracteres
Pantalla modo gráfico: 280 x 192 pixels
Sonido: Sí, por TV
Controlador de ADSR: No
Maneja Sprites: No
Periféricos: Impresora de conexión directa, interface serie, joystick, sintetizador de voz, diskette 5 ¼".
Compatible con: Otras computadoras, bajo el standard Applesoft
Otros lenguajes disponibles: Assembler (incorporado en ROM)
Precio: ≠ 325 (modelo 64 Kb) y ≠ 390 (modelo de 128 Kb)

Ordenador: TS 2068
Fabricante: Timex Sinclair
Memoria ROM: 16 Kb
Memoria RAM: 72 Kb
Memoria Usuario: +38 Kb
Basic que utiliza: Sinclair Basic Extendido
CPU: Z80A (de 8 bits)
Frecuencia de operación: 3.5 MHz
Velocidad cassette: 1500 bauds
Teclado: de plástico
Normalizado: Sí
Teclas: 42
Funciones: 35
Comandos: 79
Caracteres alfanuméricos: 32
Caracteres gráficos: 16
Colores: 8
Pantalla modo texto: 32 x 22 caracteres
Pantalla modo gráfico: 256 x 192 pixels
Sonido: Sí, tres voces
Controlador de ADSR: Sí
Maneja Sprites: No
Periféricos: Impresora térmica de conexión directa, interfaces serie y paralelo, cartridges, dos joysticks, lápiz óptico.
Compatible con: CZ 2000, TK-90 X (2)
Otros lenguajes disponibles: Logo, Assembler, Pascal
Precio: ≠ 330



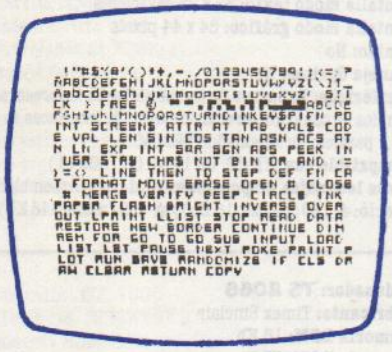




Nuevo juego de caracteres



Es este otro excelente programa ganador del Concurso que nos enviara el amigo **Diego**, el que nos permitirá ampliar las prestaciones de nuestras **Sinclair** ''grandes'' y disponer, en forma optativa, de un juego adicional de caracteres el cual, según su autor, resulta más ''vistoso'', lo que hemos podido corroborar, como seguramente lo harán ustedes.



De esta forma podremos mejorar la presentación de los programas, pantallas y listados, a través de la utilización del nuevo juego de caracteres, el que se agrega al ya disponible en la **ROM** de nuestra máquina, de modo que podremos utilizar ambos, según lo creamos conveniente. También nos agrega este programa las ''eñes'' (mayúscula y minúscula) y el símbolo de uno de los más importantes personajes del recién finalizado 1985: el **AUSTRAL** (que no pocos dolores de cabeza y sofocones ha provocado entre la familia de los programadores, principalmente del área administrativa.

ESQUEMA DE FUNCIONAMIENTO

Cada vez que nuestra computadora necesita imprimir algo primero se fija en el valor de las variables del sistema denominada **CHARS**, para posicionar el inicio de la codificación binaria de los símbolos asignados a cada uno de los caracteres de nuestra máquina.

Podremos verificar esto enseguida si examinamos el contenido de dicha variable **CHARS**, haciendo

PRINT PEEK 23606, PEEK 23607

y obtendremos **0** y **60**, lo cual nos indica (ver Manual) que la codificación binaria de los caracteres a imprimir comienza en la siguiente dirección

15616 = 0 + 60*256 + 256

Sabiendo esto será sencillo entender el esquema del funcionamiento de la propuesta de nuestro amigo Diego: almacenar ''arriba'' de la **RAMTOP** (para máquinas de **48 K RAM**), a partir de la dirección **64000** para protegerla de borrados –accidentales o no–,

LISTADO 1

```
   1 CLEAR 63999: LET t$=          "
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A  NUEVO JUEGO DE CARACTERES    A
A  Diego Rodriguez-Longchamps   A
A Rev.Programacion Popular-Nº 9 A
A CZSpectrum-TK90X-TS2068    16K A
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
"
   2 BORDER 0: PAPER 0: INK 9: C
LS : PRINT PAPER 1;t$
   5 LET a$="Diego Rodriguez pre
senta LETRAS ": FOR f=1 TO LEN a
$: LET b$=a$( TO f): PRINT AT 7,
31-LEN b$;b$: BEEP .01,f: BEEP .
01,f/3: NEXT f
   7 PAUSE 20: PRINT AT 7,0; INV
ERSE 1;a$;AT 21,1; FLASH 1;" Esp
ere un momento ,por favor ": LET
 k$="0000"
  10 LET ram=64000: LET l=NOT PI
: FOR f=ram TO 64767: LET k=767-
(f-ram): PRINT AT 11,14;k$( TO 4
-LEN STR$ k)+STR$ k: NEXT f: LET
 l=l+a: POKE f,a: NEXT f: IF l<>
39642 THEN PRINT FLASH 1;"error
en datas": STOP
  20 DATA 0,0,0,0,0,0,0,0,0,0,48,4
8,48,0,16,0
  30 DATA 0,102,102,102,0,0,0,0,
0,100,254,100,100,254,100,0
  40 DATA 16,126,80,126,18,126,1
26,16,1,98,100,8,16,38,70,128,0,
62,34,127,34,127,34,0
  50 DATA 0,24,24,48,0,0,0,0
  60 DATA 0,24,32,64,64,64,32,24
,0,24,4,2,2,2,4,24
  70 DATA 0,20,8,62,8,20,0,0,0,0,2
4,24,126,126,24,24,0,0,0,0,0,0,0,
24,24,48,0,0,0,126,126,0,0,0
  80 DATA 0,0,0,0,0,0,0,24,24,0,3,
6,12,24,48,96,0,0,126,70,74,82,4
8,126,0,0,60,76,12,12,12,12,0,0,
126,2,126,64,126,126,0
  90 DATA 0,126,2,126,2,126,126,
0,0,48,48,48,126,2,2,0,0,126,126
,64,126,2,126,0,0,112,64,126,70,
70,126,0,0,126,6,12,24,48,48,0,0
,126,66,126,66,126,126,0,0,126,6
0,98,126,2,2,0,0,24,24,0,0,24,24
,0
 100 DATA 0,0,24,24,0,24,24,48,0
,0,4,8,16,8,4,0,0,0,126,0,0,126,
0,0
 110 DATA 0,0,64,32,16,32,64,0,0
,126,2,62,32,0,32,0,60,0,114,106
,102,98,98,0,0,126,48,48,126,48,
48,0,0,124,66,124,66,124,124,0,0
,126,96,96,96,96,126,0,0,120,100
,98,98,100,120,0
 120 DATA 0,126,64,126,64,126,12
6,0,0,126,96,96,120,96,96,0,0,12
6,96,96,110,98,126,0,0,98,98,126
,98,98,98,0,0,24,24,24,24,24,24,
0,0,6,6,6,6,70,60,0,0,76,88,112,
88,76,70,0
 130 DATA 0,96,96,96,96,96,126,0
,0,97,115,109,97,97,97,0,0,98,11
4,106,102,98,98,0,0,126,98,98,98
,98,126,0,0,126,98,126,96,96,96,
0,0,126,98,106,102,126,1,0
 140 DATA 0,126,98,126,100,98,98
,0,0,126,64,126,2,126,126,0,0,12
6,24,24,24,24,24,0,0,98,98,98,98
,98,126,0,0,98,98,98,52,52,24,0,
0,97,97,97,109,115,97,0
 150 DATA 0,66,100,66,66,100,66,
0,0,98,98,98,126,24,24,0,0,126,6
,12,24,48,126,0
 160 DATA 0,60,48,48,48,48,60,0,
192,96,48,24,12,6,3,0,0,60,12,12
,12,12,60,0,24,60,90,153,24,24,2
4,0
 170 DATA 0,0,0,0,0,0,255,255,0,
60,0,126,98,98,98,0
 200 DATA 0,0,126,2,126,98,126,0
 210 DATA 64,64,126,70,70,70,126
,0
 220 DATA 0,0,126,96,96,96,126,0
,2,2,126,98,98,98,126,0
 230 DATA 0,0,126,70,126,64,126,
0,0,60,48,48,120,48,48,0,0,0,126
,98,98,126,2,62
 240 DATA 64,64,126,70,70,70,70,
0,0,8,0,24,24,24,24,0
 250 DATA 0,4,0,6,6,6,70,56,0,0,
100,104,112,104,100,0
 260 DATA 0,32,32,32,32,56,56,0,
0,0,126,106,106,106,106,0,0,0,12
6,98,98,98,98,0,0,0,126,98,98,98
,126,0,0,0,126,70,70,126,64,64
 270 DATA 0,0,126,98,98,126,2,2,
0,0,60,48,48,48,48,0,0,0,60,32,6
0,4,60,0,0,16,56,16,16,28,28,0,0
,0,98,98,98,98,126,0
 280 DATA 0,0,102,102,102,36,24,
0
 290 DATA 0,0,98,106,106,106,126
,0
 300 DATA 0,0,102,102,24,24,102,
0,0,0,98,98,126,2,30,0
 310 DATA 0,0,126,6,24,96,126,0
 320 DATA 0,16,32,32,96,32,32,16
,24,24,24,24,24,24,24,24,0,8,4,4
,8,4,4,8
 330 DATA 204,204,51,51,0,0,0,0,
60,66,153,161,161,153,66,60
 340 POKE 23607,249
 350 BEEP .5,10: CLS : PRINT ;TA
B 3; PAPER 1;" Nuevo juego de ca
racteres ";''
 400 FOR f=32 TO 128: PRINT PAPE
R 1;CHR$ f;" ";: NEXT f
 410 PRINT '''"En este nuevo jueg
o se incluyen las eñes Mayuscula
s, minusculas y el signo Austral
"'"  Ñ,ñ >> [SYMBOL SHIFT] + [2]"
;"  Ñ,ñ >> [SYMBOL SHIFT] + [X]"'"
  A >> [SYMBOL SHIFT] + [6]"
 420 PRINT '"Para volver al jueg
o de caracte-res normal teclear
POKE 23607,60y para el ""nuevo""
 POKE 23607,249"
 500 PRINT AT 21,0;TAB 6; FLASH
1;" Pulse alguna tecla ": PAUSE
0: PRINT AT 20,0;: POKE 23692,25
5: FOR f=0 TO 23: PRINT : BEEP .
01,f: NEXT f: STOP
```

incluso con **NEW**) un bloque compuesto por **768 bytes** que contienen la información de los nuevos caracteres (tal como la dispone nuestra computadora para los caracteres llamémoslos "normales") y luego **POKE**'ar la referida variable del sistema **CHARS** con la nueva dirección inicial (**–256**, según lo indicado en el Manual) del flamante juego de caracteres.

O sea, que deberemos posicionar en la variable **CHARS** la dirección **64000**, haciendo

POKE 23607,249

De esta forma la máquina sabrá que la dirección inicial es **64000** porque

64000=0 + 249*256 + 256

PROCEDIMIENTO PARA CARGAR EL LM Y OPERARLO

Teclear con cuidado el programa del **Listado 1** (líneas **1** a **500**), hacer **RUN** y esperar que el contador del centro de la pantalla llegue a cero.

Si nos equivocamos al cargar los **DATA**'s, el programa nos avisará que hay un error. Tendremos que chequear todo nuevamente, corregir el error y reiniciar la carga.

Si todo anduvo bien, veremos una pequeña demostración con los nuevos caracteres, luego de lo cual convendrá grabar de inmediato el programa y los bytes haciendo, para este último caso

SAVE "CARACTERES" CODE 64000,768

En lo sucesivo sólo deberemos cargar los bytes, haciendo

CLEAR 63999 LOAD ""CODE

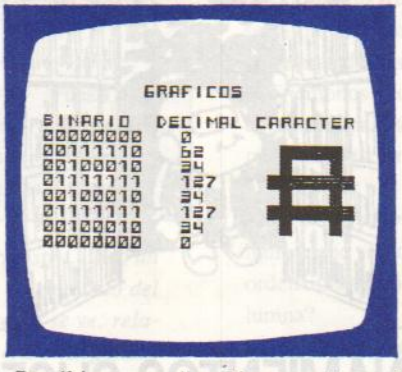
Para cambiar los caracteres al **Modo "normal"** tan sólo debemos hacer

POKE 23607,60

Y podremos reingresar a nuestro juego haciendo

POKE 23607,249

Podremos ver los nuevos caracteres ampliados; ocho veces si utilizamos el programa **"CALCULADORA"** publicado en el número 9 de **P.P.**, que nos enviara también el amigo **Diego** y que resultara asimismo ganador del Concurso.



Por último, nos dice **Diego**, un "truco" al respecto. Hagamos

POKE 23606,88

y luego tipeemos algo...

No se desesperen, se revierte haciendo

POKE 23606,0

SOFTINOTAS

1. A modo de prueba y en homenaje a **Diego**, hemos utilizado el nuevo juego de caracteres para listar el programa y el **LM** en esta nota. Espero comentarios...

2. He corregido la codificación originaria de **Diego** del número seis, porque se confundía visualmente con la letra "b" minúscula.

3. Hubiera sido interesante agregarle los acentos a tu rutina **Diego**, pero no invalida para nada tu bien merecido premio. Te felicito nuevamente.

4. Pueden alivianar y asegurar la carga de este importante programa utilizando el **Monitor LM-1** (ver **PP** número 8, página 40) para cargar el **LM** que nos enviara **Diego** recodificado en hexadecimal según el **Listado 2**. En este caso no deberán grabar el programa automáticamente a través del **Monitor** sino tal como lo dijéramos más arriba (el **Monitor** asume que es un **"programa en LM"** y en este caso no lo es y debemos evitar que transfiera el control con **RANDOMIZE USR** –línea **7060** del **Monitor**– a una zona de la memoria no preparada para ello).

LISTADO 2

```
  1 DATA "CARACTERES",64000,647
67,768,64000
  2 DATA "0000000000000000000303
0303000100000666666000000000064F
E6464FE6400",34742
  3 DATA "107E507E127E7E1001625
40810284680003E227F227F2200001B1
8300000000",20102
  4 DATA "001B2040404020180018D0
402020204180014083E0814000000181
87E7E1818D0",14608
  5 DATA "0000000000181B83000000
07E7E0000000000000000000181800030
60C1B3C60000",10667
  6 DATA "007E464A52627E00003C4
C0C0C0C0C00007E027E407E7E00007E0
27E027E7E00",31170
  7 DATA "00626262720202000007E7
E407E027E0000727407E4646767E00007E0
60C1B3030000",27573
  8 DATA "007E427E427E7E00007E6
262726202000018180000181800000001
818001818300",14428
  9 DATA "000004081008040000007
E0000700000000040E01020400000 7E0
```

```
23E20002000",14713
  10 DATA "3C007E6A6662620000 7E6
26276662000007C427C427C7C00007E6
0606060 7E00",41672
  11 DATA "007B5462626478000007E4
07E407E7E00007E60607B6060000007E6
0606E627E00",43279
  12 DATA "006262726262626200001B1
81818181800000606060646 3C0000 4C5
870584C4600",22448
  13 DATA "0060606060607E00006 17
3D6D1616100006272666626200007E6
2626262 7E00",41388
  14 DATA "007E627E6060606000007E6
26A667E0100007E627E6462620000 7E4
07E027E7E00",38043
  15 DATA "007E1818181818000006 26
2626262 7E00006262623434180000616
161607361 00",34486
  16 DATA "00426434 3836442000006 26
2627E1818000076060C183076000003C3
0303033C00",23564
  17 DATA "C0603018 0C0603000003C0
C0C0C0C3C001 83C5A9918181800000000
0000000FFFF",26765
```

```
  18 DATA "003C007E6262620000007
E027E627E004040 7E4646467E0000007
E6060607E00",36318
  19 DATA "02027E6262627E0000007
E467E407E00003C30307830300000007
E62627E023E",31851
  20 DATA "0407E4646464600000800
0181818180000040006060646 3800006
46870686400",21990
  21 DATA "0020202020383800000007
E6A6A6A6A0000007E626262620000007
E6262627E00",34963
  22 DATA "00007E4646 7E4040000007
E62627E020200003C303030300000003
C203C043C00",19912
  23 DATA "0010381818 1C1C0000006
2626262 7E0000006666662418000000 6
26A6A6A7E00",30759
  24 DATA "0000666618 1866000000 6
2627E021E0000007E0618607E0000102
0206020010",21538
  25 DATA "1818181818181818000080
4040304040 8CCCC3333000000003C424
9A1A1994 23C",35553
```

CA SETE VIRGEN
para COMPUTACION

- Fabricación propia
- Utilizamos cinta Ampex Ferrocobalto (USA)
- Las medidas se preparan en el día
- Consúltenos o haga su pedido.

Producciones ECCOSOUND S.A.

Ofrecemos calidad y precio al servicio de la informática.

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

rel. 12

MANEJO DE ARCHIVOS (IV)



ORDENAMIENTOS SUCESIVOS

Softi sigue insistiendo en que apliquemos nuestra micro en algo más que juegos... Qué plomo, no???...

En las dos notas anteriores de esta serie –en que nos ocupamos de la manera de implementar un manejo de archivos aceptable en computadoras CZ1000/1500 y TK83/85 y similares– nos declarábamos sumamente afligidos por el problema del **ordenamiento** de los datos almacenados en nuestro archivo.

Desarrollamos métodos de ordenamiento sobre un vector. Como los vectores son el soporte físico de la información, clasificada en **campos**, que estamos manejando, aplicamos estos métodos de ordenamiento a nuestro archivo. Una sofisticación que añadimos fue el uso del **vector indicial**, un vector auxiliar que ''rastrea'' los cambios de posición en el vector en ordenamiento, permitiendo conservar la relación, dentro de un **registro**, del dato que se ordenó y los restantes datos en los demás campos.

Este tratamiento **indexado** iba a ahorrarnos tiempo de ejecución en ordenamientos en archivos de muchos campos.

Quedó por considerar el problema del ordenamiento sucesivo por varios campos distintos.

Volvamos a nuestro primer ejemplo de la biblioteca. Si una primera vez ordenamos –vector indicial mediante– nuestro archivo por la columna de títulos, entonces ésto introduce en realidad un orden parcial en el conjunto de vectores, quedando relacionados los contenidos de la columna ''títulos'' con los del resto de las columnas a través del vector indicial.

En estas condiciones, ordenar por una nueva columna, digamos, ''autores'', supone la reutilización del vector indicial, al que habrá que cargar nuevamente con los contenidos iniciales: Q(1)=1,Q(2)=2,...etc.

Si no modificamos la situación de los datos ya ordenados sobre el vector ''títulos'', esta reutilización del vector indicial va a aislarlos del resto del archivo, porque es el único vínculo de éstos con el resto de los registros en que ingresaron. Luego de ordenar por ''autores'', dispondremos del resto de la información sobre los otros campos, convenientemente relacionada con la columna ''autores''; pero ya no podremos acceder en forma lógica a los contenidos de ''títulos''. Habremos quebrado la relación horizontal que los vinculaba al resto del registro.

Un pequeño ejemplo ilustrará este problema.

1. **Archivo sin ordenar, tal cual ingresó la información:**

Q	TITULOS	AUTORES	GENERO
1	Martín Fierro	José Hernández	Poesía
2	La Cautiva	Esteban Echeverría	Poesía
3	Don Segundo Sombra	Ricardo Güiraldes	Novela
4	El Kirundi	Facundu Burundarena	Ensayo

2. **Archivo ordenado por la columna "títulos"**

Q	TITULOS	AUTORES	GENERO
3	Don Segundo Sombra	José Hernández	Poesía
4	El Kirundi	Esteban Echeverría	Poesía
2	La Cautiva	Ricardo Güiraldes	Novela
1	Martín Fierro	Facundu Burundarena	Ensayo

Nótese lo necesario de referirse al resto del registro, dado un ítem presente en el vector "títulos", mediante el contenido del vector indicial. Este archivo, presentado así como se ve, relaciona, entre otras cosas, a José Hernández con el Don Segundo, por no hablar de lo inexacto que es atribuirle el Martín Fierro a Facundu Burundarena. El jamás hubiera escrito una obra en verso, porque como en Kirundi todas las palabras tienen idéntica terminación y riman todas entre sí, el concepto de poesía es totalmente desconocido en Burundi.

3. **Archivo a punto de ser ordenado por la columna "autores"**

Q	TITULOS	AUTORES	GENERO
1	Don Segundo Sombra	José Hernández	Poesía
2	El Kirundi	Esteban Echeverría	Poesía
3	La Cautiva	Ricardo Güiraldes	Novela
4	Martín Fierro	Facundu Burundarena	Ensayo

...donde ya se ha perdido toda vinculación lógica entre los contenidos de "títulos" y sus respectivos autores y géneros.

4. **Archivo ordenado por "autores"**

	TITULOS	AUTORES	GENERO
2	Don Segundo Sombra	Esteban Echeverría	Poesía
4	El Kirundi	Facundu Burundarena	Poesía
1	La Cautiva	José Hernández	Novela
3	Martín Fierro	Ricardo Güiraldes	Ensayo

...donde el contenido del vector indicial nos remite al "género" correspondiente a la obra de un determinado autor, pero donde el título de la misma es irrecuperable.

Dos soluciones serían:

A. **Antes de ordenar por segunda vez, devolver los contenidos de "títulos" a sus lugares originales**, aprovechando que éstos se mencionan en las posiciones paralelas de Q.

B. **Efectuar el ordenamiento de todos los vectores, no en los vectores mismos, sino en una "copia" exterior al archivo, vale decir, en un segundo vector auxiliar**. Con lo cual **los contenidos que ingresemos en el archivo quedarán fijos**, sin moverse nunca. En forma previa al ordenamiento, habrá que copiar los contenidos de la columna a ordenar en ese vector auxiliar "copia" y cargar los consabidos **1,2,3,...** en el **Q**.

Nos parece más elegante la segunda opción, quizá de puro acostumbrados a usar vectores auxiliares por todos lados.

Llamemos a este vector "copia" C$, considerando que debe ser alfanumérico para albergar datos de cualquiera de las tres columnas del archivo.

Fijémonos ahora en lo siguiente: nuestro programa archivero debe ser capaz, según dijimos, de ordenar un campo cualquiera; más aún, deseamos poder ordenar varios campos, uno tras otro. Lo segundo nos condujo a la introducción del **C$**. Pero *¿cómo agregar al programa la posibilidad de opción?* Es decir, ¿cómo indicar que vamos a ordenar por tal o cual columna, y efectuar el ordenamiento sin escribir un programa aparte para cada columna?

En los algoritmos, que presentamos en su momento, referentes al ordenamiento de vectores, constantemente se hace mención a un solo vector. Cada algoritmo puede servir para cualquiera de ellos, pero ¿vamos acaso a escribir el mismo programa tres veces, previendo el ordenamiento por títulos, por autores o por género, y reemplazando **T$** por **A$** o **G$** según corresponda? Un programador competente se cortaría el bus de control antes de hacer eso.

Más bien reconozcamos que la estructura de datos usada hasta ahora es insuficiente. Con vectores sueltos como soporte de nuestro archivo no iremos muy lejos, al no haber una manera cómoda, en BASIC, de referirnos a uno u otro dentro de un programa que, supuestamente, debe tratar a todos ellos por igual.

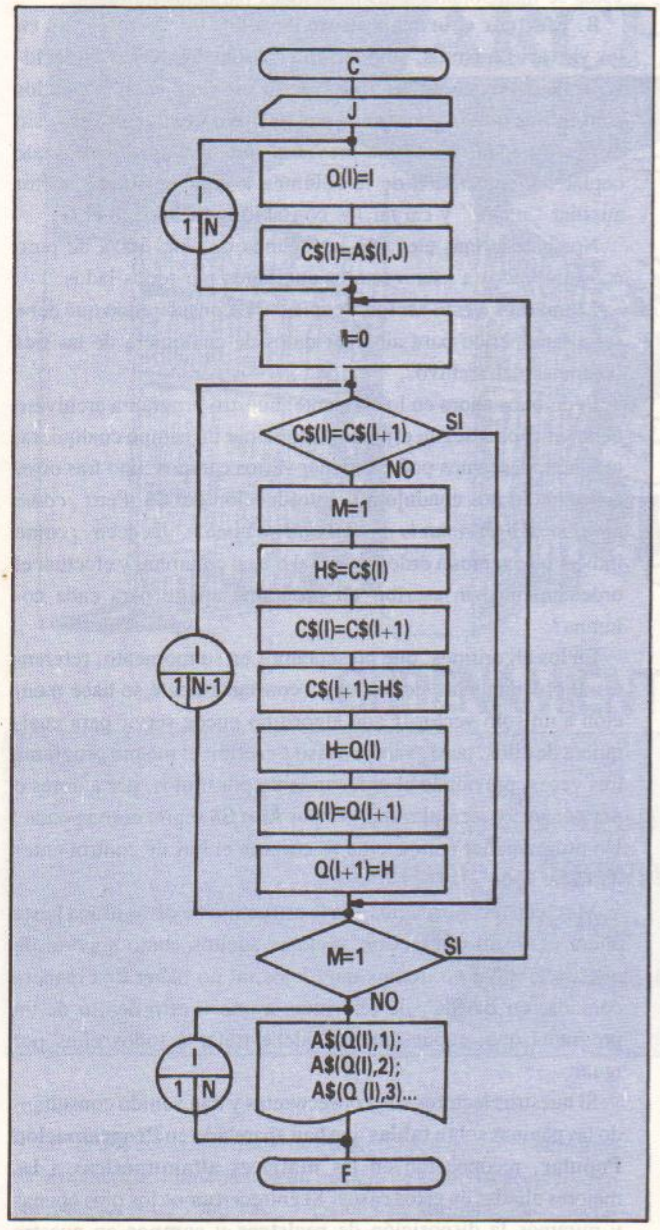
Si nuestros lectores son consecuentes y han venido consultando las páginas sobre **tablas** que han aparecido en **Programación Popular**, reconocerán en las **matrices alfanuméricas** a las mejores aliadas en estos casos. Si entrecerramos los ojos apenas y miramos la disposición de **registros** y **campos** en nuestro archivo, veremos cómo concuerdan con la descripción de **filas** y **columnas**, respectivamente, de una matriz.

La diferencia es que cada **vector** de los que veníamos usando será ahora una **columna** de la matriz que guardará el archivo completo, y será fácilmente identificable por su número de orden. Así, en el ejemplo visto más arriba, "títulos" es la columna 1 de la matriz, digamos **A$**, y todos sus elementos son reconocibles como

A$ (algo,1)

...expresión en la que **"algo"** es un número que hace referencia a la fila en que se halla el elemento en cuestión. Nada más fácil que cambiar ese **"1"** de la segunda coordenada por cualquier otro número de columna, para lo cual en la rutina de ordenamiento dejaremos abierta la posibilidad de que sea cualquiera mediante el sencillo expediente de nombrarlo con una variable.

Algoritmo que ordena la matriz por una cualquiera de sus columnas, en forma indexada:



(Una vez más, se suponen cargados los datos fila por fila –registro a registro–.)

Notemos que se ha elegido una **matriz alfanumérica** para alojar los datos. ¿Qué ocurre si nuestro archivo involucra información de tipo **numérico** ?

(Por ejemplo, si una de las columnas contiene las cantidades de páginas respectivas de los libros en existencia.)

Al figurar estas cifras como elementos de un arreglo alfanumérico, serán interpretadas como *strings*. A la hora del ordenamiento, las comparaciones serán hechas como entre tales. Por lo cual un dato ''1000'', alfanumérico, será declarado menor que otro dato, también alfanumérico, ''200'', ya que el *orden*, aplicado a *strings*, que entiende la computadora, es alfabético y no numérico; y alfabetícamente *anterior* corresponde a *menor* en este caso.

Un posible tratamiento de esta cuestión sería alterar levemente la etapa de *carga* de datos, preparando aquellas columnas de la matriz que vayan a contener especímenes de interés numérico para que no acepten datos sin antes agregar a su izquierda cierta cantidad de caracteres iguales y de orden alfabético menor que los dígitos **0,1,2...**, por ejemplo, un espacio en blanco. El efecto de esto debería ser dejar a todos los datos de igual longitud, y con un número variable de blancos adelante. Qué cantidad de blancos colocar lo responderá la observación de que, al dimensionar una matriz, *necesitamos* especificar la longitud que van a tener sus elementos. Esta está dada por lo que sea preciso con los demás datos. Supongamos que una cantidad 10 de caracteres sea suficiente para los títulos, autores y géneros. Si vamos a ingresar cantidades de páginas, respectivamente, **150,980,35** y **3550**, nos interesa entonces convertir estos datos a

xxxxxxx150 (donde la ''x'' representa el espacio),
xxxxxxx980,
xxxxxxxx35 y
xxxxxx3550, donde ahora todos tienen el mismo número de caracteres y, como puede verificarse, su orden alfabético coincide con el numérico.

De modo que, en teoría, ya está todo dispuesto para que carguemos la información y, mediante métodos de avanzada, ordenemos y listemos el archivo ordenado a nuestro placer.

Será de capital importancia que prestemos igual atención a las etapas de **modificaciones** y **búsqueda**. No vaya a ser que una sorpresiva **carta de Burundi** eche todo nuestro trabajo por tierra...

FE DE ERRATAS

Aún no llegó la esperada *carta de Burundi* que señale los errores cometidos en **Manejo de Archivos III, Programación Popular** N° 5, pág. 34, razón por la cual la escribiremos nosotros.

Allí se publicaron un diagrama de flujo y una correspondiente rutina de ordenamiento con los siguientes ''bugs'':

DONDE DICE...	...DEBE DECIR:
A(I); B(I); C(I)	A(I); B(Q(I)); C(Q(I))
PRINT A (I); B(I); C(I)	RINT A(I); B(Q(I)); C(Q(I))

...lo lamentamos en modo *profundu*.



BOLA LOCA

CONCURSO DE PROGRAMAS

GANO $ 2S

Páguese a Christian Tottino

La Plata, B.A.

Ediciones del Futuro S.A.

Softi

Acá tenemos un lindo programita, ganador también del concurso, que nos enviara **Christian**, que logró hacer con muy pocas instrucciones en Basic un original y entretenido juego. Se trata de impedir que legue al suelo una "bola" que caerá desde arriba, contando con una "red" que podremos mover a izquierda y derecha, usando las teclas **[Q] y [P].**



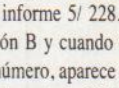
Cuando logremos acumular 210 puntos pasaremos a otra "fase" del juego, el que se hará progresivamente más difícil, puesto que la maldita "bola" caerá desde alturas cada vez menores. Al llegar a la última fase, habrá que tratar de seguir acumulando todos los puntos que podamos, hasta perder las cinco vidas de que disponemos (podemos alterar la cantidad de vidas cambiando el valor de la variable j en la línea **1600**).

Softinota: he hecho algunos "retoques" al programa original de **Christian** (cantidad de vidas en pantalla y no borrar la pantalla al final) para mejorar libremente la presentación. Te felicito nuevamente, amigo platense.

Veamos ahora el programa...

```
    REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                      @
@            BOLA LOCA                 @
@  Christian Tottino-La Plata          @
@ Rev.Programacion Popular-Nº 9        @
@ CZ-Spectrum-TK90X-TS2068    2K       @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

  10 FOR n=USR "a" TO USR "a"+39
  READ x: POKE n,x: NEXT n
  11 LET l=2: LET j=0: LET n=0
  12 PAPER 5: CLS : LET i=0: LET
 t=1: LET f=0: LET g=3
  13 PRINT AT 0,0;"puntos: "; FO
R w=j TO 4: PRINT AT 0,23+w*2; I
NK 4;"■": NEXT w
  20 FOR a=4 TO 27: PRINT AT t,a
 ;INK 6;"▼" AT 16,a; PAPER 1;"#"
  NEXT a: LET c=15: LET h=l+1
  30 LET k=0: FOR x=1 TO 8: LET
 a=c: LET c=c-(INKEY$="q" AND c>4
 )+(INKEY$="p" AND c<27)
  40 IF a=c THEN NEXT x
  50 FOR m=1 TO 1: NEXT m: PRINT
 AT 14,a;" ";AT 14,c; INK 4;"■"
  LET k=k+1: NEXT x
  75 IF f=1 THEN GO TO 130
  80 LET e=INT (RND*23)+1: LET g
=g+e: PRINT AT h,g;"●": LET f=1:
 GO TO 30
  130 IF ATTR (h+1,g)<>40 THEN GO
 TO 150
  140 PRINT AT h,g;" ";AT h+1,g;"
●": LET h=h+1: GO TO 30
  150 PRINT AT h,g;"0": BEEP 0.08
,22: BEEP 0.08,25: BEEP 0.08,18:
 PRINT AT h,g;"%": PAUSE 10: PR
INT AT h,g;"0": PAUSE 10: PRINT
AT h,g;"%": PAUSE 10: PRINT AT h
,g;" ": IF ATTR (h+1,g)=14 THEN
 GO TO 1600
  200 LET h=l+1: LET g=4: LET f=0
: LET i=i+10*k: LET n=n+i: PRINT
 AT 0,6,i: IF i>200 AND l<10 THE
N GO TO 12
  201 GO TO 30
1000 DATA 255,126,60,60,60,24,24
,24,255,255,255,0,0,0,0,0,24,126
,255,255,255,255,126,24,0
1001 DATA 0,24,102,66,66,102,24,
0,145,74,36,70,138,36,82,137
1600 LET j=j+1: PRINT AT 0,21+j*
2;" ": IF j=5 THEN PRINT AT 19,0
;"juego terminado";AT 20,0;"punt
aje total: ";n: PAUSE 100: PRINT
 FLASH 1;" presione una tecla p
ara seguir ": PAUSE 0: GO TO 11
1700 BEEP 0.5,-4: LET g=3: LET h
=l+1: LET f=0: GO TO 30
```



(viene de pág. 21)

UTILIDAD DEL PROGRAMA:

Este programa sirve para evaluar a 45 alumnos, que es el número máximo permitido por grado, sobre el tema: NUMEROS ROMANOS.

Solamente es necesario disponer de una computadora de 8K de memoria. El programa contiene 2 opciones:

A) Dado un número arábigo, los alumnos deben expresarlo con notación romana.

B) Dado un número romano, los alumnos deben expresarlo con notación arábiga.

FUNCIONAMIENTO

1º) Oprimir GOTO 1 o RUN 1.

2º) La maestra indica la opción y el número de ejercicios para cad alumno, despreocupándose luego de la preparación de los temas, de la corrección y de la calificación. (La computadora prepara los temas, con muy poca probabilidad de repetirlos, ya que dispone de 3.999 números).

3º) El alumno debe escribir su nombre y contestar los ejercicios.

4º) El programa está preparado para 45 alumnos, pero cuando el número de alumnos es menor, tal como lo indican las instrucciones, al aparecer en pantalla "NOMBRE DEL ALUMNO:", debe pulsar ENTER.

5º) Aparecido el listado de nombres y calificaciones, la maestra copia las notas en su libreta y luego pulsa CONT si se encuentra en pantalla el informe 5/ 228.

6º) En funcionamiento con la opción B y cuando el alumno oprime una letra en vez de un número, aparece el informe 2/ 176. En este caso corresponde oprimir GOTO 182.

AVISOFTIAVISOFTIAVI

VENDO: Timex 2068 con manuales, transformador y programas, en buen estado. Eduardo TE. 654-3573, después de las 17 hs.

VENDO: TK 85 $ 150, 15 programas $ 20, Joystick $ 10. Sr. Marcelo Loza. TE. 22-0235

VENDO: Programas TI 99 4A $ 100, cien juegos en Basic y extended $ 15 (incluye casete y simulador de vuelo), variedad de juegos, lecciones aprendizaje basic. TE. 204-3842, Angel.

VENDO: Sintetizador de sonidos Tron para computadoras CZ 1000/ 1500, TK 83, 85 (un año de garantía) $ 25 programas de juego con sonido varios títulos $ 15. TE. 244-4272 de 8 a 14 hs.

VENDO: Sinclair 2068 como nueva, completa, manual original, fuente, programación originales, simulador Spectrum. TE. 87-1767/89-5765.

VENDO: Programas para TK 83/85 y compatibles, 5 programas a elección por $ 2, Martin. TE. 86-4574.

VENDO: Para Commodore 64, programas, accesorios, Interface, grabador, pulsador, Reset, Cartridge, fast load. Ing. Eduardo Berkesy. TE. 792-7983.

VENDO: TS 1000, con módulo de aplicación de 10 K, cable conector y casete de juegos. Adrian. TE. 983-9949, después de las 18 hs.

VENDO: Programas para Sinclair 2068 y Spectrum. Martin. TE. 299-0412.

VENDO: Sinclair 1000 con expansor de 16 K y cables y manual. Sr. Ruben. TE. 766-2820.

VENDO: Casete TK 2000, 5 juegos $ 6, libros en portugués $ 15. Carlos. TE. 654-8316.

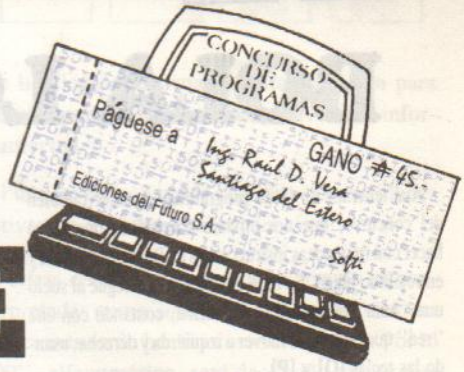
VENDO: TK 85, 16 K con cables, programas, juegos y manual de uso en castellano, excelente estado. TE. 88-2294.

VENDO O PERMUTO: Radio Shack TRS 80 NS 10 por Commodore 64. Eros. TE. 747-2360.

VENDO: Computadora CZ 1000, casi nueva. Horacio. TE. 664-5313.

VENDO O PERMUTO: TI 99 4A con 2 módulos de juegos, muy barata y en perfecto estado o permuto por Commodore 64. Diferencia a convenir. Manuel Rubio. Pte. Roca 955, Reconquista. Santa Fe 20288.

En este número finalizamos con el análisis de este excelente programa de Raúl (ver P.P. Nº 8 y 9)



COSTOS DE FABRICACION

DIAGRAMA DE FLUJOS

He creído muy conveniente incluir antes que la **estructura del programa** una información más completa y más clara que la reemplace con ventaja. Esto es precisamente el **diagrama de flujos** del programa. Para una mejor comprensión del mismo digamos:

a. Se han eliminado los detalles superfluos dando prioridad a la **lógica de ejecución** del mismo.

b. Los **conectores de página** (círculos con un número) contienen la información adicional del número de línea donde se localiza la rutina en el programa.

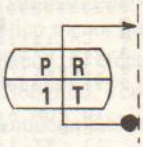
d. El bloque que representa **impresión en pantalla** es PRINT mientras que el que indica **impresión en papel** es COPY todo esto para erradicar ambigüedades desde el principio.

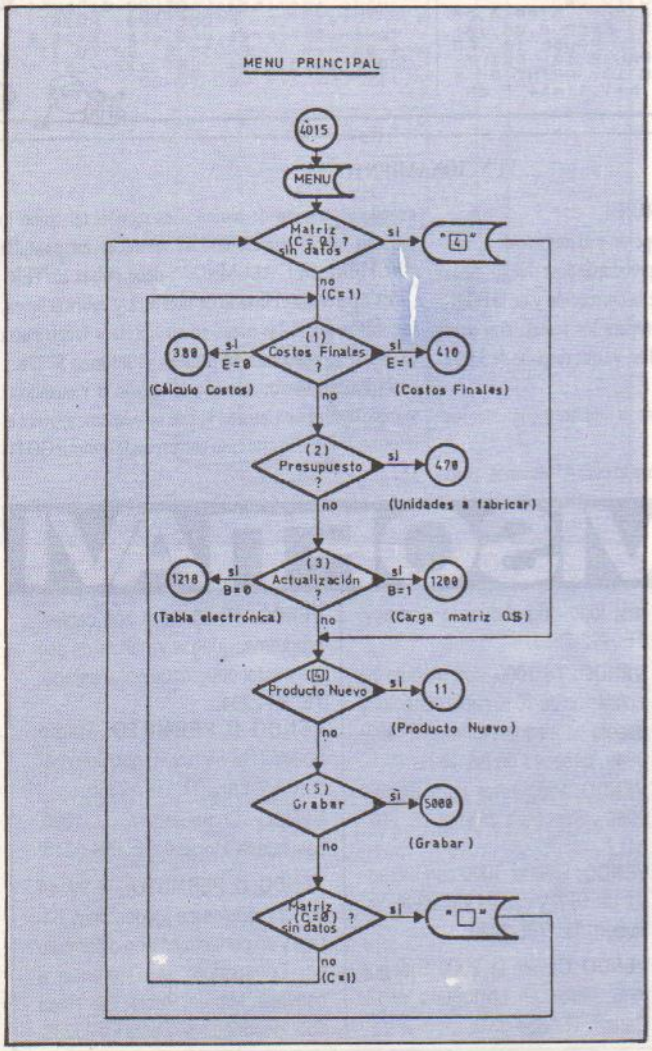
e. Personalmente me inclino para los bucles **FOR/STEP/NEXT** al uso de las siguientes representaciones

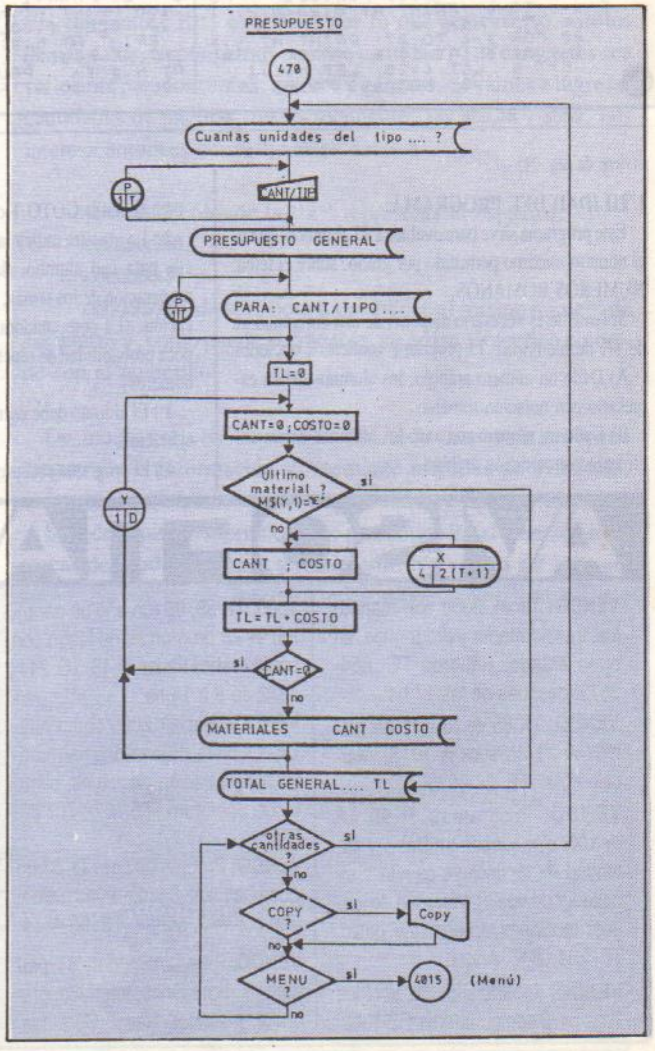
FOR I=1 TO 20
NEXT I



FOR P=1 TO T STEP R
NEXT P







f. La leyenda que aparece en las proximidades de cada conector de página corresponde al nombre de la subrrutina a donde es dirigido el control mediante ese conector.

g. Muchos bloques de decisión del tipo ◇ no son codificados en el programa mediante sentencia **IF/THEN** sino mediante algoritmos de bifurcación múltiples contenidos como expresión de los argumentos de las sentencias **GOTO**.

RUTINAS DESTACABLES:

Están constituidas por algoritmos que cumplen determinada función y que a "simple vista" no pueden detectarse fácilmente y muchos menos comprender sin dificultad su funcionamiento. A continuación intentamos aclararlas.

• ALGORITMO DE RECORTE DE CIFRAS

Habíamos citado ya la dificultad principal de nuestro humilde ordenador de poseer un máximo de 32 columnas. Si queremos en este pequeño espacio representar tabas con números, es necesario "recortar" las cifras de 9 dígitos que maneja hasta una cantidad apropiada que sin perder significación permita ser encolumnada en tablas. Creo personalmente que un total de cuatro caracteres (incluida la coma o punto decimal) son suficientes para cumplir los objetivos que persigue este programa, aunque mediante un adecuado cambio de la constante **CR** que es precisamente la cantidad de **"cifras recortadas"** y una modificación en el formato de las tablas será posible obtener en pantalla los resultados con cinco o más caracteres.

El algoritmo que lleva a cabo este recorte de cifras asume la siguiente forma:

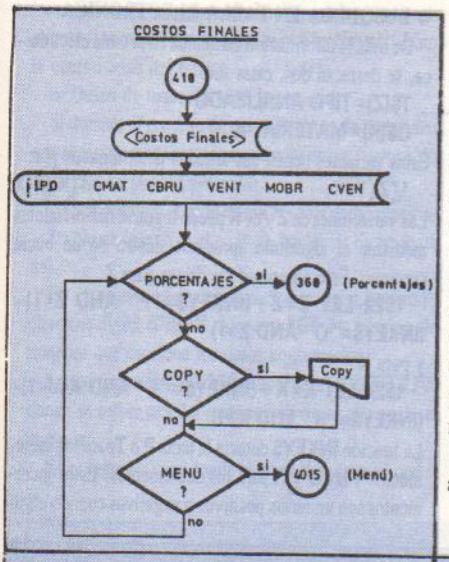
(STR$W+" ")(TO CR)

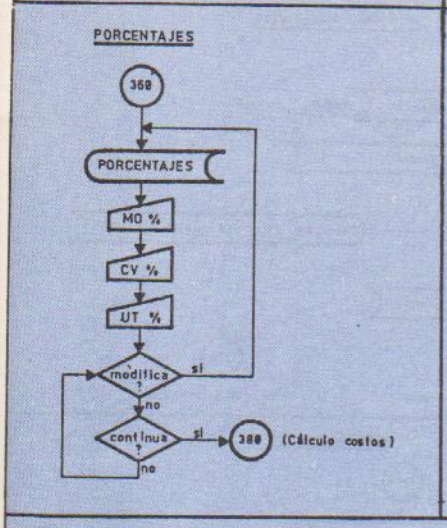
con una cantidad de espacios igual a **(CR–1)** en este caso **4–1=3** y son los que están entre comillas.

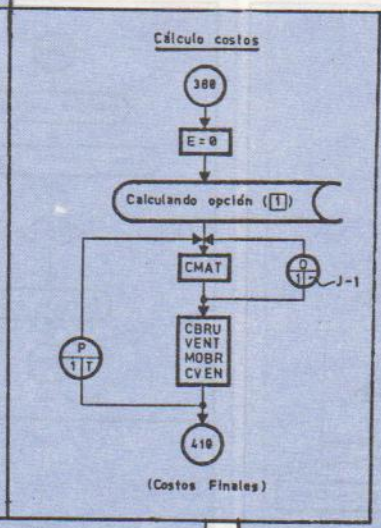
El paréntesis izquierdo constituye una cadena formada por el número a recortar **W** concatenada con una cantidad de espacios mínima para evitar luego una operación de fragmentación ilegal indicada por el paréntesis de la derecha. El paréntesis derecho conforma una fragmentación izquierda hasta el caracter **CR**, la cual será ilegal si la cadena (paréntesis izquierdo) no contiene la cantidad mínima de cuatro caracteres, y es precisamente ésta la función de los **CR–1** espacios encadenados con la string del número a recortar. En el peor caso, la cantidad mínima de caracteres que contendría **STR$ W** sería de uno correspondiente por ejemplo a **0=W**. Este algoritmo tiene el inconveniente que recorta sin permiso previo los números que superen **9999**, para este caso, al comienzo de esta nota citamos algunas soluciones. No obstante este inconveniente creo que es muy útil para la gran mayoría de los casos prácticos.

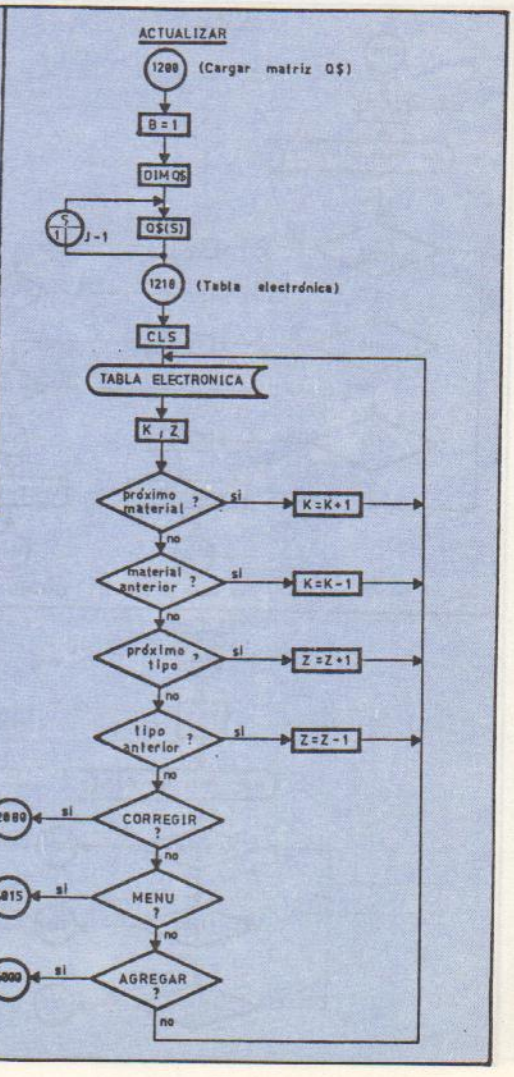
• PROTECCION DIVISION POR CERO

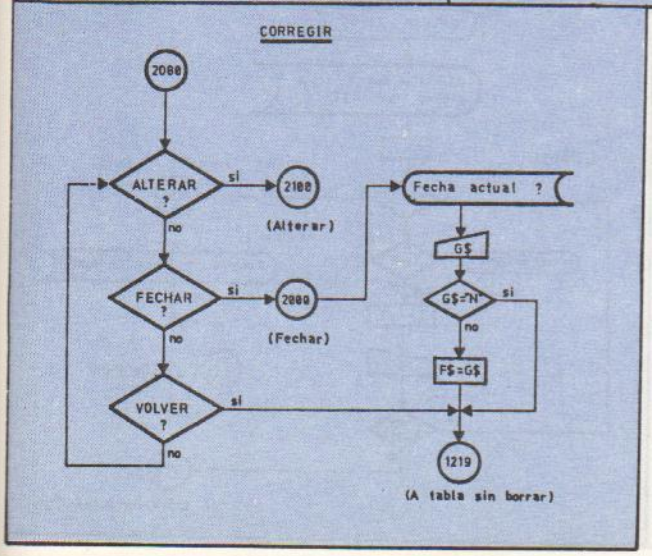
En la línea **2500** comienza una rutina donde se efectúa una división. Aquí precisamente es donde debe cuidarse ingresar a ella con un determinador nulo. Esta rutina es llamada desde las rutinas **NUEVO COSTO** y **NUEVA CANTIDAD DE UNIDADES/**. En la segunda aparece un algoritmo como el siguiente:











2330 GOTO VAL "2500–185 * (C(K,DO)≤0)"

reemplazando constantes:

GOTO VAL "2500–185 * (C(K,2)≤0)"

eliminando la función **VAL** la que cumple únicamente el cometido de requerir menor cantidad de bytes de memoria (con fines de ahorro puramente) nos queda:

GOTO 2500–185 * (C(K,2)≤0)

aquí observamos que la expresión **(C(K,2)=0)** constituye una proposición y de su veracidad o no tendremos para ella el valor **1** ó **0** respectivamente. Para el caso en que sea verdad es:

GOTO 2500–185

efectuando la resta:

GOTO 2315 que corresponde al número de línea donde la máquina pedirá nuevamente la entrada **(INPUT)** de **C(K,DO)** puesto que se ha detectado que el valor anterior es nulo o negativo mediante la proposición ya comentada.

Un razonamiento idéntico puede hacerse para la primera rutina con la diferencia que la única condición que se impone aquí es que no se aceptarán cantidades negativas debido a que son precios de costo pero sí pueden ser nulos porque por otra parte este valor es el numerador de la división que se realiza en la rutina **2500**. El algoritmo que estamos comentando es:

2200 GOTO VAL "2500–320 * (C(K,U)<0)"

si la proposición que está entre paréntesis es verdadera, el control de la información vuelve a la línea **2180** donde se solicitará un nuevo valor de **C(K,U)** que no debe ser negativo.

Para el caso en que ambas proposiciones que están en las líneas **2330** y **2200** sean falsas, obtenemos simplemente **GOTO 2500** que es donde se encuentra la división:

2505 LET C(K,3)=C(K,U)/C(K,DO)

● NO IMPRIMIR CANTIDADES NULAS

Esta facilidad es bastante deseable a la hora de los listados de materiales necesarios para el lote. Es conveniente no escribir en la lista todos aquellos materiales que participan en el lote de fabricación con cantidades nulas. Este detalle de programación se localiza en la subrutina **PRESUPUESTOS** dentro del bucle **Y** y precisamente en la línea:

565 IF C(Y, DO*T+CI)=0 THEN GOTO VAL "575"

la que podía haber sido escrita también:

565 IF C(Y, DO*T+CI)=0 THEN NEXT Y

reemplazamos las constantes en primer término para clarificar la expresión:

565 IF C(Y,2*T+5)=0 TIEN NEXT Y

La condición de nulidad es referida al elemento **C(Y,2T+5)** de la matriz de costos que es la que contiene la información de la cantidad de unidades del material **M$(Y)** que intervienen en el tipo **T**. La sentencia **NEXT Y** saltea las líneas **570** y **573** que son las que imprimen en pantalla para continuar con el siguiente material para el caso de que la presente cantidad sea nula.

● BUSQUEDA EN TABLA ELECTRONICA

De toda la información contenida en la tabla electrónica, se destacan dos, estas son:

T$(Z)=TIPO ANALIZADO

Q$(K)=MATERIAL K

Estas variables tienen sus ámbitos determinados por:

1≤Z≤T **1≤K≤J–1**

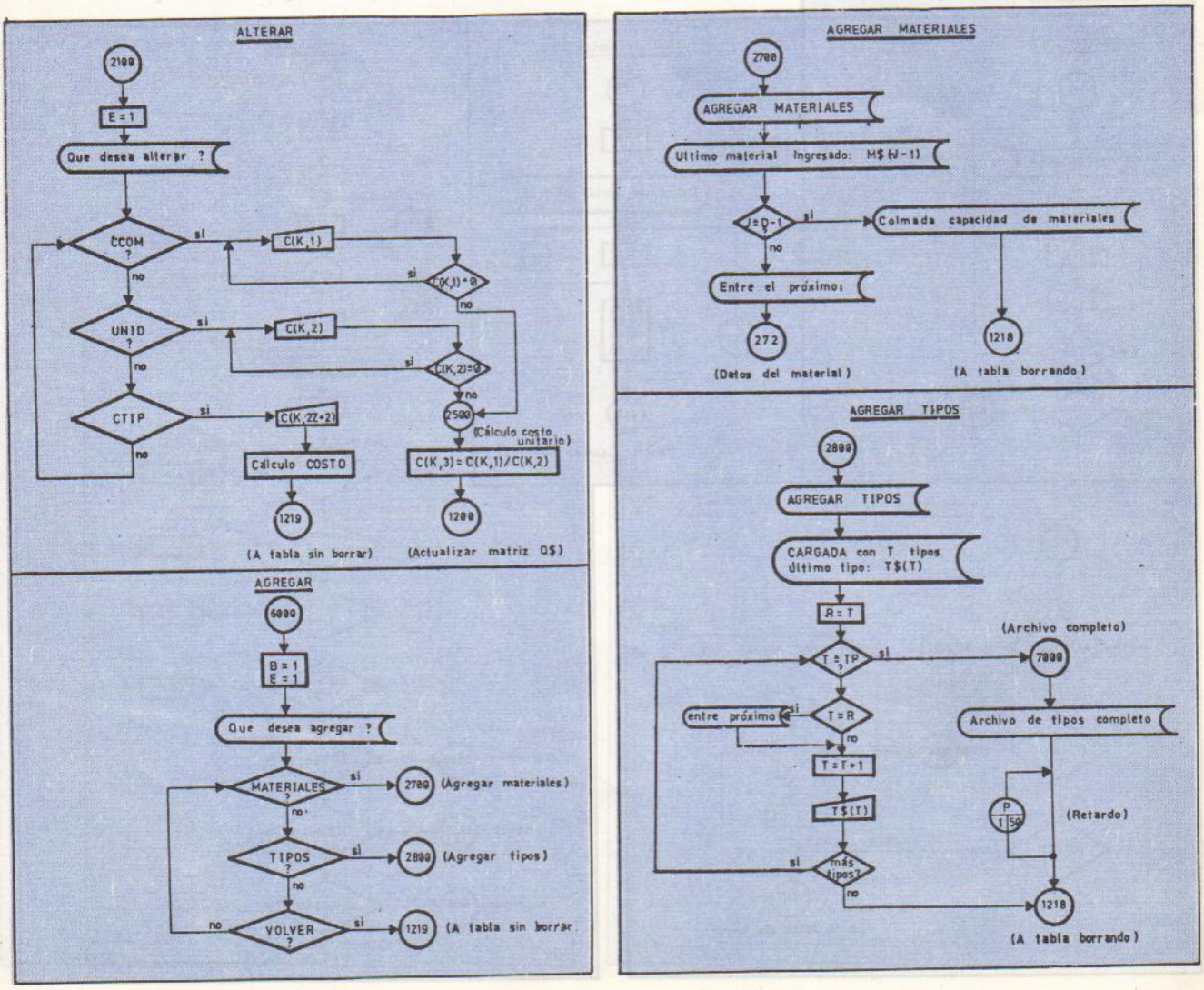
Las variaciones de **Z** y de **K** puede lograrse definiéndolas mediante el algoritmo apropiado dentro de un bucle continuo. Este algoritmo puede ser para **Z**:

1522 LET Z=Z+(INKEY$="P" AND Z<T)–(INKEY$="O" AND Z>1)

y para **K**

1524 LET K=K+(INKEY$="T" AND K<J–1)–(INKEY$="R" AND K>1)

La función **INKEY$** detecta la tecla **P** o **T** para los incrementos y las **O** y **R** para los decrementos. Estos incrementos son unitarios positivos o negativos cuando algu-



na de las teclas nombradas es pulsada (el incremento en valor absoluto es el resultado nulo o unitario de las proposiciones que se encuentran entre paréntesis) y es nulo cuando ninguna tecla es pulsada.

RECOMENDACIONES FINALES

Estas recomendaciones persiguen el fin de advertir a los usuarios del programa, de detalles, atajos, comportamiento "normal" y todo aquello que facilite al máximo la operatividad del mismo.

a. Datos de material mal ingresado

Si durante el proceso de carga de datos de materiales, nos equivocamos, para enmendar el error tenemos dos casos que se pueden presentar.

1°) Se completó el ciclo de bucles para dicho material.

2°) No se completó el ciclo de bucles de dicho material.

En el primer caso debemos hacer: (con comandos directos) digite **BREAK**, luego... **LET J=** número del material que contiene los datos equivocados.

Luego digite **GOTO 272** o si prefiere **GOTO 2700** donde se informará del material previo al que se desea ingresar.

En el segundo caso donde el bucle del material que se está cargando no se ha completado, digite **BREAK** y luego **GOTO 272** o con mayores referencias **GOTO 2700**.

Si deseamos modificar los datos (todos o algunos) del material por ejemplo el que lleva el número **15**, haremos:

BREAK
LET J=15
GOTO 272 (con rapidez)
CARGAR DATOS
ENTRAR STOP (como respuesta a la solicitud del próximo material)

LET J= número total de materiales más uno (esto es la restitución del valor original **J** antes de la modificación que se acaba de hacer).

b. Antes de comenzar a usarlo al programa

Asegúrese de disponer todos los datos necesarios para su caso particular, si no obstante estas precauciones le faltaran algunos, invéntelos con una cantidad que sea posible que exista en la práctica. Luego podrá buscar estos datos faltantes y mediante el uso de la tabla electrónica completarlos en el ordenador.

c. Grabaciones en cassettes

Se supone que este programa será para **uso o aplicación** por lo que se pone en evidencia lo innecesario de las REMs que sirvan de rótulo a las diversas subrutinas con el fin de ahorrar un poco más de 400 bytes que pueden resultar muy útiles al momento de su utilización en un problema exigente.

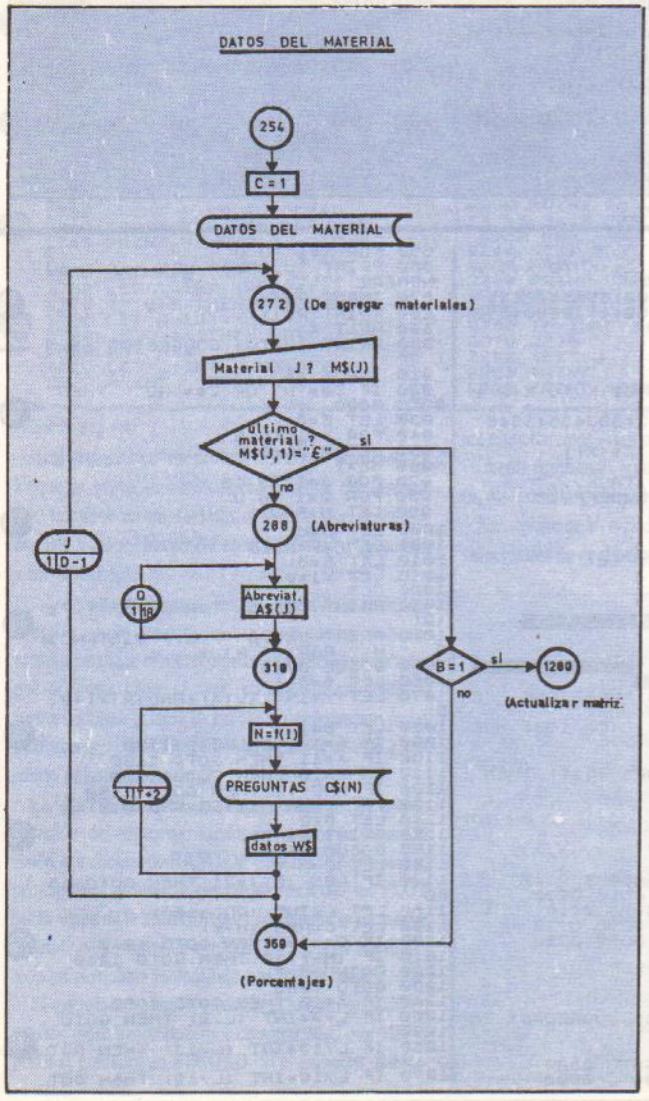
Una vez que tiene Ud. la máquina cargada con su problema de costo de fabricación resuelto, sugerimos grabarlo dos o tres veces en otro cassette aparte quedando así disponible para futuras actualizaciones. Conserve con sumo cuidado la copia de seguridad efectuada al principio.

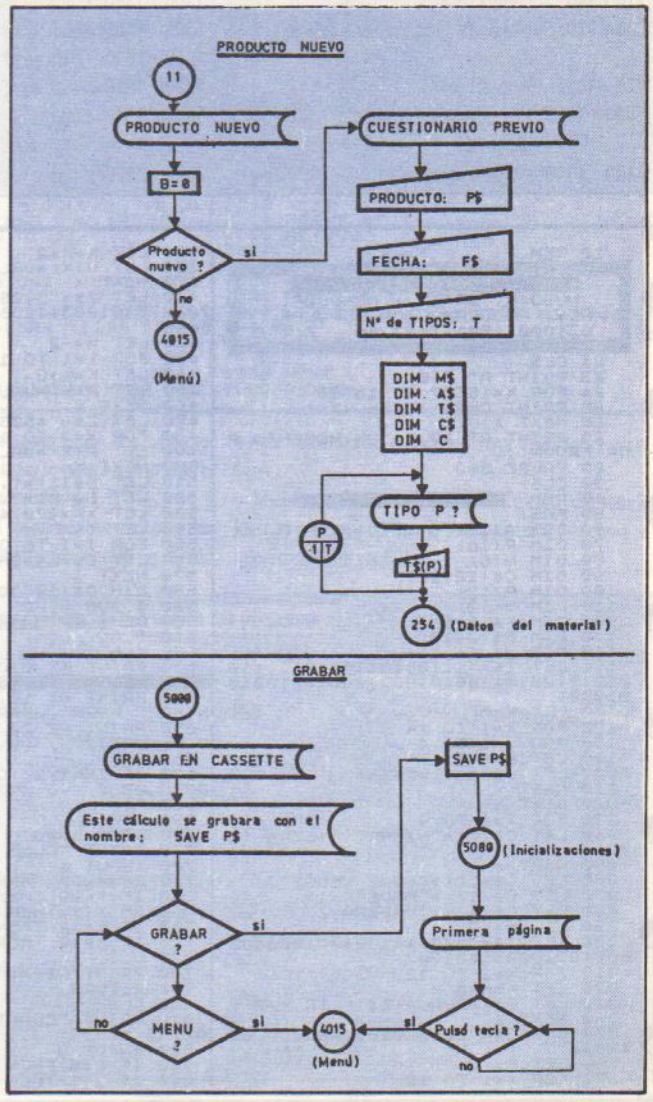
d. Cómo hacerlo arrancar

Una vez digitado el programa completo seguramente, además de las consabidas operaciones de debugging que mis lectores ya conocen bien, yo me permitiré agregarle una operación más. Consiste en entregar a un amigo o compañero la tarea de una minuciosa revisión final para detectar los BUGS más frecuentes que son los de digitación. Es importante que sea otra persona la que repase el programa puesto que en nuestro inconsciente anidan errores conceptuales "muy convincentes" de nuestro consciente.

Hecho lo anterior nos remitimos al tema ya desarrollado **TABLA DE VARIABLES FANTASMAS**. Allí tomaremos las instrucciones en comandos directos para **"hacer arrancar"** el programa.

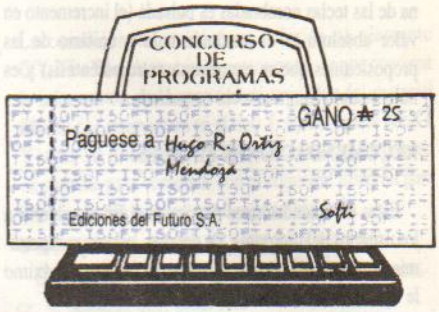
FIN





GOLF

El eximio golfista mendocino *Hugo* nos envía esta simulación de su juego predilecto, que a nosotros nos parece muy bien lograda. Con ustedes, el autor:



Comentarios del programa

El juego es más que nada una simple simulación. El mismo fue escrito en 1984, como de texto simplemente, y para enviarlo al concurso le agregué las subrutinas de dibujo, de ahí que éstas no funcionen del todo bien (sólo en un 80%). De todas maneras las mismas pueden eliminarse sin afectar el juego propiamente dicho; además dada la baja resolución del **TS1000,** no puede esperarse mucho más. La única instrucción es la elección del palo de golf, para cada golpe, la misma está ayudada por una subrutina a la que se bifurca entrando **"99",** luego se puede elegir el palo correspondiente mediante un número del 1 al 12.

Comentario aparte para la línea **2550** del programa parece complicada pero no lo es. Sencillamente decide si debe imprimir la pelotita de izquierda a derecha o de derecha a izquierda mediante operadores lógicos (no se me ocurrió nada mejor).

Tabla de variables

A.: distancia de cada hoyo
P.: par de cada hoyo
N.: score de c/participante
C$.: nombre de c/palo de golf
R.: alcance de cada palo de golf
V.: variación de alcance p/azar
P$.: nombre de cada participante
B$.: sonido del golpe (1=bien 2=errado)
J$.: variable de azar p/golpe del putter
E.: par del juego
C.: distancia al hoyo después de c/golpe
S.: score de cada hoyo

Auxiliares
K - Z$ - X - Y - XP - S$ - U - H - Y1 - L - A - L8 - A1 - I - Y2 - F - Q - J

Estructura del programa

10 - 40: Presentación Rev. Programación Popular
50 - 610: Inicialización de variables
620 - 700: Presentación de programa
710 - 790: Subrutina de toma de datos
800 - 920: Subrutina de toma de datos de los participantes
930 - 1015: Inicio del juego
1020 - 1120: Primer mensaje y cálculo de distancia alcanzada
1130 - 1140: Golpe errado
1150 - 1230: Subrutina común deriva s/distancia alcanzada
1240 - 1280: Subrutina de azar
1300 - 1380: Caída en trampa de arena
1390 - 1480: Salida o no de la trampa de arena
1490 - 1610: Caída en el Bosque
1620 - 1650: Caída en el espejo de agua
1660 - 2040: Llegada al "Green" finalización del hoyo
2050 - 2290: Score y posiciones del juego
2300 - 2340: Fin del juego
2350 - 2430: Impresión de palos de golf y sus alcances
2440 - 2450: Estas líneas son un "BUG" del programa
2500 - 2560: Dibujo de la cancha y trayectoria de la pelotita
2700 - 2860: Dibujo del "Green" y trayectoria de la pelotita

```
  10 REM
 JUEGO DE GOLF
   HUGO R. ORTIZ - MENDOZA
 REV. PROGRAMACION POPULAR
 CZ1000/1500-TK83/85   8.0KB
  21 CLS
  23 PRINT AT 7,0;
  24 FOR X=16537 TO 16728
  25 PRINT CHR$ (PEEK (X));
  26 NEXT X
  28 PRINT AT 20,5;"UN MOMENTO P
OR FAVOR....."
  29 PAUSE 300
  30 CLS
  40 REM  INICIALIZACION
  50 FAST
  60 DIM A(18)
  70 DIM P(18)
  80 DIM N(6)
  90 DIM C$(12,10)
 100 DIM R(12)
 110 DIM V(12)
 120 DIM P$(6,8)
 130 DIM B$(2,10)
 140 DIM J$(14,4)
 150 LET Z$="11101101101101111110
01010100101100101001100010010010
01000"
 160 LET K=0
 170 FOR X=1 TO 14
 180 FOR Y=1 TO 4
 190 LET K=K+1
 200 LET J$(X,Y)=Z$(K)
 210 NEXT Y
 220 NEXT X
 230 FOR X=1 TO 9
 240 LET C$(X)="HIERRO "+STR$ (X
)
 250 NEXT X
 260 LET C$(10)="SAND WEDGE"
 270 LET C$(11)="MADERA 1"
 280 LET C$(12)="MADERA 2"
 290 LET K=-2
 300 LET Z$="17014013012010008006
00400200011801170"
 310 FOR X=1 TO 12
 320 LET K=K+3
 330 LET R(X)=VAL (Z$(K TO K+2))
 340 NEXT X
 350 LET Z$="30302020302030302024
9030"
 360 LET K=-1
 370 FOR X=1 TO 12
 380 LET K=K+2
 390 LET V(X)=VAL (Z$(K TO K+1))
 400 NEXT X
 410 LET Z$="34050021852540118437
0222751619034423655042719660435
544"
 420 LET K=-2
 430 FOR X=1 TO 18
 440 LET K=K+3
 450 LET A(X)=VAL (Z$(K TO K+2))
 460 NEXT X
 470 LET Z$="45354343534354354 5"
 480 FOR X=1 TO 18
 490 LET P(X)=VAL (Z$(X))
 500 NEXT X
 510 LET B$(1)="...TACK..."
 520 LET B$(2)="..PUUFFF.."
 530 LET XP=270/48
 540 LET S$=""
 550 FOR I=1 TO 32
 560 LET S$=S$+CHR$ (0)
 570 NEXT I
 580 DIM D$(12,30)
 590 GOSUB 9600
 600 REM  INICIO DEL JUEGO
 610 SLOW
 620 CLS
 630 PRINT AT 0,0;" J U E G O
  D E   G O L F "
 640 PRINT AT 10,3;"PRESIONE ""E
NTER"" PARA JUGAR"
 650 PRINT AT 20,4;"(C) 1984 HUG
O R. ORTIZ"
 660 RAND
 670 IF INKEY$<>CHR$ (118) THEN
GOTO 0660
 680 REM
 690 REM
 700 GOTO 0800
 710 PRINT "CLUB ??"
 720 INPUT X
 730 IF X=99 THEN GOSUB 2340
 740 IF X=99 THEN GOTO 0710
 750 IF X<1 THEN GOTO 0710
 760 IF X>12 THEN GOTO 0710
 770 IF X=10 THEN GOTO 0710
 775 CLS
 780 PRINT C$(X)
 790 RETURN
 800 CLS
 810 PRINT "CUANTOS JUGADORES (M
AX. = 6) ??";;;
 820 INPUT U
 830 IF U>6 THEN GOTO 0800
 840 IF U<1 THEN GOTO 0800
 850 FOR X=1 TO U
 860 PRINT "JUGADOR NRO ";X;" SU
 NOMBRE ??"
 870 INPUT P$(X)
 880 PRINT P$(X);;;;
 890 NEXT X
 900 PRINT "DATOS CORRECTOS (S/N
) ??"
 910 INPUT Z$
 920 IF Z$="N" OR Z$="NO"  THEN
GOTO 0800
 930 LET E=0
 940 FOR X=1 TO 6
 950 LET N(X)=0
 960 NEXT X
 970 FOR J=1 TO 18
 980 FOR Q=1 TO U
 990 LET H=A(J)
1000 LET C=H
1005 LET XP=H/48
1010 LET S=0
1015 LET Y1=0
1020 CLS
1030 PRINT "ES EL TURNO DE: ";P$
(Q)
1040 PRINT "HOYO: ";J;" DISTANCI
A: ";H;" PAR: ";P(J)
1050 GOSUB 0710
1060 LET S=S+1
1070 LET Y=INT (V(X)*RND+R(X)+0.
5)
1080 LET B=1
1090 IF S=1 THEN GOTO 1150
1100 IF X=11 THEN GOTO 1130
1110 IF X=10 THEN GOTO 1150
1120 IF RND>.2 THEN GOTO 1150
1130 LET Y=INT (1+30*RND+0.5)
1140 LET B=2
1150 PRINT B$(B);
1155 GOSUB 2500
1160 LET C=H-Y
1170 IF ABS (C)<=21 THEN GOTO 16
80
1180 LET L=INT (RND*100)
1190 LET C=ABS (H-Y)
1200 IF C<=30 THEN GOTO 1240
1210 IF (H-Y)>0 THEN GOTO 1240
1220 PRINT " ";
1230 GOTO 1290
1240 IF X=10 THEN GOTO 1650
1250 IF L/9=INT (L/9) THEN GOTO
1290
1260 IF L/13=INT (L/13) THEN GOT
O 1490
1270 IF L/19=INT (L/19) THEN GOT
```

```
0 1620
1280 GOTO 1650
1290 LET B=1
1300 LET A=4
1310 LET L8=INT (RND*6)+1
1320 PRINT "SU GOLPE: ";S;" A LA
 ARENA"
1330 LET S=S+1
1340 IF B<2 THEN GOTO 1390
1350 PRINT "DONDE APRENDIO A JUG
AR AL GOLF ? (PRESIONE ""ENTER""
)"
1360 REM
1370 IF INKEY$<>CHR$ (118) THEN
GOTO 1370
1380 GOTO 1450
1390 PRINT " FUERZA P/PUTTER (1-
4) ??"
1400 INPUT A1
1410 IF A1<1 OR A1>4 THEN GOTO 1
390
1415 CLS
1420 IF J$(L8+A,A1)="1" THEN GOT
O 1450
1430 LET B=2
1440 GOTO 1320
1450 LET X=10
1460 LET H=C
1470 PRINT "CLUB ";C$(10)
1480 GOTO 1070
1490 PRINT "SU GOLPE: ";S,,"CUID
ADO CON EL BOSQUE."
1500 PRINT " EL CADDIE LE SUGIER
E QUE USE LOS HIERROS 8 O 9"
1510 PRINT " HOYO A: ";C;" YARDA
S"
1520 LET H=C
1530 LET L=INT (RND*4)+1
1540 GOSUB 0710
1550 IF X=8 OR X=9 THEN GOTO 106
0
1560 IF L=3 THEN GOTO 1060
1570 PRINT " Y SE FUE AL BOSQUE.
..."
1580 LET S=S+1
1590 PRINT "UD. PERDIO EL TIRO Y
 EL CADDIE"
1600 PRINT "SE ESTA ACORDANDO DE
 SU FAMILIA"
1610 GOTO 1650
1620 PRINT " AL AGUA..."
1630 LET S=S+1
1640 PRINT ,," SU GOLPE: ";S
1650 PRINT "HOYO A ";C;" YARDAS"
1660 LET H=C
1670 IF C>21 THEN GOTO 1050
1680 FOR I=1 TO 30
1690 NEXT I
1700 LET C=ABS (C)
1710 IF C=0 THEN GOTO 2000
1720 PRINT
1730 PRINT " EN EL GREEN..."
1740 PRINT " SU GOLPE: ";S
1745 PRINT " HOYO A: ",C*3," PIE
S"
1750 LET A=0
1760 LET L=INT (RND*4)+1
1770 IF C<8 THEN GOTO 1820
1780 LET A=10
1790 IF C>14 THEN GOTO 1820
1800 LET A=4
1810 LET L=INT (RND*6)+1
1820 PRINT " FUERZA P/PUTTER (1-
4) ??"
1830 INPUT A1
1840 IF A1<1 OR A1>4 THEN GOTO 1
820
1845 CLS
1850 LET S=S+1
1860 IF J$(L+A,A1)="0" THEN GOTO
 1940
1870 IF C<6 THEN GOTO 2040
1880 PRINT "DRIBBLE...";
1890 IF C<12 THEN GOTO 2040
1900 PRINT "SQUIBBLE...";
1910 IF C<18 THEN GOTO 2040
1920 PRINT "TRICKLE....PLUNK.."
1930 GOTO 2040
1940 LET C=C-7
1942 LET Y2=C*2.5
1945 IF Y2<1 THEN LET Y2=46
1947 GOSUB 2700
1950 IF C>0 THEN GOTO 1740
1960 PRINT "LA PELOTITA QUEDO AL
 BORDE DEL HOYO SOPLELA (PRESIO
NE ""ENTER"")"
1970 IF INKEY$<>CHR$ (118) THEN
GOTO 1970
1980 PRINT "...PLOP..."
1990 GOTO 2045
2000 IF S<>1 THEN GOTO 2030
2010 PRINT "$$$$$ HOYOOOO EN UNO
OOOOO $$$$$"
2020 GOTO 2040
2030 PRINT "$$$$$ GRAN GOLPE LA
EMBOCO $$$$$"
2040 LET Y2=48
2042 GOSUB 2700
2045 LET N(J)=N(J)+S
2050 PRINT
2060 PRINT "**********************
***********"
2070 PRINT "  ";S;" GOLPES - HOY
O ";J;"  PAR: ";P(J);"  "
2080 PRINT "**********************
***********"
2090 PRINT
2100 PRINT "PRESIONE ""ENTER"" P
ARA SEGUIR"
2110 RAND
2120 IF INKEY$<>CHR$ (118) THEN
GOTO 2110
2130 NEXT J
2140 PRINT
2150 LET E=P(J)+E
2160 CLS
2170 PRINT "SCORE EN ";J;" HOYOS
 PAR ";E
2180 PRINT
2190 FOR W=1 TO J
2200 IF N(W)<>E THEN GOTO 2230
2210 PRINT P$(W);" GOLPES ";N(W)
;" = A LA PAR"
2220 GOTO 2240
2230 PRINT P$(W);" GOLPES ";N(W)
;(" = 5/LA PAR" AND N(W)>E);(" =
 B/LA PAR" AND N(W)<E)
2240 NEXT W
2250 PRINT
2260 PRINT "PRESIONE ""ENTER"" P
ARA SEGUIR"
2270 RAND
2280 IF INKEY$<>CHR$ (118) THEN
GOTO 2270
2290 NEXT J
2300 PRINT
2310 PRINT "$$$$$$$$$$FIN DE JUE
GO$$$$$$$$$$"
2320 PRINT
2330 GOTO 9990
2340 CLS
2350 PRINT "SUS PALOS DE GOLF"
2360 PRINT
2370 PRINT "#  MIN.ALC
.  MAX.ALC."
2380 FOR L=1 TO 12
2390 PRINT D$(L)
2400 NEXT L
2410 PRINT
2420 PRINT "HOYO ";J;" A ";C;" Y
ARDAS - GOLPE: ";S
2430 RETURN
2440 PRINT TAB 10;"END PROGRAM"
2450 REM
2500 PRINT AT 21,0;"
          "
2510 PRINT AT 17,29;"*"
2520 PRINT AT 18,28;"***"
2530 PRINT AT 19,24;"  *****"
2540 PRINT AT 20,24;"    I"
2550 FOR F=(Y1/XP)*(Y1<A(J))+((
Y1+Y)/XP)*(Y1>=A(J)) TO (((Y1+Y
)/XP)*(Y1<A(J)))+(Y1/XP)*(Y1>=A(
J)) STEP (Y1<A(J))-(Y1>A(J))
2555 IF F>61 THEN GOTO 2580
2560 PLOT F,2
2570 UNPLOT F,2
2575 PRINT AT 20,24;"    I"
2580 NEXT F
2590 LET Y1=Y1+Y*(Y1<A(J))-Y*(Y1
>=A(J))
2592 IF F>61 THEN GOTO 2600
2595 PLOT F,2
2600 FOR I=1 TO 30
2610 NEXT I
2620 FOR I=17 TO 21
2630 PRINT AT I,0;"
2640 NEXT I
2650 PRINT AT 7,0;
2660 RETURN
2700 PRINT AT 21,0;"
          "
2710 PRINT AT 17,25;"
2720 PRINT AT 18,25;"
2730 PRINT AT 19,25;"
2740 PRINT AT 20,25;"
2750 FOR F=0 TO Y2
2760 PLOT F,2
2770 UNPLOT F,2
2780 NEXT F
2790 IF Y2<>48 THEN PLOT F,2
2795 IF Y2=48 THEN PRINT AT 21,2
4;"
2800 FOR I=1 TO 30
2810 NEXT I
2820 FOR I=17 TO 21
2830 PRINT AT I,0;"
2840 NEXT I
2850 PRINT AT 7,0;
2860 RETURN
3000 SAVE "GOLF"
3100 RUN
9600 REM
9800 FOR L=1 TO 12
9900 LET D$(L)=C$(L)+S$( TO 5)+S
TR$ A(L)
9920 LET D$(L,26 TO )=STR$ (U(L)
+R(L))
9950 NEXT L
9999 RETURN
```

(viene de pág. 16)

Si el grabador no **'monitorea'**, es decir, no reproduce lo que se graba simultáneamente, **habrá que grabar previamente** en un cassette el sonido que deseamos analizar y luego **'hacérselo escuchar'** a la computadora a través del EAR.

Al completarse el espacio en la memoria destinado a la grabación, lo que ocurrirá en unos segundos, volveremos a visualizar el menú en pantalla; una primera cosa que podemos hacer es 'ver' nuestra voz pulsando en forma continua la tecla **8**. En cualquier momento podemos averiguar qué dirección de memoria se está graficando pulsando la tecla **D**.

Lo que veremos es una representación del sonido obtenida del siguiente modo: se rastrea la memoria **RAM** por encima de **RAMTOP** (que habíamos fijado al principio) y se analiza cada byte haciendo un **PLOT** hacia arriba de la línea cero por cada bit que se encuentre 'a uno'. Al terminar de explorar los 8 bits de esa dirección en particular se continúa con los siguientes.

Si queremos volver a escuchar el sonido grabado en versión digitalizada podemos conectar MIC con MIC y poner el grabador en RECORD. Si el grabador monitorea escucharemos –desconectando EAR– el sonido previamente codificado, y en otro caso **habrá que grabarlo en un cassette** y después reproducirlo.

Las opciones **V** y **B** del menú permiten **alterar las velocidades de grabaciones y reproducción** (grabando a más velocidad se consigue mayor fidelidad y viceversa).

Mientras estemos examinando el 'ploteo' del sonido logrado podemos oprimir **Z** y obtendremos una copia en impresora de la pantalla."

Estructura del programa

1	Soporte del LM
30-60	Contienen los códigos del LM en Hexadecimal.
70-220	Colocan LM en **REM** de línea **1**.
130-170	Anti-Bug, detectan errores de tipeado.
250	Llama a la subrutina que imprime el menú.
330	Si se pulsó **ENTER** imprime el menú.
340	Si no se pulsó **G** saltea rutina de grabación.
350-380	Llama a subrutina en LM para grabar lo que se encuentre en conector EAR.
390	Si no se pulsó **R** saltea la siguiente rutina.
400-450	Llama a subrutina en LM para reproducir el sonido grabado.
460	Si no se pulsó **8** saltea rutina de gráficos.
470-570	Rutina de gráficos.
580	Si no se pulsó **V** saltea la siguiente rutina.
590-640	Llama a s.r. en LM para **SCROLL** izquierdo.
650-740	**INPUT** de velocidad de grabación.
750	Si no se pulsó **B** saltea la siguiente rutina.
760-810	**SCROLL** izquierdo.
820-910	**INPUT** de velocidad de reproducción.
930-1030	S.r. de menú.



Lenguaje de máquina (IV)

Sistema Binario INC, DEC

En nuestra nota anterior hablamos un poco acerca de la notación convencional que suele usarse al escribir números cuando programamos en **LM**, tanto si representan instrucciones como si representan datos.

Al referirnos a las ventajas de esta notación, el **sistema hexadecimal**, mencionamos como al acaso que era sumamente fácil "traducir" un número dado de su expresión **binaria** a la **decimal** y viceversa.

Vamos a insistir en esto, refrescando de paso nuestros conocimientos sobre los números binarios y su manejo.

4.1. Un byte, dos bytes...

Resumimos algunos hechos importantes:

1. Un byte consta de ocho bits.

Es decir que, para hacernos idea de qué cosa es un byte, una posición de memoria perteneciente a la RAM o a la ROM, basta pensar sencillamente en un conjunto de ocho casilleros, de los cuales cada uno puede contener un *cero* ("bit apagado") o un *uno* ("bit encendido").

2. En un byte pueden escribirse números entre 0 y 255.

El número más pequeño que es posible escribir en un byte es 0 ("todos los bits apagados") y el mayor es 255 ("todos los bits encendidos"); veremos en un momento una justificación de este hecho.

(En realidad esto es válido sólo si consideramos números no negativos; encontraremos más adelante una manera natural de representar los números menores que 0.)

3. Todo número perteneciente a esa gama ("números a un byte") es representable por dos dígitos hexadecimales. Así...

0 decimal **00H,**
255 decimal .....**FFH.**

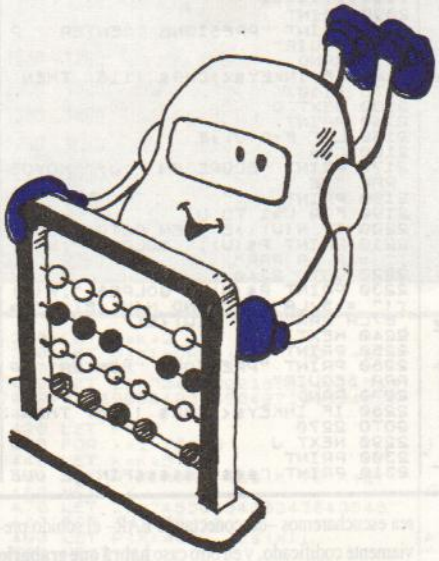
4. Los números a dos bytes están entre 0 y 65535, y quedan expresados por cuatro cifras hexadecimales. Así...

0 decimal .....**00 00 H**
65535 decimal .....**FF FF H**

4.2. El sistema binario

Ahora volvamos a nuestra idea de los ocho casilleros e intentemos explicar *qué tienen que ver los bytes con los números binarios*. Para ello examinaremos el **sistema numérico posicional de base dos**, que no es otra cosa que los números 0 y 1 representando cifras de una manera convencional, algo distinta de la habitual.

El **sistema de base dos, o binario**, como sabemos, consta de sólo dos dígitos, **0** y **1**. A pesar de tener tan pocos dígitos, es tan poderoso como el sistema decimal habitual, en el sentido de que podemos servirnos de él



para contar conjuntos de cualquier cantidad de elementos, misión fundamental de todo sistema numérico que se precie.

En **sistema decimal**, los elementos de un conjunto se cuentan del siguiente modo: a cada elemento del conjunto a contar lo identificamos con un dígito de los diez existentes en el sistema...

ELEMENTOS	DIGITOS DEL SISTEMA
ninguno	0
un elemento	1
dos elementos	2

Vamos apareando dígitos con elementos, y si tenemos menos elementos por contar que dígitos, cuando agotamos los elementos del conjunto, el último dígito apareado indica la cantidad de elementos del conjunto:

cuatro elementos 4

Si, por el contrario, se acaba primero la lista de dígitos y quedan aún elementos por contar,

nueve elementos y faltan... 9

...entonces volvemos a repetir la lista de dígitos apareándolos con los elementos que faltan, pero indicando que hemos agotado la lista de dígitos una primera vez agregando a la izquierda el primer dígito significativo de la lista, que es el 1.

10
11
12
etc.

Y si agotamos la lista de dígitos una vez más y quedan elementos por contar, lo indicamos reemplazando el "1" por el siguiente dígito en la lista, y continuamos con la cuenta repitiendo la lista desde el principio.

19
20
21
etc.

Si agotamos la lista tantas veces como dígitos tenemos,

97
98
99

...hacemos lo mismo que antes pero un lugar o **posición** más a la izquierda:

100
101
102
etc.

Por supuesto, esto es conocidísimo de todos nosotros; pero nos interesa llamar la atención sobre esta manera de contar porque *se aplica en forma absolutamente igual a todos los sistemas numéricos posicionales*.

Es decir que si, en particular, nuestra lista de dígitos consta sólo del 0 y el 1, la manera de contar será la misma:

ningún elemento..... 0
uno.....1 (se agotó la lista)
dos.....10
tres.....11 (se agotó una segunda vez y además no quedan dígitos para colocar a la izquierda)
cuatro 100
cinco .....101 (se agotó la lista)
seis.....110
siete.....111 (se agotó una cuarta vez)
ocho .....1000 ...etc.

Hemos representado los números del 0 al ocho en **sistema binario**, y es evidente que nada nos impide seguir contando elementos del conjunto, sencillamente agregando un dígito "1" a la izquierda cada vez que sea necesario.

4.3. Fórmulas de conversión binario/decimal

4.3.1. Claro está que si se desea representar el número –digamos– 75 en base dos, no hay necesidad de contar de uno en uno hasta llegar a él; más bien, recordemos la vieja *"receta" para pasar un número decimal a su expresión binaria*:

75 | 2	Lo dividimos por 2 y	
1 37	anotamos el resto:	1
37 | 2	Dividimos el cociente	
1 18	por 2 y anotamos el resto al lado del resto anterior:	11
18 | 2	Dividimos el cociente	
0 9	por 2 y anotamos el resto al lado de los anteriores:	011
9 | 2	...etc...	1011
1 4		
4 | 2	...etc...	01011
0 2		
2 | 2	...hasta que obtenemos el último resto y el último cociente posibles y los anotamos al lado de los restos antes obtenidos:	1001011
0 1		

Conclusión: *la representación binaria del número 75 decimal es 1001011.*

4.3.2. Recordemos también la *"receta inversa"* que, dado un número binario, nos permite averiguar su valor en escritura decimal:

Sea dado el número en base dos "1001011", y numeremos sus dígitos de derecha a izquierda, a partir de cero.

1 0 0 1 0 1 1

6 5 4 3 2 1 0

Tomemos todos los números de la fila anterior que correspondan a dígitos "1" del número binario dado, a saber, 6,3,1, y 0.

1 0 0 1 0 1 1

6 5 4 3 2 1 0

Y efectuemos la suma de las potencias de 2 afectadas a esos exponentes:

$2^6+2^3+2^1+2^0=$
$=64+8+2+1=75$

...que, coherentemente con lo recién visto, es la representación decimal de 1001011 en base 2.

4.4. Binarios y bytes

Esta aritmética elemental binaria que hemos visto nos permite sospechar que la descripción "de entrecasa" que hemos hecho de los bytes encaja muy bien con lo que es en realidad un número en base dos de ocho dígitos; la analogía, a esta altura, no puede menos de ser clara. Volvamos a los "hechos importantes" del principio de esta charla:

1. Un byte consta de ocho bits significa que por cada configuración que podamos concebir de "bits encendidos" y "bits apagados" existe un número en base dos de ocho dígitos escritos con "unos" donde haya bits "encendidos" y "ceros" donde los haya "apagados".

2. En un byte pueden escribirse números entre 2 y 255 corresponde al hecho de que con ocho dígitos binarios –todos ceros– obtenemos el número cero, el menor de ellos, y con "todos unos" estamos escribiendo en realidad el número 255 en base dos, como podemos verificar haciendo la conversión, tal cual se indicó más arriba, de decimal a binario.

3 y 4. Números a un byte: dos dígitos hexadecimales, y números a dos bytes: cuatro dígitos hexadecimales, son afirmaciones importantes porque nos permiten manejar de manera *cómoda y estándar*, atendiéndonos de modo fiel a la estructura de la memoria de nuestra computadora y de los registros del microprocesador, números que sean manejables por este último.

4.4. INC, DEC

Un sencillo programa como el que sigue puede aclarar lo recién visto definitivamente:

LD B,0	Cargamos **B** con **0**
LD C,FF	Cargamos **C** con **255**(**11111111** en base dos)
INC C	Aumentamos el contenido de **C** en **1**
RET	Volvemos a BASIC

La lógica indica que, como **C** es un registro cuya estructura física coincide con los ocho bits que reunidos conforman un byte, no puede contener un número superior a **255**. Hemos cargado **C** con **FFH**; aumentarlo en **1** (**INC C**) significa, al menos en teoría, cargarlo con **100H**. Pero si los números a un byte se expresan mediante **dos** dígitos hexadecimales, hay algo que sobra en los **100H** que se supone contiene ahora **C**.

Si ordenamos **PRINT USR 16514** luego de ensamblar y cargar el programa anterior en una línea **REM**, lo que se presenta en pantalla es el contenido de **BC** al momento de retorno a BASIC, número obtenido multiplicando el contenido de **B** (byte de orden alto) por **256** y sumándole el de **C** (byte de orden bajo).

Ejercicio: interpretar el resultado que se ve en pantalla.

La instrucción **INC C** antes utilizada es nueva a los lectores de estas notas; pero aquellos despiertos habrán deducido inmediatamente su significado, y aquellos más despiertos aún habrán recurrido inmediatamente a la **tabla de códigos del Z-80** que se halla en el manual de nuestra computadora. De todos modos, agreguemos a la lista de instrucciones y nemónimos dada antes lo siguiente:

Codif. decimal	Hexa decimal	Instrucción	Efecto
04	04H	INC B	Incrementa el valor de **B** en **1**
12	0CH	INC C	Incrementa el valor de **C** en **1**
05	05H	DEC B	Decrementa el valor de **B** en **1**
13	0DH	DEC C	Decrementa el valor de **C** en **1**

Sprites en C-64

Softi: sigue dándonos ideas para optimizar los programas en C-64

Veremos un pequeño pero muy interesante programa; basta con cargarlo a la máquina y colocar **RUN** para quedar sorprendido.
Este programa pone en movimiento sobre la pantalla un sprite, esto no es nada extraordinario, lo que es increíble en nuestra C-64 es que el sprite sigue moviéndose en la pantalla **aunque el programa se detenga, se borre** y como si eso fuera poco, dicho sprite seguirá su vuelo si cargamos y ejecutamos tro programa en Basic.
A modo de ejemplo, hemos definido el sprite con las iniciales de **Programación Popular,** pero de todos modos, están invitados a cambiar la codificación del sprite, dicha codificación se ubica en las líneas **100, 101, 110, 111, 120** y **121** del programa. Hay que tener sumo cuidado al redefinirlo, ya que **un solo error** en la cantidad de datos, es suficiente como para que no funcione el programa o hasta llegar a destruirse por completo. De hecho es conveniente copiarlo tal cual como está, probarlo, y si funciona grabarlo, en caso contrario revisarlo (puede suceder que si el programa se corrió y no funcionó por un error de algún data o algún **POKE,** una vez corregido tampoco funcione; en dicho caso se deberá apagar la **C-64** y encenderla para copiar nuevamente el programa.). Una vez grabado, se puede comenzar a recodificar el sprite; recuérdese que la cantidad de datos del sprite es de **63** ya que el sprite está formado por una matriz de **24 columnas** (3 segmentos de 8 bits) y **21 filas.**
Para los "manos sueltas" (y ya que estamos para los demás también) que a todo **POKE** que encuentran le cambian la numeración para observar lo que sucede, eso está muy bien, pero antes de dar rienda suelta a los dedos y meter la pata en algo, **Softi** nos recomienda cambiar la velocidad del sprite.
Prestemos atención:
En la línea **40** hay un poke que está en la posición **252,** aquí es donde le damos velocidad a nuestro **OVI (Objeto Volador Identificado)** de la siguiente forma:
40 POKE 251,5: POKE 252,3
El **3** del último **POKE** es la velocidad del **OVI;** si deseamos disminuir la velocidad, deberemos aumentar el valor, por ejemplo a **55** (bastante lento)
40 POKE 251,5:POKE 252,55
En caso de querer aumentar la velocidad, reduzcamos el valor; **cuidado,** el valor mínimo es **1** y el máximo (más lento) es **255.**
Sí, **Softi** se preparó para todo y para todos; el color del **OVI,** se lo damos en la línea **21,** recordemos que hay 16 colores y van del **0** al **15.**

21 POKE V + 46,01; POKE V +14,0 :POKE V+15,120

+En el primer **POKE** tenemos un **01** lo cual nos indica que el color del **OVI** es **blanco.**

El juego consiste ahora en encontrar aplicaciones a esta rutina... **a cargo de los lectores.**

```
READY.

  1 REM  ******************************
  2 REM  **                          **
  3 REM  **  I N I E R R U P        **
  4 REM  **                          **
  5 REM  **   PROGRAMACION POPULAR   **
  6 REM  **                          **
  7 REM  **  C O M M O D O R E  6 4  **
  8 REM  **                          **
  9 REM  ******************************
 10 FOR T=12736 TO 12798:READ A
 11 POKE T,A:NEXT
 20 V=53248:POKE2047,199:POKEV+21,128
 21 POKE V+46,01:POKEV+14,0:POKEV+15,120
 30 POKE V+29,128
 40 POKE 251,5:POKE 252,3
 54 REM
 55 REM
 56 REM  DEFINICION DEL SPRITE
 57 REM
100 DATA 128,0,8,60,0,0,34,0,0,34,0,0
101 DATA 60,0,0,32,0,0,33,0,0,34,128,0
110 DATA 01,0,0,0,0,0,0,0
111 DATA 60,0,0,034,0,0,34,0,0,60,0
120 DATA 0,32,0,0,33,0,0,34
121 DATA 128,0,1,0,0,0,0,0,0,0,128,0,8
400 REM
500 REM  ***  ESTA ES LA RUTINA  ***
501 REM
510 FOR T=49152 TO 49229:READ A
511 POKE T,A:NEXT:SYS 49152
520 DATA 120,169,13,141,20,3,169,192
521 DATA 141,21,3,88,96,198
522 DATA 251,165,251,240,3,76
530 DATA 74,192,173,16,208,41,128,240
531 DATA 23,173,14,208,201
532 DATA 101,208,31,173,16,208
540 DATA 37,127,141,16,208,169,0,141,14
541 DATA 208,76,67,192,173
542 DATA 14,208,201,255,208
550 DATA 8,173,16,208,5,1,88,141,16,208
551 DATA 238,14,208,165,252,133
552 DATA 251,76,49,234,0

READY.
```

AHORA SI... TENEMOS DISPONIBLES TODOS LOS NUMEROS ATRASADOS DE P. P.



SUSCRIPTORES GANADORES DEL CONCURSO DE PP NUMERO 9

Slemenson, Martín **Bariloche**
una computadora CZ2000 (Spectrum) de 48 donada por Czerweny Electrónica SA

Camino, Roberto **Viedma, Río Negro**
Ganó un lápiz óptico DEC para Commodore 64 donado por DEC Electrónica
El sorteo fue realizado el 15/2/86 ante Escribano Público
FELICITACIONES!!!

☑ $ 1,80 c/u





Tomo 1 P.P
Revista N° 1 al 6
encuadernadas para consultas y colección

☑ $ 9,90

Casetes de los programas de P.P. de los N° 1 al 6

Máquinas	Cód.	P.P
C-64	1	1/6
Spectrum	2	1/3
TK 90X	3	4/6
TS - 2068		
CZ 1000	4	1/3
CZ 1500	5	4/6
TK 83/85		
TI 99/4A	6	1/6

☑ $ 5,30 C/CASETE

SOLO TIENES QUE COMPLETAR Y ENVIAR SIN FRANQUEO EL TALON

Graficando en MSX

Como ya se ha informado a los lectores de **Programación Popular**, el **MSX Basic de Microsoft** es un standard al que ya adhieren más de 20 marcas distintas de computadoras, como es el caso de **Talent**, la primera en Argentina.

Esta pareja de programas permitirá la graficación de un conjunto de datos cuantitativos siguiendo los dos esquemas más conocidos de graficación: en forma de **"tortas"** y de **"barras"**.

Ambos programas piden el ingreso de los nombres que deseemos asignar a las variables en estudio, acompañados de sus magnitudes (el requerimiento de este dato se hace con el mensaje **"Importe"**, aunque ambos programas pueden ilustrar, siguiendo el mismo algoritmo, cualquier conjunto de datos, no limitándose a estadísticas de tipo financiero).

Cada variable se representa en la salida de impresión mediante un color diferente, pudiendo incluirse **hasta diez variables** en cada graficación.

Cabe destacar que un gráfico de "barras" se presta muy bien para destacar la **diferencia en magnitud** entre los "importes" asignados a las variables, mientras que un gráfico "de torta", más bien, permite aclarar la **contribución relativa**, en términos de **porcentajes** de cada una de las variables bajo estudio.

El programa **"Gráficos en barras"** admite además como opción la salida **impresa por plotter color**.

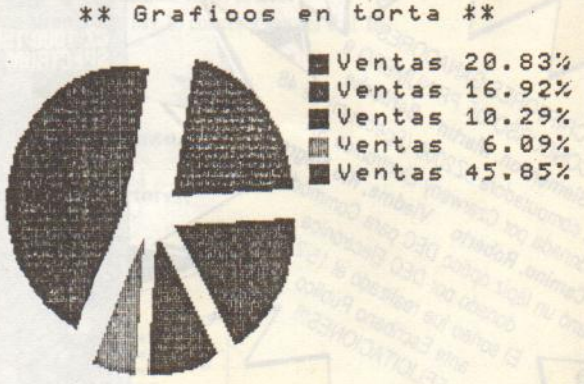
por Gustavo Hidalgo y Hugo Caro.

Gráficos en torta - Estructura del programa

80	Definición de la función R#que interviene en cálculos de coordenadas de impresión.
90-100	Preguntan cantidad de variables a graficar.
110	Dimensiona vectores.
120-180	Reciben los nombres e importes de cada variable.
190-310	Inicializan variables **Basic**, calculan suma de importes y porcentaje representado por cada uno.
320-470	Grafican y pintan el sector circular correspondiente a cada variable.
480-520	Imprimen colores de referencia.
530	Mantiene imagen hasta pulsar **RETURN**.

Gráficos en torta - Tabla de variables

A$()	Vector de nombres de las variables.
AN	Angulo central de cada sector a representar.
C()	Vector que contiene códigos de colores.
CA%	Cantidad de variables a graficar.
C$	Recibe transitoriamente los códigos de colores.
DA()	Vector de importes atribuidos a las variables.
I$	Recibe los importes de las variables.
N$	Recibe temporariamente los nombres de las variables.
S	Suma acumulada de los importes.
SA,SD	Determinan ángulo inicial y final de cada sector a graficar.
X,Y	Coordenadas vértice de referencias de colores.
XO,YO	Coordenadas vértice de cada sector.
X1,Y1	Punto interior a cada sector que utiliza la sentencia **PAINT**.




```
10 '
20 ' GRAFICOS EN TORTA
30 ' Revista Programacion Popular
40 ' PERMITE LA GRAFICACION DE HASTA
50 ' 10 VARIABLES EN DIFERENTES
60 ' COLORES
61 ' AUTOR: HUGO D. CARO
62 '
70 SCREEN 0
80 DEFFNR#(X#,Y%)=SGN(X#)*ABS(INT(
(X#*10^Y%)+.50001)/10^Y%)
90 CLS:PRINT TAB(10);"GRAFICOS EN TORTA":
LOCATE 1,3:INPUT"Cantidad de variables: ";CA%
100 IFVAL(CA$)>10ORVAL(CA$)<2 THEN90ELSECA%=VAL(CA$)
110 DIM A$(CA%),DA(CA%),C(CA%),D(CA%)
120 FORI=1TOCA%
130 CLS:LOCATE 1,3:PRINT"Nombre ";I;:INPUT N$:PRINT
140 I$="":LOCATE 1,5:PRINT"Importe ";I;:INPUT I$
150 IFVAL(I$)<1 THEN140
160 READ C$
170 A$(I)=N$:DA(I)=VAL(I$):C(I)=VAL(C$)
:N$="":I$="":C$=""
180 NEXTI
190 PI=3.1415926#
200 FORJJ=0TO500:NEXTJJ:COLOR1,15,1
210 SCREEN 2:OPEN"grp:"FOROUTPUTAS#1
220 PRESET (34,5)
230 PRINT#1,"** Graficos en torta **"
240 S=0
250 FORI=1TOCA%
260 S=S+DA(I)
270 NEXTI
280 FORI=1TOCA%
290 DA(I)=INT(DA(I)/S*10000)/100
300 NEXTI
310 S=-1.37:SA=S
320 FORI=1TOCA%
330 S=S+FNR#(DA(I)*PI*2/100,2):SD=S
340 IF SD>0 THEN SD=SD-(PI*2)
350 IF SD=0 THEN SD=-1E-05
360 IF I=CA%THEN SD=(PI/-2)+.2
370 SD=FNR#(SD,2):SA=FNR#(SA,2):SK=SA
380 IF SK>SD THEN SK=SK-6.08
390 AN=(ABS(SD)+ABS((SK-SD)/2))*-1
400 X=INT(COS(AN)*60)
410 Y=INT(SIN(AN)*60)
420 X0=65+INT(X/5):Y0=96+INT(Y/5)
430 CIRCLE(X0,Y0),60,C(I),SD,SA,1.15
440 X1(I)=X0+INT(X/2)
450 Y1(I)=Y0+INT(Y/2)
460 IF DA(I)>3 THEN PAINT(X1(I),Y1(I)),C(I)
470 SA=SD:NEXTI
480 FORI=1TOCA%:X=15+(I*12):Y=134
490 LINE(Y,X)-(Y+8,X+8),C(I),BF
500 N$=LEFT$(A$(I)+SPACE$(7),7)
510 PRESET (147,X):PRINT#1,N$;USING"##.##%";DA(I)
520 NEXTI
530 I$=INPUT$(1):IFI$<>CHR$(13)THEN530
540 COLOR 15,4,4:END
550 DATA 10,3,2,12,5,14,6,8,9,7
```

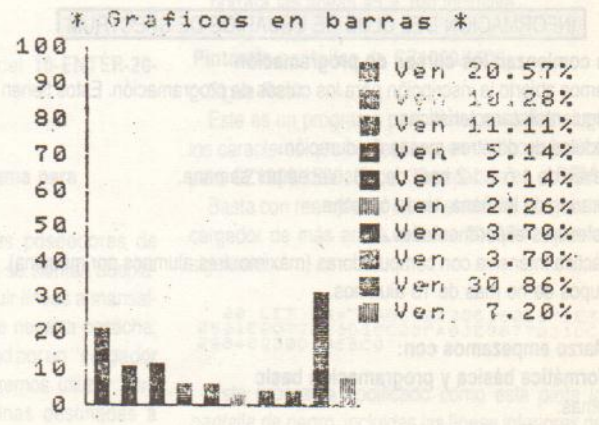
*Con estos dos programas "softi-utilitarios" abre el fuego en lo tocante a software para la nueva "vedette" del mercado argentino, la **DPC-200 MSX** de **Talent**.*

Gráficos en barras - Estructura del programa

100	Pregunta cantidad de variables a graficar.
120	Dimensiona vectores.
130-180	Piden nombres e importes de las variables.
190-230	Inicialización.
240-270	Preparan escala del gráfico.
280-340	Calculan suma de importes y porcentajes.
350-390	Dibujan las barras que representan a cada variable.
400-410	**DATA**s que contienen códigos de colores y de posición en pantalla.
420-460	Completan el gráfico con referencias color y nombres/porcentajes.
470-490	Mantienen imagen hasta pulsar **RETURN** y deciden fin de programa o pasar a rutina de ploteo en papel.
500-730	Rutina de ploteo en papel.

Gráficos en barras - Tabla de variables

A$()	Vector de nombres variables.
C()	Vector de códigos de colores.
CA%	Cantidad de variables
CT,ST	Posicionan la impresión en pantalla.
DA()	Importes de las variables.
I$	Recibe temporariamente los importes.
I,JJ	Variables de control en varios lazos.
N$	Recibe temporariamente los nombres de las variables.
P	Calcula porcentajes.
S	Suma de los importes.
SN$	Bandera "si/no" para decidir pasar a plotter
X,Y	Coordenadas vértice base de barras y referencias.




```
10 '
20 ' Graficos en barras
30 ' Revista Programacion Popular
40 ' Permite la graficacion de hasta
50 ' 10 variables en diferentes
60 ' colores
70 '
80 SCREEN 0
90 DEFFNR#(X#,Y%)=SGN(X#)*ABS(INT((X#*10^Y%)
+.50001)/10^Y%)
100 CLS:PRINT TAB(10);"Graficos en barras":LOCATE
1,3:INPUT"Plotter (S/N)";SN$:L
OCATE 1,5:INPUT "Cantidad de variables:";CA$
110 IFVAL(CA$)>10ORVAL(CA$)<2
THEN100ELSECA%=VAL(CA$)
120 DIM A$(CA%),DA(CA%),C(CA%),D(CA%)
130 RESTORE 410:FORI=1TOCA%:READ SK%
140 CLS:LOCATE 1,3:PRINT"Nombre ";I;:INPUT N$:PRINT
150 I$="":LOCATE 1,5:PRINT"Importe ";I;:INPUT I$
160 IFVAL(I$)<1 THEN150
170 A$(I)=N$:DA(I)=VAL(I$):N$="":I$="":C$=""
180 NEXTI:RESTORE400:FORI=1TOCA%:READC(I):NEXTI
190 PI=3.1415926#
200 FORJJ=0TO500:NEXTJJ:COLOR1,15,1
210 SCREEN 2:OPEN"grp:"FOROUTPUTAS#1
220 PRESET (34,5)
230 PRINT#1,"* Graficos en barras *"
240 LINE(40,20)-(40,180),13
250 LINE(40,180)-(160,180),13
260 FORI=164TO20STEP-16:LINE(35,I)-(40,I),1:NEXTI
270 FORI=1TO11:PRESET(10,(193-(I*16))):PRINT#1,
USING"###";(I-1)*10:NEXT
280 S=0
290 FORI=1TOCA%
300 S=S+DA(I)
310 NEXTI
320 FORI=1TOCA%
330 DA(I)=INT(DA(I)/S*10000)/100
340 NEXTI
350 FORI=1TOCA%:Y=SK%+(I*120/CA%)
360 P=160*DA(I)/100:X=180-P
370 LINE(Y+20/CA%,X)-(Y+80/CA%,180),C(I),BF
380 NEXTI
390 GOTO420
400 DATA 2,3,5,6,7,8,9,10,11,12
410 DATA 0,-19,1,11,17,21,24,26,29,30
420 FORI=1TOCA%:X=15+(I*12):Y=160
430 LINE(Y,X)-(Y+8,X+8),C(I),BF
440 N$=LEFT$(A$(I)+SPACE$(3),3)
450 PRESET (176,X):PRINT#1,N$;USING" ##.##%";DA(I)
460 NEXTI
470 IF INKEY$<>"" THEN 470
480 I$=INPUT$(1):IFI$<>CHR$(13)THEN480
490 IF SN$<>"S" OR SN$<>"s" THEN COLOR 15,4,4:END
500 RESTORE730:FORI=1TOCA%:READCT(I),ST(I):NEXTI
510 LPRINTCHR$(18):LPRINT"C0":LPRINT"S2":LPRINT"P
* Graficos en barras *"
520 LPRINT"A":LPRINT CHR$(18):LPRINT"S0"
530 LPRINT"I"
540 LPRINT"M50,0"
550 LPRINT"J0,-300,500,0":LPRINT"H"
560 FORI=1TO10:X=(I-1)*-30
570 LPRINT"M35,";X:LPRINT"J15,0":LPRINT"M10,
";X-5:LPRINT"P";USING"###";(11-I)*10
:NEXTI:LPRINT"M10,-300":LPRINT"P";USING"###";0
580 FORI=1TOCA%:X=55+((I-1)*43)
590 LPRINT"C";CT(I):LPRINT"M";X;",-300"
600 Y=300*DA(I)/100
610 LPRINT"J0,";Y:LPRINT"J33,0":LPRINT"J0,";-Y
620 GOSUB 710:NEXTI:LPRINT"A":LPRINTCHR$(18)
630 LPRINT"I":LPRINT"S1"
640 FORI=1TOCA%:LPRINT"C";CT(I)
650 Y=(I-1)*-25:LPRINT"M10,";Y
660 LPRINT"J0,20,20,0,0,-20,-20,0"
670 GOSUB720:LPRINT"M40,";Y:N$=LEFT$
(A$(I)+SPACE$(10),10)
680 LPRINT"P":N$;USING"##.## %":DA(I)
690 NEXTI:LPRINT"A"
700 GOTO480
710 FORJ=(-300+Y)TO-300STEP-ST(I):LPRINT"M";X;",
";J:LPRINT"J33,0":NEXTJ:RETURN
720 FORJ=1TO20STEPST(I):LPRINT"M10,";Y+J:LPRINT"
730 DATA 0,3,1,3,2,3,3,3,0,5,1,5,2,5,3,5,0,7,1,7
J20,0":NEXTJ:RETURN
```

CLUB DE USUARIOS DE SPECTRUM.

INFORMACION DEL CLUB DE USUARIOS DE SPECTRUM

Ya comienzan los cursos de programación

Hemos abierto la inscripción para los cursos de programación. Estos tienen las siguientes características:

* Módulos de dos-tres meses de duración.
* Clases de 1 ó 1 1/2 hora de, dos veces por semana.
* Horarios de mañana, tarde ó noche.
* Profesores experimentados.
* Práctica intensiva con computadoras (máximo: tres alumnos por máquina).
* Grupos de no más de 15 alumnos.

En Marzo empezamos con:

* **Informática básica y programación basic**

Temas:

* Principios de Informática.
* La computadora y sus unidades.
* La instrucción y el programa.
* Diagramación-Simbología
* Constantes-Operaciones-Variables.
* Entrada y Salida
* Comandos del Sistema Operativo.
* Transferencias de Control.
* Operadores lógicos.
* Funciones de biblioteca.
* Graficación.
* Color.
* Sonido.

Para información y reserva de turnos, llamar por teléfono al 34-8135/8357 ó personalmente en nuestra sede de Avda. de Mayo 749 Piso 3 Of. 15. Capital.

Recorte y envíe este cupón al:
CLUB USUARIOS DE SPECTRUM
Av. de Mayo 749, Piso 3, Of. 15
(1084) Capital. Tel. 34-8357/8135

Nombre y Apellido:

Domicilio:

Código Postal: **Localidad:**

Teléfono: **Edad:**

Computadora (marca y modelo):

TEMAS DE SOFT

Lenguajes de Programación para nuestras máquinas:

Todos los equipos de la línea Spectrum y parientes menores traen un intérprete Basic incorporado en su ROM. Por consiguiente son programables directamente en dicho lenguaje. Para quien desee mayor velocidad operativa está el recurso del lenguaje ensamblador, que requiere de la carga previa de un programa de ensamble en un sector protegido de la RAM. Pero si el lenguaje, además de servir a los fines de la programación propiamente dicha, pretendemos que cumpla funciones didácticas, debemos pensar en una estructura de tipo Logo. Se ha escrito bastante material sobre las bondades de este idioma. Pero, siendo lo más completo que se ha desarrollado en este campo, creemos que adolece de ciertas fallas, especialmente cuando se trata de aplicarlo en la enseñanza de chicos menores de 9 años. Existe una adaptación de este lenguaje para equipos de 16 K o menos, que supera algunos de los defectos del Logo, por ejemplo, la necesidad de direccionar a la tortuga respecto de ejes que le son propios, es decir, "giran con ella".

Este nuevo lenguaje de programación, aún muy pobremente difundido, lleva el nombre de CREADOR y posee características que lo hacen sumamente interesante. Lo ha utilizado la Secretaría de Educación de la Municipalidad de la Ciudad de Buenos Aires, en los cursos de capacitación del equipo de Informática Educativa. Se puede convertir en una interesante herramienta de trabajo para las escuelas, dado que el costo de los equipos necesarios para su utilización es de los más bajos posibles.

En próximos números de **Programación Popular**, mostraremos algunos ejemplos de la aplicación de este lenguaje y profundizaremos un poco más en sus características. (no solo de Basic vive la computadora)

CAMBIOS DE BASE. Transformaciones de números definidos según diferentes sistemas de numeración.

Este pequeño programa resuelve el traspaso de un número decimal racional (entero o fraccionario) a cualquier sistema de numeración comprendido entre el binario (base 2) y el hexadecimal (base 16). Dado que la Spectrum trae la sentencia función BIN, el traspaso de un número en base 2 a sistema decimal resulta inmediato. Por eso nos parece de utilidad este aporte, aclarando que ha sido realizado ante la inquetud de algunos amigos que encontraron en alguna publicación extranjera, unas rutinas de cálculo que tendrían que realizar la transformación planteada, pero que no han funcionado correctamente.

El programa admite cualquier número racional decimal (positivo o negativo, con o sin fracción decimal).

Esperamos que sea de utilidad especialmente para los aficionados al código máquina y a los amigos de los cálculos matemáticos.

```
10 PRINT "CLUB DE USUARIOS D
E SPECTRUM"
20 PRINT
30 PRINT "CAMBIO DE BASE DECIM
AL, A BASES ENTRE 2 Y 16,INCLUSI
VE."
40 INPUT "numero decimal:";n;"
base nueva:";d: LET sgn=0
50 IF d<2 OR d>16 THEN PRINT A
T 11,13;"ERROR": PAUSE 75: CLS
GO TO 10
55 IF n<0 THEN LET n=n*(-1): L
ET sgn=1
60 LET m$="": LET z$="": LET n
um=INT n
70 IF num<d THEN GO TO 110
80 LET coc=INT (num/d): LET re
s=num-coc*d: LET r$=STR$ res
90 IF res>=10 THEN LET r$=CHR$
(55+res)
100 LET m$=r$+m$: LET num=coc:
GO TO 70
110 LET r$=STR$ num: LET m$=r$+
m$
120 IF INT n=n THEN GO TO 190
130 INPUT "cuantas cifras desea
como maximo despues de la coma
?";d
140 LET fr=n-INT n
150 FOR w=1 TO d
155 IF fr=0 THEN GO TO 190
160 LET fr=fr*d: LET h=INT fr:
LET h$=STR$ h
170 IF h>=10 THEN LET h$=CHR$ (
h+55)
180 LET z$=z$+h$: LET fr=fr-INT
fr: NEXT w
190 IF sgn=1 THEN LET m$="-"+m$
: LET n=n*(-1)
200 PRINT "numero decim
al:";n: PRINT "numero en base
";d;":";m$: IF z$="" THEN STOP
210 PRINT ".";z$
```



SOFTI CLUB

Desde **Pinamar**, nuestro veraniego amigo **Raúl D. Lambertucci** nos envía a modo de colaboración espontánea, un sinfín de *"chiches"* que harán las delicias de los poseedores de **CZ-1000/1500 Spectrum** y afines.

Se trata de una verdadera maraña de rutinas útiles, ideas, chismes, etc. que iremos desenredando poco a poco, con la ayuda de **Raúl** y... ¿por qué no? con el aporte de otros lectores.

Borrador de líneas de programa para CZ SPECTRUM/TK90X

Este es un programa **Borrador de líneas** escrito en lenguaje de máquina que ocupa sólo 55 bytes de memoria. Está destinado a las microcomputadoras **CZ-Spectrum** y **TK90X**.

```
  40 REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@     Destructor de lineas      @
@ por Daniel Lambertucci para   @
@ Programacion Popular          @
@ para CZSpectrum y TK-90 x     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

  50 LET RT=(PEEK 23731)*256+PE
EK 23730
  60 CLEAR RT-54
  70 LET PS=(PEEK 23731)*256+PEE
K 23730+1
  80 LET CS=0
  90 FOR N=PS TO PS+54
 100 READ B
 110 POKE N,B
 120 LET CS=CS+B
 130 NEXT N
 140 READ TOT
 150 IF TOT<>CS THEN GO TO 320
 160 PRINT "O.K.CARGADO A PARTIR
DE ";PS
 200 DATA 205,121,28,205,162,45,
58,3,175,32,8,205,162,45,1
 210 DATA 2,0,24,35,96,105,35,20
5,110,25,229,205,162,45,56,3,175
,50,8
 220 DATA 193,1,1,0,24,41,96,105
,205,110,25,84,93,229,205,229,25
,1,0,0,201
 300 STOP
 310 DATA 4642
 320 CLEAR PS+53
 330 PRINT "ERROR EN DATAS
```

El programa en **BASIC** que les envío es un cargador para el borrador de líneas. El código máquina se carga en la memoria a partir de la dirección contenida en la variable **PS**, que el cargador se encarga de informarnos.

Luego de cargar el **BASIC** y ejecutarlo –el programa verificará si hemos cometido algún error de tipeo– basta con hacer

PRINT USR dirección de llamada, número de primera línea a borrar, número de última línea a borrar inclusive.

Ejemplo: **PRINT USR 65134,20,200**

...que borrará del mapa todo el bloque de líneas de programa **BASIC** comprendido entre líneas **20** y **200** inclusive.

Si la primera línea indicada no existiera, se comenzará a borrar desde la siguiente. Si la última no existiera se borrará hasta la inmediata inferior.

Si llamamos a la rutina sin números de líneas a borrar aparecerá mensaje de error "mensaje inválido":

0 cuando el número de la primera línea no es válido, es muy alto o es un dato no numérico.

2 cuando el último número no es válido.

Una vez comprobada y depurada la rutina, para grabarla basta hacer:

SAVE "DELETE" CODE dirección de llamada**,55**

Recuperar con

LOAD "DELETE" CODE

Y olvidarse para siempre del **10-ENTER-20-ENTER-30-ENTER**, ...etc.

Borrador de líneas de programa para CZ1000/1500

Para que nuestros lectores poseedores de **CZ1000/1500** y compatibles no se sientan desplazados, y puedan también destruir líneas a mansalva, incluimos este programa de nuestra cosecha, acompañado en esta oportunidad por un *"cargador universal"* en **BASIC** que podremos utilizar para cargar cualquiera de las rutinas destinadas a **CZ1000/1500** que aquí publicamos.

Los códigos del programa *LM* figuran codificados en *notación hexadecimal* como el contenido de la variable **A$**. Para cargar una u otra rutina bastará con reemplazar la línea **20** que define el string **A$** por la señalada para cada rutina, y hacer **RUN**.

Ubicar en la línea **REM** inicial tantos caracteres como se indique –por lo menos– para cada rutina.

Todas llevan dirección inicial =16514.

Programa cargador (dejar al menos 20 lugares en **1 REM**):

```
  1 REM ++++TANTOS CARACTERES++
+++++++++++++++COMO SEA++++++++
++++++++++++++NECESARIO++++++++
  10 REM
  CARGADOR DE L.M.
PROGRAMACION POPULAR  NO.10
  CZ 1000/1500 - TK 83/85

  20 LET A$="210000CDD809E521000
023CDD809D1CD5D0AC9"
  30 LET K=1
  40 FOR I=16514 TO (16514+LEN A
$/2)-1
  50 LET D=16*(CODE A$(K)-28)+CO
DE A$(K+1)-28
  60 POKE I,D
  70 LET K=K+2
  80 NEXT I
```

Después de hacer **RUN**, para utilizar este programa borremos las líneas del programa cargador dejando sólo la línea **1 REM** donde reside el código máquina.

(*¿Qué cómo se borran?* Pues, por ahora, **10-ENTER-20-ENTER** ...etc. Paciencia.)

Luego agregamos las siguientes líneas **BASIC** que nos ayudarán a utilizar el *LM* cargado, dejando el listado como sigue:

```
  1 REM ++++TANTOS CARACTERES++
+++++++++++++++COMO SEA++++++++
++++++++++++++NECESARIO++++++++
 9990 PRINT "LINEA INICIAL.FINAL?
"
 9991 INPUT Z$
 9992 LET Z=INT VAL Z$
 9993 POKE 16515,Z-256*INT (Z/256
)
 9994 POKE 16516,INT (Z/256)
 9995 LET Z=VAL Z$((LEN STR$ INT
VAL Z$)+2 TO )
 9996 POKE 16522,Z-256*INT (Z/256
)
 9997 POKE 16523,INT (Z/256)
 9998 RAND USR 16514
```

Para utilizarlo, **GOTO 9990** e ingresar los números de línea inicial y final del bloque a borrar *separados por un punto decimal*:

NUMERO DE LINEA INICIAL.FINAL?
"45.100"

...borrará las líneas **45** a **100** inclusive.

Pintando pantallas de CZ1000/1500

Sigue Raúl:

Este es un programa para pintar pantallas –con los caracteres que se deseen– en código máquina para **CZ1000/1500**. Poner 35 espacios en el **REM**.

Basta con reemplazar en el listado del programa cargador de más arriba la línea **20 A$=**... por la siguiente:

```
  20 LET A$="2A0C4023D5160E1F3E8
0511E2077231D15C23F403E8077A31DC
29840230D10E8C9"
```

Este programa codificado como está *pinta la pantalla de negro*, incluidas las líneas inferiores de edición.

Para pintar *de otro color o caracter*, hacer **POKE 16523**, código de caracter y **POKE 16535**, código de caracter.

Debemos usar un código de caracter *imprimible* (fijarse en el manual) y que *no* defina una palabra clave como **PRINT** o **COPY**, etc.

Usese con **RAND USR 16514**.

SCROLL a la izquierda (CZ1000/1500)

Este programa tiene un efecto similar a la instrucción **SCROLL** pero desplazando la pantalla *una columna a la izquierda* en lugar de hacerlo *una fila hacia arriba*. Dejar 23 caracteres en el **REM**.

Reemplazar lalínea 20 del cargador por

```
  20 LET A$="3E182A0C4023545D230
11F00EDB02B3600233DC28740C9"
```

Para usarlo con toda la pantalla hacer

```
 9000 FOR I=0 TO 31
 9010 RAND USR 16514
 9020 NEXT I
 9030 CLS
 9035 REM OJO NO OLVIDAR EL CLS
 9040 RETURN
```

Deseo si es posible, formar parte del Softi Club y también si me pueden responder algunas preguntitas:

1) Hay alguna forma de proteger el teclado de goma

del Spectrum contra posibles borrados de sus teclas por el uso intensivo?
2) Existen en Argentina teclados profesionales para el Spectrum como tienen en España?
3) Me interesaría saber si se puede colocar un interruptor de encendido y apagado en el cable de alimentación del Spectrum. Si fuera posible: Dónde tengo que ubicarlo? en el cable que va del tomacorriente a la fuente? (sería muy incómodo de manejar), o en el cable de la fuente a la máquina?
4) Existe a la venta algún reset para Spectrum que se conecte al bus trasero, al igual que lo hay para Commodore?

Miguel A. Ferrante
Capital

Ya sos socio del Softi Club, *Miguel*.

1) El teclado ya es bastante resistente a agentes externos de por sí, pero si te parece necesario, podés cubrir la consola con una hoja de celofán, aunque el efecto no será muy estético. No nos parece conveniente "jugarse" a recubrir las teclas con ninguna sustancia especial.
2) No tenemos noticia de ello.
3) Efectivamente, podés colocar un interruptor entre equipo y fuente (Ver "Temas de hard", a cargo del C.U.S., PP Nº 9.
4) Hemos visto resets exteriores, por lo común acompañando a algún tipo de interface. Podés averiguar en Informática Caballito, Rivadavia 5611, Local 4, Cap.
Retribuimos tus saludos.

Nuestro amigo Guillermo Fleckenstein, de Merlo, P.B.A., se declara "especialista" en la línea Sinclair y nos envía varias interesantes colaboraciones:

```
  1 REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      Programacion Popular          @
@    Guillermo Fleckenstein          @
@       para CZSpectrum              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

  10 POKE 23613,0: REM **inuti-
liza el BREAK***
  20 POKE (PEEK 23635+256*PEEK 2
3636)+100: REM ***desaparece el
listado!***
```

```
  1 REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      Programacion Popular          @
@    Guillermo Fleckenstein          @
@       para CZSpectrum              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

  10 CLEAR 31999
  20 FOR a=32000 TO 32018: READ
n: POKE a,n: NEXT a
  30 DATA 14,255,6,255,120,211,2
54,16,251,13,32,246,62,7,211,254
,201
  40 RANDOMIZE USR 32000
  50 REM ***la rutina es reubica
ble en cualquier direccion***
  60 REM ...es una sorpresa!...
```

```
  1 REM
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      Programacion Popular          @
@    Guillermo Fleckenstein          @
@       para CZSpectrum              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

 100 FOR n=50000 TO 50053: READ
a: POKE n,a: NEXT n
 110 LIST : RANDOMIZE USR 50000
 120 DATA 33,0,64,17,31,64,5,192
,197,213,229,6,15,126,205,122,19
5,79,26,205,122,195,119,121,18,0
5,27,16,240,225,1,32,0,9,209,235
,9,235,193,16,223,201,197,6,8,78
,203,41,203,23,16,250,193,201
 130 REM *invierte la pantalla!*
```

Muchas gracias, Guillermo...

Mis dudas son:
1. Cómo se conectan los cables de la **TI 99** (es por eso que mandé el antropófagos para **TI 99** no grabado en cassette), cómo me doy cuenta de que está grabando (en el **Spectrum**, coloca rayas en el borde).
2. ¿Se puede conectar una impresora **Alpha-Com 32** a una **"Radio Shack" MC 10-TRS 802?**
3. ¿Sirven los pokes para CZ-1000 o TK 85 (etc.) para la Spectrum?
4. En el número 2 de la revista "Personal Computer" me he enterado que sale un programa para **TI 99/4A** para hacer gráficos en alta resolución. Yo traté de conseguirlo, pero sólo alcancé a conseguir el Nº 3 de esa revista (donde me enteré). ¿Podrían

SERVICE INTEGRAL
COMMODORE
PROYECTOS
ESPECIALES EN
HARD Y SOFT

DEC
ELECTRONICA
&
COMPUTACION

Ofertas especiales de febrero:

–Lápiz optico DEC ₳45*
–Mach 5, cartridge, manual y disco ₳50
–Jane para C 128 ₳27
–Emulador de Spectrum para Commodore 64 ₳25
y muchas novedades más!

*Primero en Argentina hecho por argentinos

Zonas libres para distribución en el interior del país

Envíos al interior

Consultas y asesoramientos gratuitos

Unica dirección:
Av. Pueyrredón 1990 - 4º piso "A"
T.E. 83-5241 - Capital Federal

ref. 18

publicarlo Uds.? A más de un lector le va a interesar.
Sin más molestias, los saluda atte.

Claudio Martín Aviani
La Calera - Córdoba

1) La conexión a grabador de la consola se indica en el manual de tu máquina, *Claudio*. De las tres fichas en la otra punta del cable, la más delgada corresponde al control remoto (REM) del grabador.
Algunos grabadores tienen la polaridad de la ficha REM invertida, por lo cual si ves que no controla bien el grabador dejala directamente desconectada, o pedí a algún técnico que invierta la tal ficha.
Los otros cables van respectivamente a EAR y MIC del grabador.
La única indicación de que la máquina está grabando datos es el mensaje correspondiente –no existe retorno visual–.
2) No es posible en forma directa.
3) No, las direcciones de memoria a que se hacen referencia son totalmente distintas.
4) La graficación en alta resolución es un ideal largamente soñado por todos los usuarios de TI-99, *Claudio*. Una vez más invitamos a los lectores a recoger el guante que arrojás.

Yo tengo una CZ1500 y me encantaría saber si hay programas de truco para la misma. Si los hay me gustaría que los publiquen en la revista.
Vivo en Río Colorado (R.N.) y no hay ningún lugar que se dedique a la venta de cassettes p/computadoras ni a las mismas, teniendo que remitirme por cualquier cosa referente al tema a Bahía Blanca.
Quisiera pedirles que me asocien al SOFTI CLUB (si es posible).

Juan M. Etchegaray

No conocemos que se haya publicado algún programa como el que querés, aunque sería por demás interesante. Invitamos a los ingeniosos lectores a devanar su tejido neuronal en pos de tan atractivo programa.
Sos un socio más del Softi Club, *Juan*. Hasta pronto.

Mientras leía el artículo Basic III se me ocurrió que era todo un desafío que Euler hubiera podido a los siete años calcular, "a pura álgebra", la suma de los números naturales de 1 a 100.
Tengo 47 años, así que me pareció fantástico estar cercano al nivel de Euler a los siete, cuando después de muchas vueltas me encontré con la respuesta que les adjunto más abajo.
Seguramente no descubrí nada nuevo, pero me pareció interesante enviársela:

```
  1 REM
  ************************************
  *  SUMA DE NUMEROS NATURALES       *
  *  ROBERTO A.ILLA - LAS FLORES     *
  *  PROGRAMACION POPULAR  NO.10     *
  *  CZ1000/1500-TK83/85  M.M KB     *
  ************************************
 10 PRINT "NUMERO",,"INICIAL?";
 20 INPUT A
 30 PRINT TAB 32-LEN STR$ A;A
 40 PRINT "NUMERO",,"FINAL?";
 50 INPUT B
 60 PRINT TAB 32-LEN STR$ B;B
 65 PAUSE 50
 70 CLS
 80 PRINT "SUMA DE NUMEROS",,"N
ATURALES DE ";A;" A ";B
 90 LET Y=INT (((B**2-B)/2+B)+.
5)-INT ((((A-1)**2-(A-1))/2+(A-1
))+.5)
100 PRINT TAB 32-LEN STR$ Y;Y
110 GOTO 10
```

i = número inicial
f = número final
Suma= $((f^2 - f)/2 + f) - (((i-1)^2 - (i-1))/2 + (i-1))$

Roberto A. Illa
Las Flores, P.B.A.

Tu fórmula es correcta, *Roberto*. Muchas gracias por tu gentileza y felicitaciones.

1° He leído en una revista que la COMMODORE 64 posee una memoria libre para programación en BASIC mucho menor a 64K, me gustaría que me aclaren esto diciéndome cuál es la memoria libre para programación en BASIC que realmente posee esta máquina.

2° Quisiera que me dijeran cómo se esta vendiendo hasta ahora la TALENT MSX y qué futuro le ven en el mercado argetnino.
Me gustaría que me hagan socio del SoftiClub.

Jorge E. Barontini
Gral. Rodríguez

1) Efectivamente Jorge, la memoria libre para programación, o memoria de usuario, queda siendo igual a unos 39Kb.
En todo momento, la cantidad de memoria libre te la puede decir la instrucción FRE().
2) La información que tenemos es, concretamente, que se lanzó una pequeña cantidad de máquinas al mercado y que fue agotada. El producto es bueno, y adscribir al standard MSX lo dota de un futuro promisorio, que dependerá también de la adhesión de otros fabricantes que proporcionen accesorios, sofware, etc., para proveer convenientemente al usuario.
Por supuesto ya sos socio del Softi Club.

QUIERO SABER:
(1) Si hay en el país **Joysticks** de control remoto ,para la **CZ 2000**.
(2) Si hay sistemas de almacenamiento externo (como micro drives, disketes, etc.)
(3) Si hay teclados profesionales.
(4) Y por último, si hay impresoras pequeñas de bajos precios.
PDT.: Me despido atentamente y les deseo un buen año **1986.**

Carlos Nicolás Elosegui
Viedma,
Río Negro

1) Sí; tenés que asegurarte de que sean compatibles con la *Interface II Sinclair* o con la interface particular que vayas a usar.
2) Existen aunque es difícil conseguirlos; se conectan a través de la *Interface I Sinclair*.
3) No conocemos marcas (en nuestro país).
4) Sí, la impresora *Alphacom-32* es de precio razonable; imprime a 32 columnas, incluso caracteres gráficos.
Agradecemos y retirbuimos tus deseos, *Carlos*.

TRUCO
POKEar:
POKE USR 3030, 100 y saldrá un new.
Luego, se vuelve a POKEar y saldrá esta frase CNONSENSE IN BASIC.
Finalmente pulsar ENTER y saldrá el cursor arriba.

Soy poseedor de una **CZ1000** con 16K de RAM y deseo hacerles las siguientes consultas:
–¿Pueden ser utilizadas las conexiones para grabador y expansor de memoria como puertas en serie y en paralelo, mediante la ampliación de circuitos adicionales? En ese caso, ¿se puede conectar a ellas un model o un generador de sonido?
–¿Existe en la **Argentina** algún libro que enseñe lenguaje máquina para la **CZ1000**? ¿El lenguaje máquina de la **SPECTRUM** es el mismo de la **CZ1000**?
Sobre su revista sólo puedo decir que es una de las mejores junto con **K64**.
Como sugerencia, pienso que deberían publicar más programas en CM, más circuitos en hardware, una lista de los últimos equipos y accesorios y un curso de CM.

Miguel Pieszko
Resistencia, Chaco

Como ya dijéramos en otra oportunidad, *Miguel*, todo puede hacerse. Sin embargo, no tenemos noticias de su producción comercial. Quizá algún lector pueda responderte en sentido afirmativo. Con respecto a la literatura sobre LM, podés consultar LINGUAGE DE MAQUINA PARA O TK de *Flavio Rossini*, en portugés pero excelente y entendible, o dirigirte al *Club de Usuarios de Spectrum*, Av. de Mayo 749, piso 3, of. 15. El lenguaje máquina de *Spectrum* es idéntico al de *CZ1000* porque comparten el mismo microprocesador *Z80*, pero la forma de empleo puede variar de una a otra, según la aplicación de que se trate. El curso de CM ya es una realidad en *P.P.*.

PREGUNTA	HECHA POR
¿Cómo hallar la memoria ocupada por un programa en **CZ 1000/1500?**	**Hugo Ayestarán,** Chubut **Pablo Gianera,** C.F.
La respuesta es **PRINT (PEEK 16396 + 256 * PEEK 16397) - 16565**	
El programa **SCROLL** al revés, para **TI-99**, no me funciona. Está escrito en Basic común o Extendido?	**Javier Fornari,** Rafaela, S. Fe.
Se requiere el módulo de Basic Extendido para correrlo, **Javier**	
El progrma **Castle Dungeon**, P.P. N° 6, se detiene con error **3/900. ¿Por qué?**	**Alberto Natán,** C.F.
Verificá que estén las líneas **260** y **860.** A nosotros no nos da problemas.	
Publiquen consejos sobre cómo regular el cabezal del grabador. ¿Recomiendan alguna marca especial de grabador o de cassette?	**Gabriel Reyt,** Santa Fe **Alberto E. Cynomon,** C.F.
El cabezal se regula ajustando o aflojando el tornillo (al que se accede generalmente desde el exterior del grabador) hasta obtener el "pico" máximo de agudos posible. No hace falta emplear cinta ni grabador especial, aunque si usás cinta de metal conviene emplear un grabador preparado para ello.	
Los POKE que hago en mi **TS2068** no producen el efecto que se espera en **Spectrum**. ¿Por qué?	**Gustavo Rigo,** Quilmes, P.B.A.
Porque el sistema operativo de ambas máquinas es distinto y aunque las variables del sistema manejan información en forma similar, las direcciones de memoria de las mismas cambian de una a otra máquina.	
Tengo 12 años y tengo una **CZ1500.** Espero no molestarlos con esta pregunta: ¿Qué es una consola?	**Segundo Cerrato,** C.F.
Se llama así a la unidad constituyente de una computadora que incluye teclado y CPU, y que conforma una configuración mínima que a veces puede ampliarse con periféricos como diskettera, impresora, etc.	
Les pido que me informen si la **TK-2000** es compatible con otras computadoras, en especial con las otras **TK.**	**Daniel Robles,** P. Deseado, S. Cruz
No es así, Daniel. La **TK** maneja el formato Applesoft Basic que no comparten otras computadoras salvo las de la línea Apple y compatibles	
Tengo una **CZ1000** + Pack de 16 Kb. A veces llevando mucho tiempo cargando un programa se altera la pantalla y queda todo bloqueado, y pierdo todo lo teclado. ¿Qué puedo hacer?	**Jorge A. Ríos,** Paraná, E. Ríos
Posiblemente el problema esté en la conexión del pack de memoria que suele aflojarse con el uso. Podés tratar de recubrir la "boca" del pack con una banda de goma bien ajustada para mejorar el contacto (cuidando de no tocar los peines metálicos) o adquirir una bandeja especial que se encuentra en el comercio y que sostiene el conjunto firmemente.	
El programa **Backgammon**, N° 6, para **TI-99**, se detiene con error **SUBPROGRAM NOT FOUND IN 720.**	**Ramón Fernández,** C.F.
Revisá si has tipeado la palabra HCHAR correctamente.	

PREGUNTA	HECHA POR
Tengo un grabador que en lugar de conexión EAR tiene "MONITOR". ¿Puedo enchufar allí el plug para cargar programas?	**P. Gianera,** C.F.
Sí, son equivalentes	
Tengo una **CZ1500** y les hago algunas preguntas: ¿Cómo puedo hacer para escribir en cualquier lugar de la pantalla? ¿Qué impresora puedo conectarle a la computadora?	**Leonardo Ferrer,** San Juan
Revisá el capítulo del manual correspondiente al manejo de la pantalla, especialmente la instrucción **PRINT AT.** Podés conectar una impresora Alphacom-32 directamente al slot trasero.	
Poseo una **TI-99/4A.** ¿Existe en Argentina algún banco de datos hecho con un computador como el mío y en Basic? ¿En qué consiste? ¿Con qué máquinas es compatible la mía, si es que existe?.	**Elías Simacek,** Barranqueras, Chaco.
Existe un "banco de programas" importante en manos del Club de Usuarios de **TI-99**, pero creemos que te referís más bien a una base de datos compartidos por varios usuarios gracias a la intercomunicación mediante modem. Esto se ha hecho para administrar información especializada a nivel profesional, pero no conocemos que se haya aplicado un sistema basado en la **TI-99.** Lamentablemente, respondiendo a tu segunda pregunta, la TI constituye una "isla" en cuanto a compatibilidad con otras máquinas.	
¿Qué diferencias existen entre la **Commodore 16** y la **Commodore 64?**	**Daniel Delgado,** C.F.
Podés consultarlas en el artículo "Galería de retratos" que incluimos en este número de P.P.	
Tengo un Datasette General Electric y hay ciertos programas que entran en el Datasette de Commodore y no en el mío. ¿Por qué?	**M.E. Zacagnino,** Lanús, P.B.A.
Me gustaría saber la diferencia más importante entre el Basic 7.0 de la **Commodore 128** y el Basic 2.0 de **Commodore 64.**	**Gustavo Lerea,** Cdoro. Rivadavia, Chubut.
¿Dónde puedo conseguir un monitor para **C-64?**	**Marcelo Williner,** S. Fe
Estos amigos –que además plantean otras dudas– y todos aquellos en situación similar pueden recurrir a DEC o a Baudio S.A.	
Deseo confirmar si realmente no existe ningún curso de inglés para **TK90 X** (16K) como los hay para otras marcas. ¿Cuál es la dirección del Club de Usuarios de **TK-90?**	**Cristina S.P. de Kleinermann,** C.F.
No lo conocemos; para cualquier consulta o servicio propio de un club de usuarios está el C.U.S., Av. de Mayo 749, 3° of. 15.	
Una última cosita: publiquen alguna vez una fotografía del misterioso contestador de cartas, porque cada vez que leo la sección del Softi Club me queda la intriga de quién podrá ser.	**Maximiliamo W. Spagnuolo,** Quilmes, P.B.A.
El que contesta las cartas, **Maximiliano,** aparece fotografiado en la revista un promedio de 30 veces por número. Descubrilo.	

Vaya nuestro saludo a una gran cantidad de lectores a los que no pudimos hacer espacio en esta edición de Softi-Club. Prometemos formalmente ir contestando sus preguntas y mostrando sus colaboraciones en la medida de lo posible. Un gran abrazo a todos ellos:

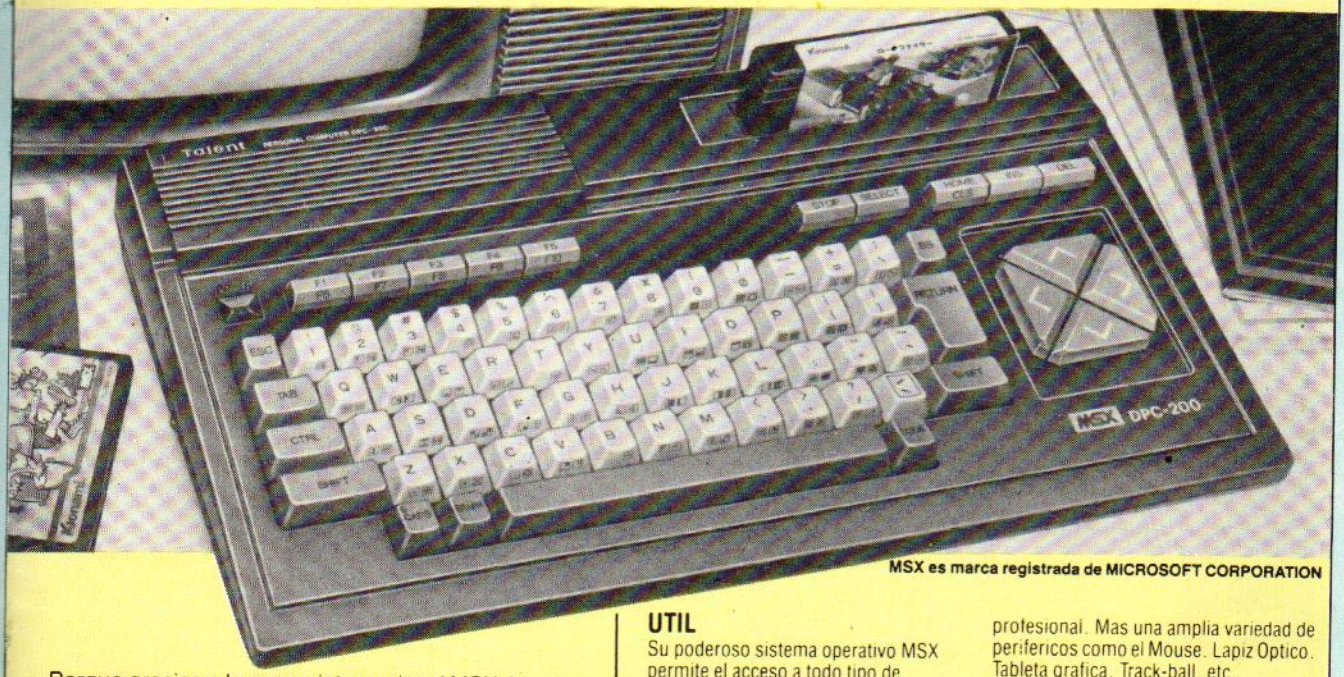
Marcelo D. Otero, La Pampa
Daniel Aquino, Resistencia, Chaco
Carlos M. Hájez, Río Grande, Tierra del Fuego.
Luis Gigiotti, Quilmes, P.B.A.
Claudio Grosman, Santa Fe.
F. Deurer, Plottier, Neuquén.
Luis Della Rocca, Sto. Tomé, Santa Fe.
Pablo Prieto, Ro. Colorado, Río Negro.
Néstor González, Quilmes, P.B.A.
Sergio Sedoff, Oberá, Mnes.
Eduardo Bensi, Gral. Güemes, Salta.
Sebastián Miriuka, Huanguelén, P.B.A.
Pablo Rovner, C.F.
Carlos González, Córdoba.
Gonzalo Urbistondo, C.F.
M. Melnechuk, Posadas, Misiones.
M.A. Scapolla, M. Coronado, P.B.A.
J. Comerce, B. Blanca. P.B.A.
G. Perdiguero, Paraná, E. Ríos.
Aldo J. Rossini, Gral. Güemes, P.B.A.
N.O. Silva, Pcia. R.S. Peña, Chaco
Gustavo Gómez, Córdoba.
Walter Casado, Alta Gracia, Córdoba.
Norberto Moretti, C.F.
Andrés Goldwasser, C.F.
Rodrigo Borgia, Rosario, Sta. Fe.
Gustavo González, Pcia. Chaco.
Mauricio Impallari, F. Varela, P.B.A.
Javier Cencig, C.F.



A la computadora personal

Talent MSX

nada le es imposible



MSX es marca registrada de MICROSOFT CORPORATION

Porque gracias a la norma internacional MSX, la TALENT MSX trasciende sus propios límites.
Hasta ahora, cuando usted compraba una computadora personal de cualquier marca, quedaba automáticamente desconectado del resto del mundo de la computación. Porque los distintos equipos y sistemas no eran compatibles entre sí.
Hasta que dos grandes empresas de informática, la Microsoft Corp. de EE.UU. y la ASCII del Japón se pusieron de acuerdo para crear una norma standard: la MSX. Que se expandió también rápidamente en Europa. Y que hoy TALENT presenta por primera vez en la Argentina.
Mientras que la mayoría de las computadoras de su tipo que se ofrecen en el mercado nacional, han sido discontinuadas por obsoletas en sus lugares de origen, TALENT MSX tiene casi ilimitadas posibilidades de desarrollo. Porque la norma MSX es en todo el mundo inteligencia en crecimiento.
La TALENT MSX pone a su disposición un mundo de software para elegir. Y con la incorporación de todos sus periféricos llega a ser una auténtica computadora profesional.

UTIL

Su poderoso sistema operativo MSX permite el acceso a todo tipo de procesamiento de datos:

- Planillas de calculo.
- Procesadores de palabra.
- Graficos de negocios.
- Bases de datos (d Base II, etc.)
- Contabilidad general, sueldos, y jornales, costos, etc., desarrollados bajo CP/M en Basic, Cobol, Pascal o C.

Con la posibilidad de conexion a linea telefonica permite la transferencia y consulta de datos entre computadoras personales, profesionales o bancos de datos.
La grabacion de archivos es en formato MS-DOS, haciendola compatible con las computadoras profesionales.

DIDACTICA

Dispone de tres lenguajes para la enseñanza de computacion: LOGO como lenguaje de induccion para los mas chicos. Lenguaje de Programacion en castellano, para todos los que quieran aprender a programar sin conocimientos previos. Y Basic MSX como lenguaje profesional. Mas una amplia variedad de periféricos como el Mouse, Lapiz Optico, Tableta grafica, Track-ball, etc.

DIVERTIDA

La mas genial para Video-Juegos. Por la amplisima biblioteca de programas **–todos nuevos–** de la norma MSX en el mundo. Y ademas, el Basic MSX permite al usuario generar sus propios juegos con un manejo tan simple, como solo TALENT MSX puede ofrecer.

CARACTERISTICAS TECNICAS

- Memoria principal 64 KB ampliable hasta 576 KB.
- Memoria de video: 16 KB RAM.
- ROM incorporada de 32 KB con el MSX-Basic de Microsoft.
- Graficos completos, hasta 32 sprites y 16 colores simultaneos.
- Generador de sonido de 3 voces y 8 octavas.
- Conexion para cualquier grabador.
- Interfaz para salida impresora paralela.
- Conectores para cartuchos y expansiones.
- Fuente para 220 V y modulador PAL-N incorporado.

DISTRIBUIDORES AUTORIZADOS: CAPITAL FEDERAL: AMATRIX - Bolivar 173 - ARGECINT - Av. de Mayo 1402 y Rivadavia 11332 - BAIDAT - Juramento 2349 y Araoz 2606 - COMPU PRANDO - Av. de Mayo 965 - COMPUSHOP - Córdoba 1464 - COMPUTIQUE - Córdoba 1111 - E.P. - COMPUTRONIC - Viamonte 2096 - CP 67 CLUB - Florida 683 - L. 18 y 19 - DALTON COMPUTACION - Cabildo 2283 - MICROSTAR - Maipu 191 y Callao 462 - Q.S.P. - Bartolome Mitre 864 - SERVICIOS EN INFORMATICA - Parana 164 - **ACASSUSO:** MICROSTAR - Eduardo Costa 892 - **BOULOGNE:** COMPUTIQUE - CARREFOUR - Bdo. de Irigoyen 2647 y Panamericana - **MARTINEZ:** VIDEO BYTE - H. Yrigoyen 32/40 - **RAMOS MEJIA:** MANIAC - Rivadavia 13734 - **SAN ISIDRO:** F. CORATELLA - Cosme Beccar 249 - **VICENTE LOPEZ:** SERVICIOS EN INFORMATICA - Av. del Libertador 882 - **LA PLATA:** CADEMA - Calle 7 Nº1240 - 01 INFORMATICA : Calle 48 Nº 535 - **MAR DEL PLATA :** FAST - Catamarca 1755 - **BAHIA BLANCA :** SERCOM - Donado 327 - **COMODORO RIVADAVIA :** COMPUSER - 25 de Mayo 827 - **CORDOBA:** AUTO DATA - Paseo Santa Catalina - Local 27 - TECSIEM - Santa Rosa 715 - **MENDOZA:** INTERFACE - Sarmiento 98 - **PARANA:** GARCIA MARIO - Laprida y Santa Fe - **RIO GALLEGOS:** COMPUSER - Alberdi 385 - **RIO GRANDE:** INFORMATICA M & B - Perito Moreno 290 - **ROSARIO:** 2001 COMPUTACION - Santa Fe 1468 - MINICOMP - Maipu 862 - SISOR - Urquiza 1062 - **SANTA FE:** ARGECINT - Peatonal San Martin 2433 - L. 36 - SISOR - Rivadavia 2553 - Local 22 - **VILLA MARIA:** JUAN C. TRENTO - 9 de Julio 80 - **R.O. DEL URUGUAY:** MONTEVIDEO: SISTEMAS CADE - 25 de Mayo 626 - P.2.

```
  10 READ D: RESTORE : CLEAR D-1
: READ P,D: FOR G=0 TO D: READ K
: POKE P+G,K: NEXT G: RANDOMIZE
USR P: READ A,L,B,I: POKE P+L,B:
 RANDOMIZE USR P
  20 LET D=P+I: READ U: LET F$="
": FOR K=1 TO U: READ T: LET F$=
F$+CHR$ T: NEXT K: CLS : PRINT #
1;F$: PAUSE A+A: PRINT
  30 FOR J=1 TO I: GO SUB 40: RA
NDOMIZE USR D: PRINT Q$+Q$+F$: N
EXT J: PAUSE 3*A: RUN
  40 LET F$="": READ Q$: FOR H=1
 TO LEN Q$: LET F$=F$+CHR$ (CODE
 Q$(H)-1): NEXT H: LET Q$=CHR$ 3
2: RETURN
  50 DATA 65000,30,33,0,64,17,1,
64,1,255,23,54,254,237,176,201,1
4,255,6,255,120,211,254,16,251,1
3,32,246,62,7,211,254,201,50,10,
255,14,29,32,127,32,49,57,56,50,
32,80,105,110,99,108,97,105,114,
32,82,101,115,101,97,114,99,104,
32,78,116,100
  60 DATA "!RVF!TVTUP-OP@///","!
QBSB!OP!TVGSJS!NBT!DBSHBOEP","@S
PHSBNBT!QPS!UFDMBEP-","BIP3B!UFD
FT","!","/","!","!!!!SVO.TQFDUSVN","
!","!","SFWJTUB!EF!VTVBSJPT!FO!D
BTFUF/","!","!QFEJMB!DPO!FM!UBMP
O!EF!Q/Q/","P!CVTDBMB!FO!LJPTLPT
","Z!EJTRVFSJBT//"
```