

La
pequeña
gran
puerta

ZX81

sinclair

TS 1000

TS 1500

**Curso de
programación
BASIC**

ANTONIO BELLIDO

LEG **PARANINFO** SA

ZX81 sINclair

Curso de programación BASIC

935.00

ANTONIO BELLIDO

ZX81 sinclair

Curso de programación BASIC

SEGUNDA EDICION CORREGIDA



MADRID 1984

© ANTONIO BELLIDO

Reservados los derechos de edición,
reproducción y adaptación para todos
los países.

IMPRESO EN ESPAÑA
PRINTED IN SPAIN

ISBN: 84-283-1277-X

Depósito Legal: M-40.740-1983



Magallanes, 25 - MADRID (15)

(3-3338)

ALCO, artes gráficas. Jaspe, 34. Madrid-26

Introducción general	7
Comentarios a este curso	9
Esquema de conexiones	11
Teclado: su manejo	12
Algunos consejos previos	13

PRIMERA PARTE

Manejando el ZX81	15
Una presentación formal	17
Algunos de mis símbolos	18
Código de errores	19
Lección de inglés	21
¿El casete... para qué?	22
 Programando en BASIC con el ZX81	25
Segunda lección de inglés	27
De no saber nada a saber algo	28
Modificando un programa	32
Haz tus programas claros	34
Jugando con las palabras	37
Dominando la pantalla	39
¡Vete a ...!	45
Recordando de todo un poco	47
Si quieres, entonces decido	49
De aquí para allá y vuelta a empezar	53
Subrutinas	58
¿Arrays?	61
Array de caracteres	62
Arrays numéricos	69

Subcadenas	74
Algunos consejos posteriores	75
Otros comandos y funciones	76

SEGUNDA PARTE

Practicando el BASIC con el ZX81	79
Ejercicios y soluciones	81
Modelo de impreso para desarrollar tus propios programas	127

Introducción general

Cualquier ordenador necesita "programas" que le digan *qué debe hacer* y *cómo debe hacerlo*. Esto quiere decir que, cuando usted se pone frente a una máquina de este tipo, se le ofrecen dos opciones:

1. Coger un programa ya confeccionado y "dárselo" al computador.
2. Desarrollar un programa y volver al punto anterior.

De aquí se puede deducir que es posible manejar un ordenador sin saber programarlo, y esto es cierto si lo que se pretende de él son aplicaciones concretas tales como realizar una facturación o llevar un fichero de historiales clínicos o, simplemente, jugar a las damas.

Para estos casos basta con proveerse de la biblioteca de programas adecuada y seguir las instrucciones específicas de cada caso.

Si por el contrario usted tiene interés en crear sus propios programas deberá conocer algún lenguaje de programación.

No cabe duda que, según la envergadura del programa que se pretenda desarrollar, algunas veces será imprescindible la colaboración de expertos profesionales en este área, pero, llevando el tema a los términos en que se mueven las personas de nuestra época, existen un sinnúmero de aplicaciones cotidianas del ordenador que pueden ser realizadas por el propio usuario por razones de comodidad y precio.

BASIC es el lenguaje de programación más popular de cuantos existen, tanto por su facilidad de aprendizaje como por sus características.

BASIC es un lenguaje que "entienden" la práctica totalidad de los microordenadores disponibles en el mercado.

Y BASIC son los comandos con que opera el ZX81.

El ZX81 de SINCLAIR, por sus extraordinarias posibilidades, su precio asequible y la gran cantidad de programas existentes, ha sido seleccionado para desarrollar este curso.

Nosotros consideramos que tratar de aprender programación sin ordenador no pasa de una introducción al tema.

Por esta razón le recomendamos se provea de un ZX81 (si aún no lo tiene).

Este computador es chiquitito pero robusto y, salvo golpes desafortunados, no suele estropearse. Por tanto no dude en comprarlo de segunda mano si este camino lo considera mejor.

Nuevos los puede conseguir en todas las tiendas especializadas.

No obstante, si tiene dificultades para conseguirlo o desea información, diríjase a BELLTONS Informática, le proporcionaremos un ZX81 nuevo o usado.

Al atravesar "La pequeña gran puerta" habrá entrado en un mundo apasionante y estará en condiciones de escribir programas bastante sofisticados, pudiendo estudiar cualquier otra versión de BASIC que un nuevo y distinto ordenador pudiera exigirle.

En todo caso la pretensión de este libro es "hacerle ver" como "razona" un computador, y va dirigido a todas aquellas personas que deseen aprender a manejar un microordenador, sin tener conocimientos previos de programación ni de la lengua inglesa.

Todos los textos simulan un simpático diálogo entre la máquina y usted, al final del cual su cerebro se habrá estructurado de forma que piense en términos que entienda la máquina.

Cada capítulo tiene ejercicios resueltos y adecuados al nivel de conocimientos adquiridos, en correspondencia con la parte 2ª "Ejercicios de programación".

En "Ejercicios de programación" se incluye un modelo de impreso para desarrollar sus propios programas.

Cada lector puede ir a la velocidad que estime conveniente. Nosotros sólo le recomendamos que no dude en volver atrás cada vez que lo crea oportuno.

Haga los ejercicios. Todos se adaptan a la configuración mínima del ZX81, es decir, 1 K RAM.

... y siempre que se le ocurra hacer tal o cual cosa, ¡hágala! y vea qué sucede. Esto será muy importante en su proceso de aprendizaje del lenguaje de programación de alto nivel conocido por BASIC.

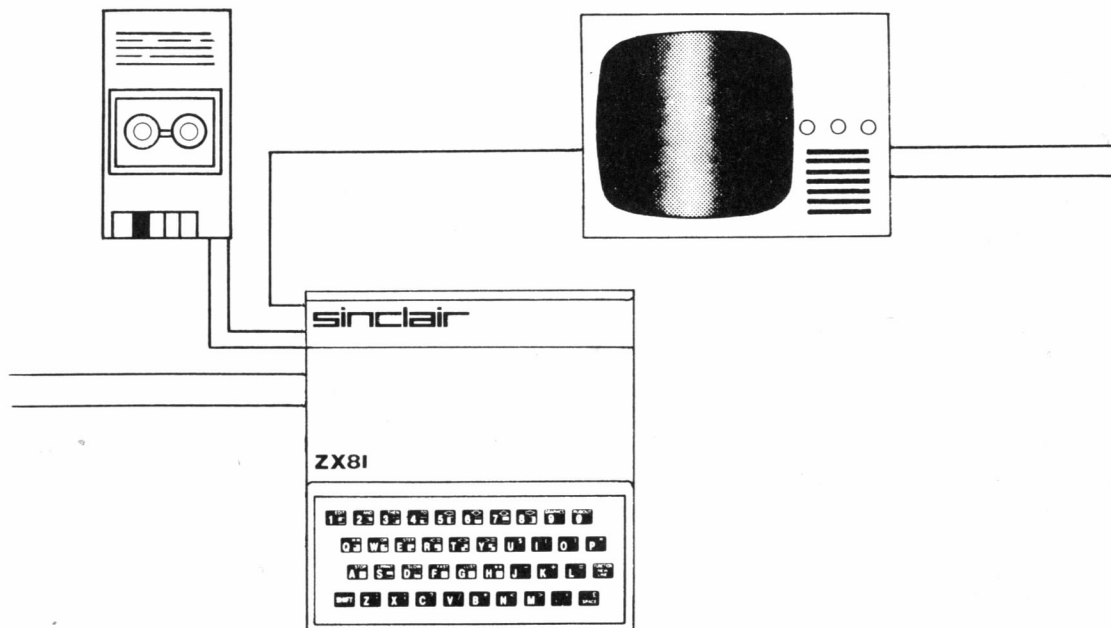
Para poder seguir este curso, el lector deberá disponer de:

- * Un ZX81 en su configuración mínima
- * Un televisor con UHF
- * Un casete, aunque no es imprescindible

En el caso de usar el casete siga las instrucciones del epígrafe: "EL CASETE... ¿PARA QUE?"

Conecte el equipo según el esquema que figura más abajo.

Para que el TV y el ZX81 estén debidamente sintonizados debe aparecer una K en el ángulo inferior izquierdo de la pantalla, para ello deberá conectar el cable de antena del equipo de serie con la toma de antena de UHF del TV y su ZX81. Hecho esto, encienda el televisor y, en los canales más altos de UHF, mueva el sintonizador lentamente. Sea paciente la primera vez, piense que ahora está acoplando sus máquinas. Si es posible deje un canal ya presintonizado.



El teclado del ZX81 parece, a simple vista, muy complicado. No lo es tanto. A través de La Pequeña Gran Puerta todos "sus misterios" quedarán desvelados sin dificultad.

En este apartado haremos una presentación general sobre cómo usarlo y, de esta forma, comenzar a "perderle el respeto".

Nada más conectar su equipo aparece una **K** en la esquina inferior izquierda; ahora, haga lo que haga, si aprieta una tecla donde figure *una letra* le aparecerá la palabra que figura encima de la propia tecla. P. ej. Apriete B. Aparecerá SCROLL.

Para borrar SCROLL apriete SHIFT y RUBOUT (ϕ).

Si aprieta SHIFT y GRAPHICS (9) la letra **K** del ángulo inferior izquierdo se transformará en una **G**. O dicho de otra forma, el **cursor** se ha cambiado de una **K** a una **G**. Si vuelve a apretar la B, la pantalla le mostrará una **B** en negativo. Borre la **B** (SHIFT y RUBOUT). Para cambiar la **G** en la **K** original, apriete de nuevo SHIFT y GRAPHICS.

Teclee SHIFT y FUNCTION (tecla NEWLINE). El cursor se transforma en **F**.

Si aprieta nuevamente B obtendrá la palabra que está justamente debajo de la tecla B propiamente dicha.

Si aprieta SHIFT y RUBOUT aparecerá automáticamente la **K** de nuevo.

El último símbolo que contiene la tecla B es un * en rojo y para obtenerlo bastará con que apriete SHIFT y B simultáneamente.

Estas líneas le dan una idea general del funcionamiento del teclado pero, ni debe pretender que con esto conoce el teclado ni desesperar ante su aparente complejidad.

Le volvemos a recomendar paciencia.

Todo se andará.

Si tiene alguna ampliación de memoria para su ZX81 no la conecte o desconecte con la máquina enchufada a la red.

Nuestro mejor consejo es que sea paciente y siga el curso en forma progresiva atendiendo a las instrucciones que se dan.

No obstante es probable que su curiosidad le lleve a manejar el computador antes de avanzar en el estudio de este libro. También es posible que cometa errores (apretando teclas incorrectas, p. ej.) en los ejercicios propuestos en los diferentes apartados.

Por estas u otras razones se encontrará, de vez en cuando, en callejones sin salida.

Si el ló en que está es muy grande, desenchufe el jack de entrada de corriente. Esta medida déjela como último recurso pues "perderá" todo lo que haya hecho en el computador.

Si el problema es mediano, puede apretar simultáneamente las teclas SHIFT y 5 u 8 para desplazar el cursor de la pantalla a izquierda o derecha respectivamente y más tarde, "borrar" apretando SHIFT y ϕ .

Los errores cometidos aparecen en la esquina inferior izquierda de la pantalla de T.V. en forma de mensajes en clave. Para interpretar estas claves diríjase al CODIGO DE ERRORES.

PRIMERA PARTE

Manejando el ZX8I

Yo soy tu microcomputador. Me llamo ZX81.

Estoy en condiciones de hacer todo lo que me ordenes, pero primero debes aprender a manejarme. Para ello has de saber:

- 1º Que mi cerebro es muy tonto comparado con el tuyo, pero suelo ser más rápido.
- 2º Que las palabras que yo entiendo son inglesas.

Lo segundo no debe ser motivo de preocupación para tí porque yo te explicaré en tu idioma todo lo necesario.

Con respecto a lo primero te diré que todo lo que yo puedo hacer es:

- A) Guardar información en memoria.
- B) Decidir entre dos posibilidades.
- C) Repetir una operación hasta que (o mientras que) suceda algo.
- D) Hacer sumas, restas y cosas por el estilo.

Mis hermanos mayores, los grandes computadores, hacen prácticamente lo mismo, pero más rápido y con más capacidad de memoria.

La capacidad de memoria se mide en K.

Yo de fábrica salgo con una K, pero me puedes ampliar cuando quieras a 16 K ó 64 K.

También me dejo conectar una impresora y muchas más cosas. No soy tan pequeñito como parezco.

Así los interpreto yo:

Símbolo	Significado
=	Me ordena dejar "algo" IGUAL a "algo"
+	Me ordena SUMAR.
-	Me ordena RESTAR.
*	Me ordena MULTIPLICAR.
/	Me ordena DIVIDIR.
**	Me ordena ELEVAR un número a una potencia. Por ejemplo: 2^3 2^{**3}
CUALQUIER NUMERO	Representa ese número.
CUALQUIER NUMERO ENTRE COMILLAS O CUALQUIER COMBINACION DE CARACTERES ENTRE COMILLAS	Representa UNA CADENA DE CARACTERES. Por ejemplo: "7527" yo no lo puedo multiplicar por 2 porque para mi "7527" es un texto y lo trataría igual que si me dijeras "DON QUIJOTE".
CUALQUIER LETRA O GRUPO DE LETRAS Y NUMEROS QUE EMPIECEN POR UNA LETRA	Representa UNA VARIABLE NUMERICA a la cual daré el valor numérico que me indiques. Por ejemplo: $ZX81 = 14990$
UNA SOLA LETRA SEGUIDA DE \$	Representa UNA VARIABLE DE CARACTERES a la cual daré el significado que me indiques, siempre que lo escribas entre comillas. Por ejemplo: $A\$ = \text{"APRENDIENDO BASIC ZX81"}$

No tienes por que aprenderte esto de memoria. Esta hoja es simplemente un apunte que usarás siempre que dudes.

Todo lo sabrás usar poco a poco.

Como sabes yo te doy los mensajes de error mediante dos números que aparecen en el margen inferior izquierdo de la pantalla: con el primero te indico el *tipo de error* con el segundo *la línea de programación donde se ha producido*.

Consulta este epígrafe cuando lo necesites.

Algunos de estos mensajes servirán exclusivamente para decirte en qué situación estoy, otros provendrán de errores de programación propiamente y, en algunos casos, te resultará más difícil averiguar la causa que lo produce.

TIPO DE ERROR	COMENTARIOS	EJEMPLO
ϕ	Todo ha salido bien Si pones un GOTO que me manda a una línea de programación más alta que la última	10 LET A = 1 20 PRINT A 30 GOTO 40
1	He leído un NEXT pero no he encontrado su FOR correspondiente. Sin embargo, he localizado una variable con la misma denominación.	10 LET A = 1 20 PRINT A 30 NEXT A
2	No hay valor para una variable. Si es una variable simple ... Falta el LET correspondiente a B. Si pertenece a un array ... Falta el DIM correspondiente a A. Si pertenece a un bucle FOR/NEXT ... Falta el FOR correspondiente a NEXT X	10 LET A = 1 20 PRINT B 10 PRINT A (1) 20 LET A = 1 20 PRINT A 30 NEXT X
3	No existe la variable que me pide dentro de un array definido en un DIM.	10 DIM A (2) 20 PRINT A (3)
4	Mi memoria está llena. No puedo obbedecer... er... te. Lea COMENTARIOS POSTERIORES	
5	Pantalla llena. CLS borra la pantalla. No las variables.	
6	He hecho un cálculo matemático que supera mis posibilidades	10 LET A = = 10**10**10 20 PRINT A

TIPO DE
ERROR

COMENTARIOS

EJEMPLO

- | | | |
|---|--|---------------------------------------|
| 7 | He leído un RETURN pero no he encontrado el GOSUB correspondiente. | |
| 8 | Sólo puedes usar INPUT dentro de un programa, es decir, con su correspondiente línea de programa. | INPUT A |
| 9 | He leído una instrucción STOP y por eso me detengo
Si aprietas CONT seguiré ejecutando el programa | 10 LET A = 1
20 STOP
30 PRINT A |
| A | El programa quiere que haga cosas que yo no puedo hacer, en términos matemáticos, con SQR, LN, ARCSIN y ARCOS. | 10 LET A = -1
20 PRINT LN A |
| B | El programa quiere que haga cosas que yo no puedo hacer en términos físicos. | 10 PRINT AT
20, 35; "A" |
| C | La cadena de caracteres que sigue a la sentencia VAL no representa una función numérica válida | 10 LET A = VAL
"Y.3" + 10 |
| D | Alguien ha apretado la tecla BREAK
También si aprietas STOP/NEWLINE cuando yo espero una entrada INPUT. | |
| F | No puedo "guardar" en tu casete un programa cuyo nombre sea la cadena de caracteres vacía" " | SAVE" " |

Te voy a explicar el significado que tienen para mi las palabras inglesas del lenguaje de programacion BASIC que vamos a utilizar inicialmente.

La aplicación exacta de estas SENTENCIAS, que así se llaman las órdenes o comandos que me das, la tendrás muy pronto.

De momento basta con que cojas la idea.

Sentencia	Significado
LOAD	Me ordena CARGAR en mi memoria el programa contenido en un casete.
SAVE	Me ordena GUARDAR el contenido del programa que hay en mi memoria a un casete.
NEWLINE	Es la palabra que necesito para obedecer cualquier orden. A un soldado le dirías: ¡Fiiirmes. YA! A mí, por ejemplo: LOAD, NEWLINE
RUN	Me ordena CORRER un programa, es decir, me indica que lo haga funcionar de manera que tu lo puedas utilizar.

Vamos a usarlas.

¿Jugamos una partida de ajedrez? ¿Sí?. Bueno pues haz el programa adecuado.

¿No sabes? ¿Es muy complicado?

En ese caso compra un casete con el programa que te interese y cárgalo (LOAD) en mi memoria.

Todos los ordenadores que existen en el mercado tienen colecciones de programas preparados para cubrir diferentes necesidades. Los soportes suelen ser cintas o discos magnéticos.

Este camino te permite disfrutar inmediatamente de tu ordenador sea cual sea el tamaño del mismo y tu nivel de aprendizaje.

Yo sólo leo programas grabados en cintas casetes.

Te explicaré el procedimiento que debes seguir:

- 1º Enchufa a la red tu magnetofón a casetes.
- 2º En el equipo de serie que me acompaña hay un cable con dos jacks (uno negro y otro gris). Conecta el gris, por ejemplo, a la entrada EAR (auricular) del magnetofón y a mi entrada EAR. No te equivoques: gris con gris o negro con negro.
- 3º Pon el volumen del magnetofón a tres cuartos de volumen total.

Ya tienes el equipo preparado para cargar (LOAD) en mi memoria cualquier programa grabado en casete.

El programa me dice qué debo hacer y cuando lo debo hacer, por esta razón, para ponerme a trabajar necesito programas que bien haces tú, bien los encargas a otras personas, o bien los compras pregrabados.

Todos los casetes te indican, entre comillas, los "nombres" de los programas que contienen.

Supongamos que me cargas (LOAD) un juego titulado "INVASORES".

Este sería el proceso:

- 1º Comprobar que el casete está rebobinado.
- 2º Aprieta la tecla LOAD
Escribe "INVASORES". No olvides las comillas.

3º Pon en marcha el magnetofón para reproducir (Como si fueras a oír a Julio Iglesias).

4º Aprieta la tecla NEWLINE.

Inmediatamente aparecerán unas rayas inclinadas en la pantalla de tu televisor y, posteriormente, unas bandas negras y blancas del mismo ancho aproximadamente.

Al final de la carga (LOAD), y en el margen inferior izquierdo aparecerá ϕ/ϕ para indicarte que la carga (LOAD) ha sido un éxito y tengo el programa en memoria.

Para que puedas jugar a los invasores me tendrás que dar la orden de correr (RUN) el programa con lo cual me prepararé para hacerlo funcionar en el momento que aprietes la tecla NEWLINE.

Resumiendo:

1º Aprieta la tecla RUN.

2º Aprieta la tecla NEWLINE.

En todo caso lee las instrucciones del casete, pues puede que no hiciera falta este último proceso.

Es posible que yo no haya podido cargar (LOAD) en mi memoria el programa y, después de las bandas blancas y negras vuelvan a aparecer en la tele las líneas inclinadas o una K en el margen inferior izquierdo.

En este caso tendrás que repetir el proceso de carga (LOAD).

Para evitar problemas de carga (LOAD) te haré algunas recomendaciones:

1º Utiliza casetes de buena calidad.

2º No conectes simultáneamente los cables a MIC y EAR. SOLO EAR CON EAR.

3º Gradúa el volumen de tu magnetofón hasta que yo "oiga" el programa con claridad.

Haz la prueba de variar el volumen durante la carga de cualquier programa y observa la pantalla. Sólo con un determinado volumen puedo cargar (LOAD) el programa.

4º Limpia la cabeza de grabación de tu magnetofón de vez en cuando. ¿Eh?

Vamos a invertir los términos.

(Este epígrafe no te será útil hasta más adelante, por tanto puedes dejarlo de momento).

Ahora eres tú quien ha desarrollado un programa para calcular, por ejemplo, tus impuestos y lo quieres guardar (SAVE) en un casete con el nombre de "IMPUESTOS". O bien quieres copiar en otro casete un programa que ha hecho un amigo tuyo.

Primero conecta tu equipo en la forma explicada más atrás pero con la diferencia de conectar un jack (el negro por ejemplo) a la entrada de micrófono de tu magnetofón y el otro extremo (jack negro también) a mi toma MIC.

Después sigue estos pasos:

- 1º Cuida de no guardar un programa encima de otro, pues perderías el anterior.
- 2º Aprietas SAVE
Escribe "IMPUESTOS". No olvides las comillas.
- 3º Pon en marcha tu magnetofón PARA GRABAR.
- 4º Aprieta NEWLINE.

La pantalla se pondrá gris durante unos instantes y después unas líneas blancas y oscuras aparecerán.

Al final, y en la esquina inferior izquierda, te daré el mensaje ϕ/ϕ . La grabación ha terminado.

Para el magnetofón después de unos tres segundos de que el mensaje ϕ/ϕ haya aparecido.

Si las cosas no suceden así repite el proceso variando el volumen.

Para comprobar que la grabación fue efectuada desconecta el jack de entrada de corriente o aprieta NEW, NEWLINE con lo cual "pierdo" toda la información que tengo en memoria.

Ahora sigue el proceso de carga (LOAD) explicado en las páginas 22 y 23.

Cuando tengas mi nuevo mensaje ϕ/ϕ , lista el programa (LIST, NEWLINE).

Ya lo tienes disponible para cuando lo necesites.

Por eso el casete es tan útil.

Pronto lo vas a utilizar.

Cuando te haga falta relee estas hojas y úsalas de guía hasta que estés seguro de los procesos.

Programando en BASIC con el ZX81

El lenguaje de programación BASIC fue desarrollado con la intención de permitir una fácil comunicación entre los ordenadores y los humanos... que hablan inglés. Por esta razón debes estudiar estas lecciones sin más ánimo que coger una idea del significado de las palabras que vas a usar en lo sucesivo.

Sentencia**Significado**

PRINT

Me ordena IMPRIMIR en la pantalla de TV.

LET

Me ordena DEJAR algo (número o letras) igual a algo (números o letras).

INPUT

Me ordena ESPERAR UNA ENTRADA de datos.

LIST

Me ordena LISTAR en la pantalla el programa que tengo en mi memoria.

SQR

Me ordena hallar la RAIZ CUADRADA de un número.

REM

Me ordena RECORDAR algo pero sin que influya en la programación.

Si te parece bien podríamos hacer juntos tu primer programa.

¿Qué es pronto? ¿Qué todavía no sabes? Tonterías.

Fijate en este ejemplo:

Vamos a hacer un programa para sumar dos números cualquiera, los cuales tu me dirás a través de mi teclado.

Ahora te diré como debes "hablarme".

"Espera (INPUT) a que te dé el primer número (A), después espera INPUT a que te dé el otro número (B). A continuación deja (LET) en tu memoria el resultado (C) de sumar los dos números anteriores ($A + B$). Finalmente, imprime (PRINT) en la pantalla al valor (C) de la suma".

Esto escrito en mi teclado, y de acuerdo con lo estudiado en las páginas, 18 y 27 sería una cosa así:

(No teclees nada en mi teclado hasta que yo te avise).

INPUT A

INPUT B

LET C = A + B

PRINT C

Si tratas de introducir estas líneas en mi ordenado cerebro no conseguirás nada.

Necesito que me digas claramente el orden en que tengo que actuar.

Esto sería un programa:

(No teclees nada todavía. No seas impaciente).;

Programa

Comentarios

1 INPUT A

Primera LINEA DE PROGRAMA.

2 INPUT B

Segunda LINEA DE PROGRAMA.

3 LET C = A + B

Tercera LINEA DE PROGRAMA.

4 PRINT C

Cuarta LINEA DE PROGRAMA.

Estas líneas de programa deben ir de menor a mayor justo en el orden en que tú quieras que yo opere. Cualquier numeración que siga la norma anterior es buena, pero yo te recomiendo que vayas de 10 en 10. Así:

Programa**Comentarios**

10 INPUT A

Numerando las LINEAS DE PROGRAMA de 10 en 10, tienes la posibilidad de introducir nuevas LINEAS con nuevas INSTRUCCIONES para mí.

20 INPUT B

30 LET C = A + B

40 PRINT C

Tal vez te preguntes qué significan la A, la B o la C. Relee en la página 18 lo referente a variables numéricas.

Dado que A y B pueden ser cualquier número que tú me des por el teclado y, puesto que su suma también puede ser cualquier número, yo necesito asimilarlas a unas VARIABLES NUMERICAS y esas VARIABLES NUMERICAS son las que tú me indiques. En este caso A, B y C.

Ahora enchúfame a la corriente y al televisor. Si dudas échale un vistazo al esquema de la página 11.

Introduce el programa en mi memoria a través de mi teclado. Sigue el proceso que te indico y analiza los comentarios.

EMPIEZA A TECLEAR:

Proceso**Comentarios**

Antes de apretar ninguna tecla aparece una **K** en la esquina inferior izquierda de la pantalla. El **cursor** está representado por esa **K**.

1º escribe 10

Este número lo interpreto como el número de la LINEA DE PROGRAMACION por ser previo a ninguna SENTENCIA (ver TECLADO en pág. 12).

2º aprieta INPUT

INPUT está encima de la tecla I y es una SENTENCIA U ORDEN.
Observa como la **K** se ha transformado en **L**. Esto implica que ya no espero ninguna otra sentencia.

Proceso

Comentarios

3º aprieta A

Por estar el **cursor** representado por una **L**, yo interpretaré que tu quieres justamente la letra A y no ninguno de los otros símbolos que la acompañan en su tecla.

4º aprieta NEWLINE

Con esta orden yo sé que la línea de programa 10 está completa y que debo guardarla en mi memoria.

En este momento aparece la **K** nuevamente.

5º escribe 20

6º aprieta INPUT

7º aprieta B

8º aprieta NEWLINE

La línea de programa 20 pasa a mi memoria. El **cursor** se transforma en **K** nuevamente y aparece en el margen inferior izquierdo.

9º escribe 30

10º aprieta LET

11º aprieta C

12º aprieta =

Para obtener el símbolo = debes apretar simultáneamente SHIFT y L. De hecho todas las instrucciones en rojo exigen apretar a la vez SHIFT.

13º aprieta A

14º aprieta +

El símbolo + está en la tecla K y en rojo, por lo tanto deberás apretar simultáneamente SHIFT y K.

15º aprieta B

16º aprieta NEWLINE

La línea de programa 30 pasa a mi memoria.

17º escribe 40

18º aprieta PRINT

19º aprieta C

20º aprieta NEWLINE

La línea de programa 40 pasa a mi memoria.

Ya sé lo que quieres que haga y cómo quieres que lo haga.

Vamos a usarlo. Para ello sólo tienes que:

1º Apretar RUN CORRELO

2º Apretar NEWLINE ¡YA!

Inmediatamente verás aparecer en la esquina inferior izquierda de tu pantalla una **L**, lo cual te indica que espero el primer número.

Escríbelo y aprieta NEWLINE. (El 15 por ejemplo.)

Otra vez aparece la **L** a la espera del segundo número.

Escríbelo y aprieta NEWLINE. (El 5 por ejemplo.)

En la esquina superior izquierda aparece mi respuesta y en la inferior izquierda $\phi/40$.

Enhorabuena, eres un programador . . . bueno, ya te falta menos.

Prueba con otros números.

Supongamos ahora que ya no quieres sumar. Ahora quieres multiplicar cualquier número por cualquier otro.

Tienes dos opciones: o haces un programa nuevo o modificas éste.

Vamos a modificar éste.

Pero antes haz los **EJERCICIOS DE CALCULO**.

... nos quedamos en que la esquina superior izquierda tenías mi contestación.

En la esquina inferior izquierda aparece $\phi/40$. Es un mensaje mío para ti:

“NO HAY ERROR/ESTOY PARADO EN LA LINEA 40 A TUS ORDENES”.

Bien, como ahora se trata de modificar el programa que tengo en memoria, lo primero que has de decirme es “HAZ UN LISTADO”. Para ello:

1º LIST 10 LISTA en la pantalla el programa a partir de la línea 10.

2º NEWLINE ¡YA!

Ya lo tienes.

Observa esa pequeña flechita en la línea 10. Ese es mi **puntero**.

¿Qué queríamos hacer? ¡Ah! Multiplicar un número por otro. ¿Cuál sería el programa si lo hicieramos nuevo?

Este:

10 INPUT A

20 INPUT B

30 LET C = A * B

40 PRINT C

Que viene a decirme: “Espera (INPUT) a que dé el primer número (A), después espera (INPUT) a que te dé el otro número (B). A continuación deja (LET) en tu memoria el resultado (C) de multiplicar los dos números anteriores (A * B). Finalmente imprime (PRINT) en la pantalla el valor (C) de la multiplicación”.

Obviamente es *casi* igual a nuestro programa de sumas, sólo varía la línea 30.

Así, a ti te interesa que yo opere sobre la línea 30... y ahora estoy en la 10, según indica el **puntero**. Desplaza el **puntero** hacia abajo apretando simultáneamente SHIFT y 6.

Cuando hayas situado el **puntero** en la línea de programa 30, dame la orden de editar (EDIT) la línea sobre la cual está el **puntero** de esta forma: Aprieta simultáneamente SHIFT y 1.

Ya tienes la línea de programa 30 en la parte inferior de la pantalla que es el único lugar donde puedes introducir modificaciones.

Ahora puedes manipular la línea 30.

En este caso hay que variar todo lo que está a la derecha del igual.

Verás en la línea editada una **K** entre el 30 y LET.

Observa que sucede al apretar SHIFT/8.

La **K** se transforma en **L** y avanza hacia la derecha. Repite la operación hasta situar la **L** a la derecha de B.

ATENCIÓN: La **K** te indica que sólo acepto SENTENCIAS, tales como . . . LET, RUN, etc. La **L** te advierte que lo próximo que introduzcas lo aceptaré como un carácter sencillo: Letra o número, ícomo si fuera una máquina de escribir!

Dicho esto a borrar (RUBOUT) lo que sobra: aprieta SHIFT, ϕ .

Cada vez que hagas esto borras un caracter, un símbolo o una sentencia.

Cuando hayas borrado lo que sobra . . .

. . . escribe lo correcto:

A * B

(* se consigue mediante SHIFT, B).

Recuerda: para que yo acepte esta modificación tendrás que apretar NEWLINE.

¿Corremos el programa? (¿lo hacemos funcionar?).

¿Sí? Pues . . .

Proceso

Comentarios

1º RUN

CORRE EL PROGRAMA.

2º NEWLINE

¡YA!

¿Una **L** en la esquina inferior izquierda? pues ya sabes . . . necesito el primer número . . . ¿otra **L**?, pues ¡ale!, otro número (no te olvides de NEWLINE).

Ya tienes mi respuesta.

¿Sabes ya cómo corregir y modificar? ¿Sí?

En ese caso haz los **EJERCICIOS DE MODIFICACION**.

Afortunadamente hemos numerado las líneas de programa de 10 en 10. Gracias a esto vamos a intercalar nuevas líneas de programa sin necesidad de alterar lo ya escrito.

Si aún está el programa anterior en mi memoria dame la orden de listarlo (LIST, NEWLINE) en pantalla.

En otro caso introduce el programa de nuevo copiándolo de la página 32.

Dame estas dos líneas de programa tecleando:

Proceso**Comentarios**

5 PRINT "UN NUMERO" (NEWLINE por supuesto)

15 PRINT "OTRO NUMERO" (NEWLINE por supuesto).

(Automáticamente yo colocaré la línea 5 antes de la 10, y la 15 entre la 10 y la 20).

En estas condiciones el programa queda:

Programa**Comentarios**

5 PRINT "UN NUMERO" Imprimo "UN NUMERO".

10 INPUT A Espero la entrada del número A.

15 PRINT "OTRO NUMERO" Imprimo "OTRO NUMERO".

20 INPUT B Espero la entrada del número B.

30 LET C = A * B Dejo C igual a la multiplicación de A por B.

40 PRINT C Imprimo C.

Nota: Los espacios en blanco para los textos entre comillas los consigues apretando SPACE.

Corre el programa (RUN, NEWLINE) y hazlo funcionar con un par de números, observa que las impresiones las hago en el mismo orden que están en el programa. De arriba hacia abajo.

Ahora el programa es más claro.

Para "clarificarlo" más añade estas líneas:

(Primero lístalo: LIST, NEWLINE)

Programa	Comentarios
12 PRINT A	(No olvides NEWLINE)
22 PRINT B	(No olvides NEWLINE)
28 PRINT "RESULTADO"	(No olvides NEWLINE)

Corre el programa (RUN, NEWLINE) y hazlo funcionar con otro par de números. Mis respuestas van a renglón seguido.

¿Las ponemos a continuación?

Bastará con que añadas a las líneas 5, 15 y 28 un punto y coma (;) al final de las mismas.

¿Te acuerdas como se hacía? Te ayudaré con la modificación de la línea 15, p. ej., y tú sigues después.

1º LIST 15 (así el puntero se sitúa sobre la línea 15, ver la NOTA al pie)

2º EDIT (SHIFT/1)

Nota: Cuando aprietas LIST, seguido de un número (y después NEWLINE por supuesto) se lista el programa desde la línea de programa cuyo número es el que sigue a la instrucción LIST, p. ej.:

LIST 15, NEWLINE te dará un listado por pantalla a partir de la línea 15.

3° Correr la **K** (**L**) hasta el final.

4° Teclear; (SHIFT/X)

5° NEWLINE

Cuando hayas modificado estas líneas, vuelve a correr el programa (RUN y NEWLINE) y hazlo funcionar con otros dos números.

Ahora te doy mis respuestas justo a continuación de los textos porque así me lo indican los puntos y comas que hay al final de los mismos, en las correspondientes líneas de programa.

Si sustituyeras los puntos y comas del final de estas líneas por comas (,) te daría mis respuestas a partir de la columna 16 hacia la derecha.

Te esperan los **EJERCICIOS DE MODIFICACION** en la página 88.

A lo peor te doy la impresión de que sólo puedo manejar números.

¡Pues no es así!

También trabajo con letras.

Verás. Yo sé que tu quieres darme datos numéricos, como ya vimos en la página 18, si utilizas variables numéricas, es decir, me das cualquier combinación de letras y números con tal de que el primero sea una letra. P. ej.:

```
LET ZX81 = 14990
ó
INPUT ZX81
```

estas son variables numéricas porque guardan números o esperan entradas de números.

Pero . . . ¿y si quieres guardar textos o esperar entradas de textos? Sencillo. Cualquier letra (pero una sola) seguida de \$ (SHIFT/U) representan variables de caracteres y me indica que son textos lo que voy a manejar. P. ej.:

```
LET M$ = "ZX81" (ver nota al pie)
INPUT M$ — (espera la entrada de un texto)
             "o cualquier combinación de
             caracteres"
```

Vamos a manejarlas.

Las variables numéricas las puedo sumar, restar, multiplicar, dividir, etc. Por ejemplo:

```
LET A = B + C ó LET M = T * C + F ó LET JUAN = SQR H
```

Las variables de caracteres SOLO LAS PUEDES SUMAR. Por ejemplo:
LET E\$ = K\$ + V\$.

NOTA: Las variables de caracteres las puedes hacer igual a cualquier cadena de caracteres (entre comillas). Por ejemplo:

```
LET A$ = "5 MILLONES"
```

Teclea este programa:

Programa	Comentarios
	No te olvides NEWLINE.
10 PRINT "DIME TU NOMBRE"	Imprimo "DIME TU NOMBRE"
20 INPUT N\$	"Espero la entrada de un texto".
30 LET A\$ = "BUENOS DIAS"	Dejo A\$ igual a "BUENOS DIAS".
40 LET B\$ = A\$ + N\$	Dejo B\$ igual a A\$ + N\$.
50 PRINT B\$	Imprimo B\$.

Córrelo (RUN, NEWLINE). Teclea JUAN y NEWLINE.

Para mí sumar cadenas de caracteres no es más que poner una cadena a continuación de otra. Por esta razón la contestación que yo te daré será BUENOS DIASJUAN. Introduce ahora un espacio (SPACE) entre la última S y las comillas de la línea 30. Corre (RUN, NEWLINE) el programa de nuevo y la impresión será: BUENOS DIAS JUAN.

Haz los **EJERCICIOS CON TEXTOS Y VARIABLES DE CARACTERES.**

Yo considero que la pantalla de tu televisor tiene 24 líneas y 32 columnas, por tanto yo puedo poner la información sobre cualquiera de las 24 líneas y 32 columnas. En ambos casos comienzo a contar desde 0 y llego a 23 ó 31, según sean líneas o columnas.

Las 24 líneas para mí van numeradas de arriba a abajo y las 32 columnas van de izquierda a derecha.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0																																
1																																
2																																
3																																
4																																
5																																
6																																
7																																
8																																
9																																
10																																
11																																
12																																
13																																
14																																
15																																
16																																
17																																
18																																
19																																
20																																
21																																

Para imprimir cualquier caracter en el cuadrado de la línea 5, columna 7, tienes varias opciones:

1º PRINT AT 5,7;"A".

- 2º Después de cuatro impresiones en líneas de pantalla consecutivas a lo largo de un programa (y no borradas de la pantalla) y a continuación

PRINT TAB 7; "A"

- 3º Después de cuatro instrucciones PRINT en blanco:

10 PRINT

20 PRINT

30 PRINT

40 PRINT

50 PRINT TAB 7; "A"

- 4º Con una cadena de caracteres en blanco:

PRINT " 6 veces SPACE A", en las mismas condiciones de los dos ejemplos anteriores.

Para borrar la pantalla sin que afecte a las instrucciones del programa ni al contenido de las variables: **CLS**.

Ten a mano este esquema de la pantalla durante el estudio de este epígrafe y sus ejercicios.

¡Ah!, las dos líneas finales, la 22 y 23, me las reservo yo para comunicarme contigo y no están disponibles para tí.

La forma en que has de ordenarme donde debo imprimir es muy sencilla. Debes decirme:

PRINT AT

o lo que es igual: **imprime en ...**

Nota: Para conseguir las instrucciones que están situadas debajo de las teclas (como es el caso de AT) se debe proceder como sigue:

- 1º SHIFT y NEWLINE simultáneamente.

- 2º La tecla en cuestión.

De esta forma el cursor se transforma en **F**.

Haz esta pequeña prueba:

Programa**Comentarios**

```
10 PRINT AT
```

```
  φ, 31; "***"
```

(No te olvides de NEWLINE)

Corre el programa. Tu ya sabes que para correr un programa debes apretar RUN, NEWLINE y si no lo recuerdas . . . vete a la página 21.

Haz esta otra prueba:

Programa**Comentarios**

```
10 PRINT AT
```

```
  φ, 31; "***"
```

Imprimo * en la línea φ, columna 31.

```
20 PRINT AT
```

```
  21, 31; "***"
```

Imprimo * en la línea 21, columna 31.

```
30 PRINT AT
```

```
  10, 29; "*****"
```

Imprimo *** a partir de la línea 10, columna 29.

Corre este programa y analízalo.

Cambia la línea 30 por esta otra:

```
30 PRINT AT 10, 30; "*****"
```

Corre el programa de nuevo y observarás que el último * sale en la siguiente línea, esto se debe a que yo considero que al no haber columna disponible a la derecha continuo en la primera columna de la línea siguiente:

Recuerda:

; me obliga a escribir a continuación

, me obliga a escribir a partir de la columna 16

Muchas veces es más cómodo usar PRINT TAB.

Teclea este programa:

Programa

Comentarios

10 PRINT "***"

Borra la pantalla si contiene algo. Imprimo en la primera línea disponible en la pantalla y desde la columna ϕ . En este caso línea ϕ , columna ϕ .

20 LET A\$ = "++"

30 LET B\$ = "--"

Dejo las variables de caracteres A\$ y B\$ igual a ++ y --, respectivamente.

40 PRINT "A\$"

Imprimo en la primera línea disponible en la pantalla y desde la columna ϕ . En este caso línea 1, columna ϕ , puesto que la línea ϕ la ocupará **.

50 PRINT TAB 5; "TAB"

Imprimo en la primera línea disponible en la pantalla y desde la columna 5. En este caso línea 2, columna 5, puesto que la línea ϕ la ocupará ** y la uno ++.

60 PRINT AT 10,
12; A\$ + B\$

Imprimo en la fila 10 y a partir de la columna 12.

Corre este programa (RUN, NEWLINE) y compara lo que aparece en la pantalla con los comentarios de cada línea.

Para borrar este programa, y en general el contenido de mi memoria tienes dos opciones:

PRIMERA DESENUFAR EL JACK DE ALIMENTACION

SEGUNDA APRETAR NEW, NEWLINE

Con las instrucciones de las páginas anteriores puedes colocar números y textos en el lugar que desees de la pantalla. Arriba, abajo, a la derecha y a la izquierda, pero ¿tú sabes que...?

Yo puedo controlar la pantalla de forma que borre o imprima cosas a tu gusto.

Antes de seguir, unas SENTENCIAS muy interesantes y espectaculares:

CLS

Me ordena limpiar la pantalla y sólo la pantalla. El contenido de mi memoria no se altera.

PAUSE

Me ordena mantener en pantalla todo lo que esté impreso en la misma durante el tiempo que tú me digas, después seguiré ejecutando el programa.

Ejemplo:

PAUSE 50 — equivale a una pausa de 1 segundo.

PAUSE 500 — equivale a una pausa de 10 segundos.

POKE 16437, 255

Después de una sentencia PAUSE debes utilizar esta cosa extraña —si trabajas en modo FAST—. Es como un tranquilizante para mí. Te advierto: ¡Si no sigues este consejo puedes perder tu programa!

Nota: Una sentencia PRINT sin nada a continuación, lo interpreto como una línea en blanco.

Prueba esto:

10 PRINT "ESTOY AQUI"

20 PRINT

30 PRINT

40 PRINT "AHORA ESTOY TRES LINEAS
MAS ABAJO"

Córrelo y saca tus conclusiones.

A continuación tienes un listado de un programa que está al límite de mi memoria (1 K), por esta razón haré cosas que no son usuales.

No te preocupes y actúa de la misma forma en que lo has hecho hasta ahora, pero fíjate en ellas y así sabrás qué sucede en estas circunstancias.

ECHALE UN VISTAZO A "ALGUNOS CONSEJOS POSTERIORES".

En la página 35 te decía que la sentencia LIST seguida de un número te da un listado del programa en memoria a partir del citado número.

Cuando un programa tiene más líneas de programa de las que yo puedo mostrar en pantalla, o el contenido de las líneas es grande, o estoy al límite de mi capacidad operativa, como es el caso que te he explicado más arriba, entonces no podré darte listados completos, sólo parciales.

Para saber cómo actuar en estos casos, y cuando hayas introducido el programa que sigue, puedes probar un LIST1.

En el margen inferior izquierdo te doy un mensaje 4/φ con lo que te indico que no tengo más memoria. El listado que te ofrezco llega hasta la línea 90. Aprieta ahora LIST 90.

Te doy un listado entre 90 y 160, con el mismo mensaje 4/ ϕ .

Un LIST 160 te permitirá obtener la última línea de programación completa.

Espero que no te olvidarás apretar NEWLINE tras cada orden completa. Considerando que ya lo sabes no volveré a repetírtelo.

Estudia y escribe este programa:

Programa		Comentarios
10 PRINT AT ϕ , ϕ ; "***	***"	Deja 28 espacios (SPACE) en blanco
20 PRINT AT 21, ϕ ; "***	***"	
30 PRINT AT 2, 11; " ----- "		Teclea 8 guiones
40 PRINT AT 2, 11; "ATENCION"		
50 PRINT AT 4, 11; " ----- "		Teclea 8 guiones
60 PRINT		
70 PRINT		
80 PRINT		
90 PRINT TAB 3; "TU ZX81 SE COMUNICA CONTIGO"		
100 PRINT AT 17, 0; "ESTAS PREPARADO?"		
110 PRINT AT 20, 8; "BORRARE LA TV. ESPERAME"		
120 PAUSE 500		
130 POKE 16437, 255		
140 CLS		
150 PRINT TAB 8, "YA ESTOY AQUI"		
160 PRINT AT 15, 0; "ACABA LA PRIMERA ETAPA. REPASALA BIEN, HAZ PRUEBAS Y MODIFICACIONES SOBRE LO ESTUDIADO".		

EJERCICIOS DE IMPRESION EN PANTALLA.

¡Sí!. Vete a (GOTO) a París para subir a la Torre Eiffel.

Vete a (GOTO) a la página 21 si quieres repasar una **LECCION DE INGLES**.

Pero si quieres darme una orden de este estilo en un programa, dime:

VETE A LA LINEA DE PROGRAMA NUMERO 23 y para ello debes teclear:

GOTO 23

Introduce este programa:

Programa

Comentarios

10 REM "UN BUCLE"

La instrucción REM me ordena que REMEMORE el texto que haya entre comillas. No afecta al programa para nada y lo puedes colocar donde quieras. Saldrá en los listados con el único fin de recordarte aquello que esté entre las comillas.

20 PRINT
"PRIMER NUMERO?"

Mientras no aparezca la sentencia GOTO mi proceso normal de trabajo será ir de una línea de programación a la siguiente.

30 INPUT A

40 PRINT TAB 25; A

50 PRINT
"SEGUNDO NUMERO?"

60 INPUT B

70 PRINT TAB 25; B

80 LET C = A - B

90 PRINT
"LA RESTA ES: "; C

100 GOTO 20

GOTO me obliga a ir a la línea de programación cuyo número figura inmediatamente a continuación de la misma. En este caso a la 20.

Si corres RUN, NEWLINE este programa verás que automáticamente después de darte el resultado de la resta, te vuelve a preguntar por el primer número.

Si quisieras listar tu programa (LIST/NEWLINE) te daría un error 2/30 que quiere decir que en la línea 30 no puedes meter una variable de letras, ya que espero una variable numérica (INPUT A). Es decir, he interpretado LIST como una cadena de caracteres y no como una SENTENCIA. ¿Sabes por qué? Porque estoy en un **bucle**.

Ahora, al darte la señal de error 2/30, he salido del bucle y ya puedes listar (LIST, NEWLINE). Vuelve a correr el programa.

Si pruebas a darme datos y yo a darte respuestas, uno detrás de otro te encontrarás que llega un momento en que te daré un error 5/50, con lo cual te indico que "por estar la pantalla llena me he visto obligado a pararme en la línea de programa 50".

Si quisieras seguir tendrás que correr (RUN, NEWLINE) el programa otra vez.

Ahora GOTO "EJERCICIOS CON GOTO".

- $A > B$ La variable numérica A es mayor que la B.
- $A < B$ La A es menor que la B.
- $A \geq B$ La A es mayor o igual que la B.
- $A \leq B$ La A es menor o igual que la B.
- $A\$ > B\$$ La variable de caracteres A\$ es mayor que la B\$
- $A\$ < B\$$ La variable de caracteres A\$ es menor que B\$
- $A\$ \leq B\$$ La variable de caracteres A\$ es menor o igual que la B\$.
- $A\$ \geq B\$$ La A\$ es mayor o igual que B\$
- $A \neq B$ La variable numérica A no es igual a B.
- LET A = DEJA la variable numérica A igual a un número, a otra variable numérica o a una operación aritmética o trigonométrica.
- Ejemplo:
- LET A = SQRB
- LET A\$ = DEJA la variable de caracteres igual a un texto.
- Ejemplo:
- LET A\$ = "DOLAR"
- PRINT A IMPRIME en la pantalla el valor de la variable numérica A.
- PRINT A\$ IMPRIME en la pantalla el valor de la variable de caracteres A\$
- INPUT A ESPERA una ENTRADA de la variable numérica A
- INPUT A\$ ESPERA una ENTRADA de la variable de caracteres A\$
- RUN CORRE el programa, es decir, haz funcionar el programa que hay en memoria.
- LIST X LISTA el programa en la pantalla a partir de la línea de programa X.

- LOAD "CARGA" CARGA el programa denominado "CARGA" (1) desde un casete a la memoria.
- SAVE "GUARDA" GUARDA el programa denominado "GUARDA" (1) desde la memoria a un casete.
- GOTO X VETE a la línea del programa número X.
- REM "RECORDANDO" ... RECUERDA "RECORDANDO" Esta sentencia no afecta al programa y sólo sirve para introducir aclaraciones.
- CLS LIMPIA LA PANTALLA. Borra el contenido de la pantalla sin alterar el contenido del programa.
- PAUSE t PAUSA de t/50 segundos. t puede valer cualquier número comprendido entre 1 y 32766.
- POKE 16437, 255 Después de una sentencia PAUSE se aplica esta otra.

Estas son mis prioridades a la hora de efectuar operaciones matemáticas.

- 1º Ejecuto ante todo lo que hay dentro de los paréntesis.
- 2º Transformo primero las potencias.
- 3º Después multiplico y divido.
- 4º Después sumo y resto.
- 5º A igualdad de prioridades opero de derecha a izquierda.

Basta de esto ¡A aplicarlo!

(1) Tanto CARGA como GUARDA son ejemplos y equivalen al nombre que tenga el programa.

Hasta el momento me has dado ordenes concretas y yo no he tenido que complicarme la vida.

En lenguaje de programación BASIC (que es el que yo manejo) hay una sentencia para tomar decisiones que es muy fácil de entender . . . para un inglés:

IF THEN

Y que exactamente quiere expresar cosas como ésta:

Si (IF) A = 5 entonces (THEN) vete a (GOTO) 110.

Que viene a decir: Si la variable numérica A es igual a 5 entonces vete a la línea de programa 110.

Esto dicho en BASIC sería:

IF A = 5 THEN GOTO 110

(De no cumplirse esta condición seguiría a la próxima línea de programación).

Veamos un ejemplo:

Si (IF) L\$ >= W\$ entonces (THEN) imprime (PRINT) "PEPE".

O lo que es igual: Si la variable de caracteres L\$ es mayor o igual que la variable de caracteres W\$ entonces imprime PEPE.

Escrito como una instrucción de programa:

IF L\$ = W\$ THEN PRINT "PEPE"

Si se cumple la condición imprimo PEPE y, en cualquier caso, sigo a la próxima línea de programación.

¿Otro ejemplo más?

Mejor prueba tú. Cómo me darías en BASIC esta instrucción:

Si (IF) la variable de caracteres V\$ no es igual (<>) a Juan entonces (THEN) deja (LET) la variable numérica X igual a la variable numérica Y multiplicada por 787.

¿Una ayudita?

Ahí va:

```
IF V$ <> "JUAN" THEN LET X = Y * 787
```

En la página anterior tal vez te haya causado sorpresa el hecho de comparar dos variables de caracteres.

Pues es posible.

Para mí las letras o conjuntos de letras tienen un "peso" y por tanto las puedo comparar.

Con este programa podrás ver mi criterio a este respecto:

Programa

Comentarios

```
10 REM "COMPARAN-  
RANDO EL ABCDARIO"
```

Este programa te compara letras entre sí o letras y textos o textos con textos.

```
20 PRINT "DIME UNA  
LETRA"
```

```
30 INPUT X$
```

```
40 PRINT "DIME OTRA  
LETRA"
```

```
50 INPUT Y$
```

```
60 IF X$ > Y$ THEN  
GOTO 90
```

```
70 PRINT Y$. "ES MA-  
YOR QUE "; X$
```

Deja un espacio entre las comillas y el texto.

```
80 GOTO 20
```

```
90 PRINT X$. "ES MA-  
YOR QUE "; Y$
```

Deja un espacio entre las comillas y el texto.

```
100 GOTO 20
```

Compara las letras del abecedario entre sí y luego palabras con palabras. Luego teclea números.

El hecho de que yo pueda comparar letras y textos te será muy útil para ciertas aplicaciones.

Si tratas de listarme no podrás, ya que estoy a la espera de letras (más bien de caracteres en general), pero en ningún caso aceptaré sentencias como LIST o NEWLINE. Para salir del bucle, GOTO a página 46 y GOTO 52.

Un problema sencillo.

“Tu deseas darme cualquier número y que yo lo multiplique por 10 si es menor o igual que 100 y lo divida por 20 en cualquier otro caso y me pides que imprima la respuesta”.

Mentalmente me dirías:

“Espera (INPUT) una variable numérica. Si (IF) el valor de esa variable es menor o igual (\leq) que 100, entonces (THEN) vete (GOTO) a una línea de programa donde dejes (LET) esa variable igual ($=$) a su valor multiplicado ($*$) 10 y después imprime (PRINT) el resultado.

En cualquier otro caso deja (LET) el valor de la variable igual ($=$) a su valor dividido ($/$) por 20 y después imprime (PRINT) el resultado”.

Razona tranquilamente y ve desarrollando tu programa. Si te “encasquillas” estudia el programa que sigue y continúa intentando aplicar tus propias iniciativas.

Programa de apoyo:

```
1  REM "UN PROBLEMA SENCILLO"
10 PRINT "NUMERO?"
20 INPUT A
30 IF A <= 100 THEN GOTO 70
40 LET A = A/20
50 PRINT A
60 GOTO 10
70 LET A = A * 10
80 PRINT A
90 GOTO 10
```

En la línea de programa número 30 yo tomo una decisión: Si el número que tu me das (INPUT A) no cumple la condición ($A \leq 100$), sigo a la línea de programa 40, si la cumple me voy a la línea de programa 70.

Corre el programa.

En la esquina inferior izquierda aparece una **L** con lo que te indico que estoy a la espera de un nuevo número, y así constantemente ya que los **GOTO** de las líneas 60 y 90 (los dos únicos caminos que puedo seguir) me llevan siempre a la línea 10.

Estoy en un **BUCLE**.

Si ahora pretendieras listarme (**LIST, NEWLINE**) yo interpretaría **LIST** como una cadena de caracteres (no como la Sentencia **LIST**), esto quiere decir que te daré un error $2/2\phi$, variable no definida en la línea de programación 20, lo cual es efectivamente cierto ya que yo esperaba la entrada de un valor numérico y no de una cadena de caracteres.

En todo caso para salir de un **BUCLE** lo mejor es provocar un error y apretar después **LIST, NEWLINE**. En este caso aprieta **STOP, NEWLINE** y después **LIST, NEWLINE**. Cuando por el contrario yo espere la entrada de caracteres (cosa que notarás por que la **L** estará entre comillas) la forma de salir del bucle es "borrar las comillas de la **L**" y después **STOP, NEWLINE**. Así:

- 1° RUBOUT (SHIFT, ϕ)
- 2° \Rightarrow (SHIFT, 8)
- 3° RUBOUT (SHIFT, ϕ)
- 4° STOP, NEWLINE (ya tienes el error D/30)
- 5° LIST, NEWLINE (ya tienes el listado).

EJERCICIOS DE DECISION.

Tan práctico como tomar decisiones te puede resultar mi capacidad de repetir lo que tú me digas y hasta que se cumpla lo que me ordenes.

El lenguaje BASIC de programación fue inventado en los Estados Unidos y, como vas comprobando, es fácil de aplicar pero tiene el pequeño inconveniente de estar pensado en inglés.

Si la sentencia que ahora vamos a estudiar se hubiera escrito en la Universidad de Salamanca, hubiera salido una cosa así:

DESDE este valor **HASTA** este otro valor

haz

todo

lo

que

te

digo

hasta

que

te

encuentres

la

sentencia

SIGUIENTE valor

Realmente la Sentencia BASIC es:

FOR TO

.

.

.

.

.

.

.

NEXT

Si no te acordaras de cual es la tabla de multiplicar del 9, podrías hacer este programa:

Programa

Comentarios

1	REM "APLICANDO FOR"	
10	FOR A = 0 TO 10	DESDE A = 0 HASTA 10 (1)
20	LET B = A * 9	DEJA B = A * 9
30	PRINT B	IMPRIME B
40	NEXT A	SIGUIENTE valor de A

Esto me viene a decir:

Desde (FOR) A = 0 hasta (TO) 10 deja (LET) B = A x 9 e imprime (PRINT) el valor de B. Vuelve a la Sentencia FOR y haz el SIGUIENTE ciclo con el SIGUIENTE valor hasta (TO) el valor 10.

Evidentemente podrías haber hecho un programa más "bonito". Cambia la línea 30 por esta otra:

30 PRINT "9 por"; A; "ES"; B

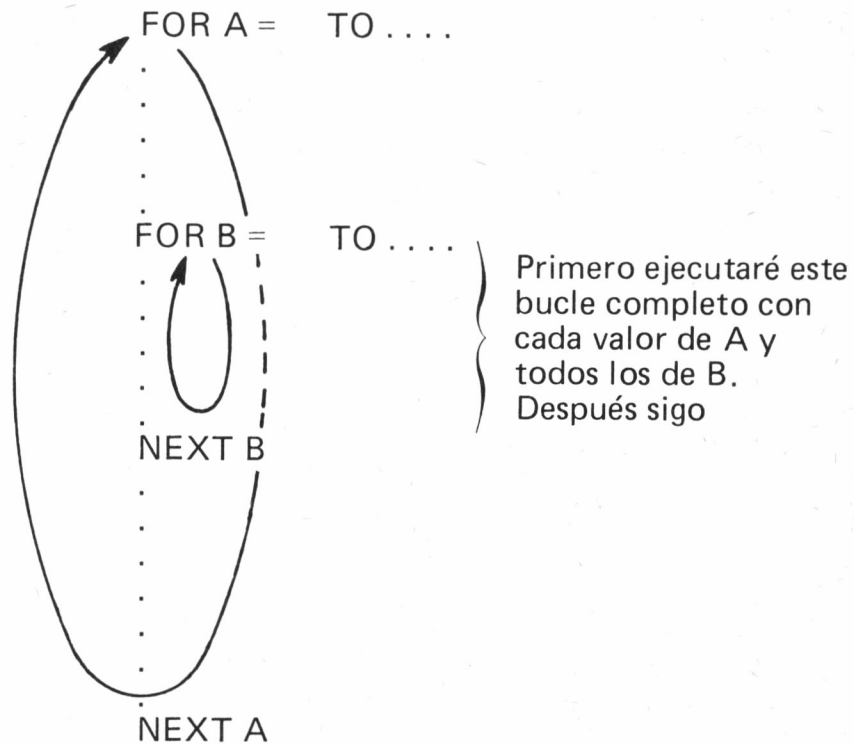
Deja un espacio en blanco (SPACE) entre las comillas y el principio y el fin de cada texto: " 9 POR ", " ES ".

(1) *Nota:* En esta Sentencia la variable (A en este caso) sólo puede ser UNA LETRA a diferencia de las anteriores que admitían más posibilidades.

Corre (RUN, NEWLINE) el programa.

Dentro de un BUCLE FOR/NEXT puedes meter otro BUCLE FOR/NEXT.

La única precaución que debes guardar es no cruzar los BUCLES. Te haré un esquema y después un ejemplo.



¿Recordamos la tabla de multiplicar por 3, por 4 y por 5?

```
1  REM "DOS BUCLES"
10  FOR A = 3 TO 5
20  FOR B = 3 TO 10
30  LET C = A * B
40  PRINT A;"POR";B;"ES";C
50  NEXT B
60  NEXT A
```

Cuando lo corras observarás que te doy un mensaje (5/4 ϕ) de que he llenado todas las líneas disponibles en la pantalla. Aprieta CONT, NEWLINE y observa.

Te diré como funciona en los bucles.

Cojo el primer valor del primer bucle ($A = 3$ en el caso anterior) y me voy a la siguiente línea de programa, como es otro bucle cojo el primer valor de este nuevo bucle ($B = \phi$ en este caso).

A continuación multiplico el valor de A (que es 3) por el de B que es ϕ . Imprimo 3 POR ϕ ES ϕ .

Me voy al siguiente valor de B que es 1 (sigo en el segundo bucle) y multiplico A (que es 3 todavía) por el nuevo valor de B e imprimo:

3 POR ϕ 1 ES 3

Y así seguiría en el segundo bucle HASTA (TO) el valor 10, que daría:

3 POR 10 ES 30

Al llegar a este último valor me iría a leer las órdenes de la línea 60, que me indican coger el siguiente valor de A del primer bucle que sería 4.

Todo el proceso del segundo bucle lo repito pero con $A = 4$.

Y así hasta (TO) llegar al valor 5 del primer bucle y al valor 10 del segundo bucle.

Mientras nada me obligue a otra cosa, cada vez que voy al siguiente valor del siguiente ciclo, simplemente le sumo UNO al valor que tuviera la variable en el ciclo precedente.

Por ejemplo:

FOR $A = 1$ TO 4

Primer ciclo $A = 1$

Segundo ciclo $A = 2$

Tercer ciclo $A = 3$

Cuarto ciclo $A = 4$

A veces interesa que los valores los coja con una secuencia distinta a la unidad.

Esto lo entenderé gracias a la Sentencia STEP, con lo cual yo sabré qué cantidad sumar al último valor que haya tomado la variable en el ciclo precedente.

Ejemplo:

FOR A = 1 TO 7 STEP 2

Primer ciclo A = 1

Segundo ciclo A = 3

Tercer ciclo A = 5

Cuarto ciclo A = 7

Ahora te toca programar:

Escribe un programa para hallar las raíces cuadradas (SQR) de los números pares entre 0 y 20.

Compara tu solución con la que te doy a continuación.

Programa de apoyo:

```
1  REM "STEP"  
10  FOR A = 0 TO 20 STEP 2  
20  LET B = SQR A  
30  PRINT B  
40  NEXT A
```

Este mismo programa un poco más completo podría ser:

```
1  REM "STEP"  
10  FOR A = 0 TO 20 STEP 2  
20  LET B = SQR A  
30  PRINT "LA RAIZ CUADRADA DE"; A; "ES", B  
40  NEXT A
```

Lo que viene a decir que desde (FOR) A = 0 hasta (TO) 20, y de dos en dos (STEP 2), deja (LET) B igual a la raíz cuadrada (SQR) de A e imprímelo. Vuelve al siguiente valor del siguiente ciclo.

De aquí a **EJERCICIOS DE REPETICION** y si dudas vuelta a empezar.

Desde ahora me declaro un tipo rutinario, corriente no, pero me gustan las rutinas.

Es más, si no tengo rutinas que seguir no puedo trabajar.

Obviamente cualquier programa es una rutina. Y esa rutina será la que siga mientras no me des otro programa.

Así pues, si asimilamos programa a rutina, una subrutina será un subprograma.

¿... y que es un subprograma o una subrutina?

Pues una subrutina es un programa, con principio y fin que está dentro del programa principal y que puede ser usado por este tantas veces como lo necesite.

Ahora bien ¿cómo sabré yo que tu quieres que yo vaya a la subrutina?

Me lo dices así:

VETE A LA SUBROUTINA

que en BASIC es:

GOSUB

Un ejemplo ayudará a aclarar esto:

Teclea de nuevo el programa de la página 51.

Supongamos que quisieras, para mayor claridad, introducir una línea de guiones entre cada pregunta y su contestación.

Para ello vamos a crear una SUBROUTINA. ¿Tienes listado ya el programa de la página 51? Sin borrar ese listado escribe este programa:

Programa

Comentarios

200 PRINT" -----"

Esto será una subrutina como ya verás.

210 RETURN

Esta palabra inglesa significa retornar o volver.

Este programa está compuesto por las líneas 200 y 210 y para correrlo independientemente de las líneas de programa comprendidas entre 1 y 90, deberás:

Proceso**Comentarios**

1° Teclear RUN 200

Esto me obliga a "correr" el programa a partir de la línea 200

2° NEWLINE (por supuesto)

En la pantalla aparecerá arriba ----- y abajo un mensaje 7/210 con lo que se indica que he cumplido la instrucción recibida en la línea de programa 200 y por eso he imprimido las 6 rayitas, pero que la línea de programa 210 no la puedo ejecutar porque para volver (RETURN) tengo que saber a donde dirigirme.

Ahora tenemos dos programas independientes, el primero va de la línea 1 a la 90 y el otro de la 200 a la 210.

Hagamos del segundo una SUBROUTINA del primero convirtiendo todo en un sólo programa.

Lista todo de nuevo (LIST 1, NEWLINE).

Introduce estas dos líneas de programa:

Proceso**Comentarios**

55 GOSUB 200

Vete a la subrutina de la línea 200.

85 GOSUB 200

Vete a la subrutina de la línea 200.

Corre el programa desde el principio, es decir:

1° Teclea RUN 1

2° NEWLINE (claro)

En la pantalla te aparece: NUMERO?

Escribe, por ejemplo, 20.

En la pantalla te aparece:

NUMERO?

200

Prueba con otro número mayor que 100, p. ej., 600 y tendrás:

NUMERO?

200

NUMERO?

30

NUMERO?

Vaya. ¿Qué ha pasado aquí?

Lista el programa desde el principio (LIST, NEWLINE).

(Recuerda que estás en un bucle y a la espera de la entrada de un número. Si aprietas STOP, NEWLINE provocas un error, con lo cual sales del bucle y ya puedes listar el programa).

Yo he seguido todas tus instrucciones ordenadamente, de tal forma que al llegar a la línea 55 ó 85, según el caso, me encuentro con que me dices "vete a la subrutina que empieza en la línea 200". GOSUB200. Bien, pues allá me voy y leo PRINT "-----", es decir "imprime seis rayitas". Las imprimo y sigo a la línea 210 que me ordena tajante: "retorna" (RETURN). Así lo hago y vuelvo . . . *a la línea de programa inmediata siguiente a la instrucción GOSUB que me mandó a la subrutina.*

Hecho esto continúo con el programa como si nada hubiera ocurrido.

Esta subrutina estaba formada por dos líneas de programa, pero imagínate que hubiera tenido 100 líneas de complicados cálculos. Imagínate además que no conoces las Sentencias GOSUB, RETURN y, consiguientemente hubieras tenido que escribir todas las instrucciones (las 100), tantas veces como las necesitaras.

¿Qué no te lo imaginas?

¿Con qué esas tenemos? Pues entonces:

GOSUB EJERCICIOS CON SUBROUTINAS y RETURN a la página 61.

Una de guerra ... no importa contra quién.

La batalla va a comenzar.

Los jefes y oficiales de las fuerzas a tu mando se reúnen contigo, que es quien tiene la responsabilidad de la operación, para recibir instrucciones respecto a la forma de distribuir las tropas.

Todos esperan, ansiosos, tus órdenes.

Los miras y les dices:

“Señores, que cada cual se ponga donde le plazca”.

El combate comienza necesitándose urgentemente la colaboración de la artillería, tus ayudantes buscan a los artilleros y no los encuentran.

Igual sucede con los infantes, los carros blindados, la intendencia...

Salvo que tu eventual enemigo quede desconcertado por tu “estrategia”, el resultado de esta imaginaria batalla está claro.

Lo usual es preparar un orden de combate que permita hacer maniobrar a las distintas unidades según convenga y de forma rápida y precisa.

Vamos a hacer un esquema.

DISPOSICION ESQUEMATICA DE LAS TROPAS

		FUERZAS DE COMBATE	JEFE AL MANDO	
ENEMIGO	ZONA DE COMBATE	SECTOR 1	INFANTERIA 1	CORONEL VALIENTE
		SECTOR 2	INFANTERIA 2	CORONEL BRAVO
		SECTOR 3	CARROS 1	CAPITAN ACERO
		SECTOR 4	ARTILLERIA 101	CAPITAN OBUS
		SECTOR 5	INFANTERIA 3	CORONEL AUDAZ
		SECTOR 6	INFANTERIA 4	CORONEL VIVO
IMAGINARIO				

En este supuesto hay 6 sectores de combate.

Si hubiera 3000 y te resultara imprescindible saber qué fuerzas hay en el sector 1121 y quien las manda, quizás te puedas imaginar el lío.

Para hacer el ejemplo fácil de manejar por mí (sólo tengo una K) y de seguir por tí lo dejaremos como está en el esquema.

RECUERDA:

Una variable de caracteres queda representada por UNA SOLA LETRA seguida del símbolo \$.

En función de lo que ya conoces, el significado de F\$ representa una variable de caracteres y nada impide que la hagas igual a cualquier cadena de caracteres.

Por ejemplo:

F\$ = "INFANTERIA – 1"

Por idénticas razones podrías escribir:

I\$ = "CORONEL VALIENTE"

y de esta forma podrías seguir:

A\$ = "INFANTERIA – 2"

B\$ = "CORONEL BRAVO"

C\$ = "CARROS – 1"

E\$ = "CAPITAN ACERO"

F\$ = "ARTILLERIA 101"

G\$ = "CAPITAN OBUS"

H\$ = "INFANTERIA – 3"

I\$ = "CORONEL AUDAZ"

K\$ = "INFANTERIA – 4"

L\$ = "CORONEL VIVO"

Y así, continuarías hasta que...

se acaben las letras y te hubieras quedado sin variables y poco o nada hubieras conseguido.

Vamos a usar un ARRAY de caracteres.

Anticipo que hay dos tipos de ARRAY:

- numéricos (trabajan con variables numéricas)
- caracteres (trabajan con variables de caracteres)

Compara las instrucciones anteriores con las que siguen. Estudia las diferencias y los comentarios.

Comentarios

F\$ (1) = "INFANTE-
RIA-1"

A la variable de caracteres F\$ la hemos convertido en F\$ (1) esto, en argot, sería decir:

EFE DOLAR DE UNO O EFE DOLAR SUBUNO, es decir, una variable de caracteres con subíndice (representado por (1)), es igual, en este caso a INFANTERIA-1.

Observa que la cadena de caracteres INFANTE-RIA-1 tiene 12 caracteres.

F\$ (2) = "INFANTE-
RIA-2"

EFE DOLAR DE DOS es igual a una cadena de caracteres de 12 caracteres.

F\$ (3) = "CARROS-1"

También podrías decir que EFE DOLAR DE TRES tiene 8 caracteres.

F\$ (4) = "ARTILLE-
RIA-101"

También: EFE DOLAR SUB 4 tiene 14 caracteres

F\$ (5) = "INTANTE-
RIA-3"

F\$ (5) tiene 12 caracteres.

F\$ (6) = "INFANTE-
RIA-4"

F\$ (6) tiene 12 caracteres.

En este caso F\$ es el prefijo y el número entre paréntesis es el sufijo.

Compara las instrucciones anteriores con las que siguen.

Comentarios

F\$ (1,12)

EFE DOLAR DE UNO, DOCE O EFE DE DOLAR SUB UNO, DOCE. Esto quiere decir que la variable de caracteres F\$ (1) puede tener hasta 12 caracteres.

Comentarios

F\$ (2,12)	F\$ (2,12). La variable de caracteres F\$ (2) puede tener hasta 12 caracteres.
F\$ (3,8)	La variable F\$ (3) puede tener hasta 8 caracteres.
F\$ (4,14)	La variable F\$ (4) puede tener hasta 14 caracteres.
F\$ (5,12)	F\$ (5) puede tener hasta 12 caracteres.
F\$ (6,12)	F\$ (6) puede tener hasta 12 caracteres.

De todas las variables de caracteres F\$ anteriores, la que admite más caracteres es, en este caso, la F\$ (4,14).

El número total de variables cuyo prefijo sea F\$ es 6.

Si yo me encontrara en una línea de programa, la instrucción DIM F\$ (6,14) inmediatamente reservaría en mi memoria espacio suficiente para aceptar *6 variables de caracteres que empiecen por F\$ y que puedan tener hasta 14 caracteres.*

En nuestra "batalla" hemos asimilado DIM F\$ (6,14) a 6 "sectores de combate" en los cuales hay unas tropas que puedes denominar como quieras siempre que tengan hasta 14 caracteres. En este caso F\$ es el prefijo.

Si escribieras DIM J\$ (6,15), y siempre referido al esquema de la página 62, estarías en condiciones de darme 6 "Jefes al mando" cuya denominación no tenga más de 15 caracteres. En este caso J\$ es el prefijo.

Bueno, pues esto son ARRAYS de caracteres:

"Conjuntos de variables de caracteres con el mismo prefijo y el mismo límite de cantidad de caracteres que puede aceptar".

Ejemplo:

Si yo me encuentro la instrucción:

DIM L\$ (100,25)

yo la interpretaría como:

"Dimensiona un conjunto de 100 variables de caracteres que todas tengan el prefijo L\$ y que sean capaces de aceptar hasta 25 caracteres cada una".

Bueno. ¿Y qué? Dirás tú.

No seas impaciente y sigamos con nuestro ejemplo.

A través de las sentencias DIM yo sé las dimensiones y las denominaciones de los ARRAY que voy a utilizar... pero estos ARRAY están vacíos porque no contienen información. Están disponibles.

En nuestra "batalla" tu utilizas F\$ para introducir en mi memoria, por sectores, las Fuerzas de Combate y J\$ los Jefes de las mismas.

¿Cómo me lo debes decir?

Empecemos a programar usando ARRAY de caracteres:

Programa**Comentarios**

30 DIM F\$ (6, 14)

Me ordena dimensionar un ARRAY de caracteres de 6 variables y hasta 14 caracteres y cuyo prefijo es F\$.

De momento está vacío. No contiene nada.

40 DIM J\$ (6, 15)

Me ordena dimensionar un ARRAY de caracteres de 6 variables y hasta 15 caracteres y cuyo prefijo es J\$.

De momento está vacío. No contiene nada.

50 PRINT "SECTOR? ";

Me ordena imprimir "SECTOR?". (Deja 2 espacios entre ? y ").

60 INPUT S

Espero la entrada de un número, que en nuestro caso deberá estar comprendido entre 1 y 6 ambos inclusive y que corresponderá a "un sector de combate".

70 PRINT S

Imprimo el valor de S después de "SECTOR"? puesto que el; de la línea 50 me obliga a ello.

80 PRINT "FUERZAS? ";

Imprimo "FUERZAS? "dejando dos espacios entre ? y ". El ; obliga a la siguiente impresión a ponerse justo a continuación.

90 INPUT F\$ (S)

Espero la entrada de una variable de caracteres.

Por tener el prefijo F\$ yo introduciré este valor en el ARRAY dimensionado en la línea 30.

La S entre paréntesis tendrá el valor que haya recibido en la línea 60.

El número de caracteres está fijo y definido en la línea 30, no puede pasar de 14.

Esto quiere decir que si el valor de S (que tú darás al correr el programa, a través del teclado) fuera por ejemplo 2, yo estaré esperando que me fijes el valor de la variable F\$ (2.14).

El ARRAY se va llenando. Va conteniendo información.

100 PRINT F\$ (S) Imprimo el texto que hayas introducido en la línea 90 y justo a continuación de lo impreso por la línea 80.

110 PRINT "JEFE? "; Imprimo JEFE? dejando dos espacios entre ? y ". El ; obliga a la siguiente impresión a ponerse justo a continuación.

120 INPUT J\$ (S) Vale el mismo comentario de la línea 90 pero referido al prefijo J\$ y al array dimensionado en la línea 40.

130 PRINT J\$ (S) Imprimo el texto que hayas introducido en la línea 120 y justo a continuación de lo impreso por la línea 110.

140 GOTO 50 Me ordena: VETE a la línea 50.

Nota: A partir de ahora, cuando quieras correr un programa no aprietes RUN, NEWLINE es mejor que sigas este procedimiento:

1º GOTO (número de la línea de programa desde donde quieres hacer funcionar el programa).

2º NEWLINE

Ya te explicaré la razón más adelante.

¿Has introducido ya el programa?

Teclea: GOTO 30, NEWLINE.

En la pantalla aparece: SECTOR?

Aprieta 1 y NEWLINE.

Ahora te pregunto: FUERZAS?

Teclea: INFANTERIA-1 y NEWLINE.

JEFE?

Teclea: CORONEL VALIENTE.

(Observa como si dejas un espacio entre CORONEL y VALIENTE yo te imprimo CORONEL VALIENT).

¿Sabes por qué? Porque sólo admito 15 caracteres.

Vuelvo a preguntarte SECTOR?, después FUERZAS?, JEFE?

Deberás seguir este ciclo 6 veces que son los sectores de combate y que, por otro lado, es el número máximo de variables de caracteres que voy a admitir (DIM F\$ (6.14) y DIM J\$ (6.15)).

Prueba a meter un sector 7 y te daré un mensaje de error 3/90 con lo cual te digo que el subíndice 7 está fuera de la dimensión que me has dado y que por esa razón estoy parado en la línea 90.

Lista el programa de nuevo (LIST, NEWLINE). ¿Lo has hecho?

Probablemente pienses que ya no sé que fuerzas hay en el sector 3 y quien las manda.

Vamos, vamos . . . teclea.

Comentarios

PRINT F\$ (3)

Me ordena imprimir el valor que tengo en memoria para la variable F\$ (3).

Y ahora

PRINT J\$ (3)

Lo mismo para la J\$ (3).

A veces pienso que me menosprecias.

Si acaso se te ocurriera correr el programa apretando RUN, NEWLINE, te obedecería claro, pero borraría el contenido de todas las variables por eso es mejor usar GOTO seguido del número de la primera línea de programa útil.

Este programa lo puedes llamar "BATALLA".

Vete a la página 24, sigue sus instrucciones y guarda (SAVE) este programa, tal como está, en un casete.

Desconéctame.

Ahora vuelve a cargarme (LOAD). ¿Ya? (instrucciones en página 22).

Aprieta:

PRINT F\$ (5) (NEWLINE, claro).

Y después:

PRINT J\$ (5) (¡no te olvides nunca de NEWLINE!).

¡Toda la información está guardada en el casete! Tanto el programa como el contenido de las variables.

Lista el programa (LIST, NEWLINE).

Para tener una visión general de los sectores, las fuerzas y sus jefes deberás añadir estas líneas:

Comentarios

150 FOR S = 1 TO 6

160 PRINT S; " ";
F\$ (S); " "; J\$ (S)

170 NEXT S

Con este bucle FOR NEXT la variable S coge todos los valores del 1 al 6, imprimiendo S, dejando un espacio, imprimiendo F\$ (S), dejando un espacio e imprimiendo J\$ (S).

Para obtener esta información te verás obligado a teclear:

GOTO 150, NEWLINE, ya que si teclearas:

GOTO 50, NEWLINE el programa correría hasta la línea 140 donde la Sentencia GOTO 50 situada en la línea 140 me haría empezar de nuevo.

Prueba ambas alternativas.

Graba en el casete este programa sobre el anterior. Lo puedes llamar "BATA-LLA".

Manejar arrays con soltura es de mucha utilidad.

Repasa bien este capítulo y haz los EJERCICIOS CON ARRAYS DE CARACTERES.

Supongamos ahora que acabas de comprar una tienda de ropa y me quieres usar como caja registradora.

Supongamos tambien que, de momento, tienes tres artículos en stock: artículo 1, artículo 2 y artículo 3, y los precios unitarios de venta son 10, 20 y 30 europesetas.

Para que yo te pueda ser útil debes diseñar un programa que me haga funcionar como una caja registradora.

Con los conocimientos de BASIC que tienes hasta ahora, este podría ser un primer programa:

Programa

Comentarios

```
1 REM "CAJA REGIS-  
TRADORA"
```

Con estas instrucciones doy valor (ϕ) a las variables A, B y C, las cuales representarán el importe de la venta por artículo (1, 2, 3).

```
3 LET A =  $\phi$ 
```

```
4 LET B =  $\phi$ 
```

```
5 LET C =  $\phi$ 
```

```
10 CLS
```

Limpio la pantalla.

```
15 PRINT "ARTICULO  
Nº?"
```

Estas dos instrucciones piden el número de identificación del artículo (1, 2 ó 3).

```
20 INPUT N
```

```
25 PRINT "UNIDADES  
VENDIDAS?"
```

Estas dos instrucciones piden la cantidad vendida del artículo anterior.

```
30 INPUT V
```

```
35 LET P1 = 1 $\phi$ 
```

Se definen los precios de los artículos (P1, P2, P3). En este ejemplo hemos supuesto 10, 20 y 30 europesetas.

```
40 LET P2 = 2 $\phi$ 
```

```
45 LET P3 = 3 $\phi$ 
```

```
50 IF N = 1 THEN LET  
A = P1*V
```

Según el artículo vendido determino el importe de la venta.


```
55 IF N = 2 THEN LET
    B = P2*V
```

```
60 IF N = 3 THEN LET
    C = P3*V
```

```
65 LET T = A+B+C
```

T representa el total sucesivo.

```
70 PRINT "OTRO AR-
    TICULO? S/N"
```

Esta bifurcación me lleva al principio del programa si hay otro artículo vendido o nos da el total por venta si hemos llegado al final.

```
75 INPUT R$
```

```
80 IF R$ = "S" THEN
    GOTO 10
```

```
85 PRINT AT 21, 0;
    "TOTAL VENTA", T
```

Es evidente que, si para una tienda con tres artículos, has necesitado un programa de 17 líneas de programa, para el caso de 3.000 artículos hubiera sido una tarea de "chinos" confeccionar el programa.

Para evitarte esta tediosa tarea el BASIC dispone de los arrays numéricos.

Si recuerdas el caso de los arrays de caracteres, los subíndices representaban: el primero el número de variables de caracteres que lo integraban y el segundo el número de caracteres máximos que forman cada variable.

En los arrays numéricos, por el contrario, los subíndices sólo fijan una, y sólo una, de las variables numéricas que integran el array. Nada más.

El array numérico se representa en la misma forma que los caracteres, pero sin el símbolo \$.

Todo array numérico debe ir precedido por la sentencia DIM —letra— seguida por un paréntesis donde se indique el número de variable que contenga, P. EJ.:

ARRAY

Variables que comprende

DIM P (6)

P (1), P (2), P (3), P (4), P (5), P (6)

DIM P (2,3)

(P (1,1), P (1,2), P (1,3)

(P (2,1), P (2,2), P (2,3)

$\left\{ \begin{array}{l} P(1,1), P(1,2) \\ P(2,1), P(2,2) \\ P(3,1), P(3,2) \end{array} \right.$
 DIM P (3,2)

En los ejemplos anteriores los tres arrays pueden contener el mismo número de variables:

P (6) 1 x 6 = 6 variables.

P (2,3). 2 x 3 = 6 variables

P (3,2). 3 x 2 = 6 variables

El programa de la caja registradora podría quedar transformado, usando un array numérico y siguiendo las instrucciones anteriores, en:

Programa

Comentarios

1 LET C = ϕ

5 DIM P (3)

Dimensiono un array numérico de tres variables numéricas denominado P (P (1), P (2) y P (3)).

Aquí almacenaré los precios de los tres artículos (1, 2, 3). Ahora esta vacío.

10 DIM T (3)

Dimensiono otro array de las mismas características que el anterior pero denominado T.

15 FOR X = 1 TO 3

20 PRINT "PRECIO
DEL ARTICULO"; X

Este bucle FOR/NEXT cogerá, como ya sabes, los valores 1, 2 y 3 sucesivamente, preguntando por los precios de los artículos 1, 2 y 3. De esta manera P (X) cogerá los valores P (1) = 10, P (2) = 20 y P (3) = 30.

25 INPUT P (X)

30 NEXT X

35 CLS

Hecho lo anterior, limpio la pantalla. Las variables numéricas están definidas.

40 PRINT "ARTICULO
NUMERO?"

Aquí propiamente empieza el programa de caja registradora y pregunto por el número del artículo vendido.

50 INPUT N

55 PRINT "UDS. VEN-
DIDAS?"

Determina el número de unidades vendidas del artículo anterior.

60 INPUT V

65 IF P (N) <> 0 THEN
LET T = P (N) * V

Según el artículo vendido (N) determino el precio de la venta.

70 LET C = C + 1

Contador cuyo valor será primero 1, luego 2 y finalmente 3.

75 LET T (C) = T

Instrucción que me sirve para poder dar valor a las tres variables que configuran el array definido en la línea 10.

80 PRINT "OTRO
ARTICULO?"

Esta bifurcación me lleva al principio del programa si hay otro artículo vendido o me indica que sume el valor de las tres variables del array DIM e imprima el resultado.

85 INPUT R\$

90 IF R\$ = "S" THEN
GOTO 35

95 PRINT AT21, 0;
"TOTAL VENTA",
T(1) + T (2) + T (3)

A simple vista puede que te parezca más largo este programa que el anterior, no obstante, con pequeñas modificaciones, podrías tenerlo dispuesto para cualquier cantidad de artículos.

Debes recordar que, una vez valoradas las variables del array de precios P (1), P (2) y P (3), no hace falta que cada vez que uses el programa les vuelvas a dar valores, para ello no aprietes RUN y NEWLINE ya que así borras el contenido de todas las variables. Bastará con que teclees GOTO 40, y NEWLINE y así pasar a correr el programa desde la pregunta "ARTICULO NUMERO?".

Si te interesa conocer el contenido de alguna variable, P (2) por ejemplo, bastaría con que apretaras: PRINT P (2), NEWLINE y en el ángulo superior izquierdo yo te diré el valor de esa variable.

EJERCICIOS CON ARRAYS NUMERICOS

Yo puedo leer cadenas de caracteres. Eso ya lo sabes.

Y también puedo leer algunos caracteres consecutivos de cualquier cadena más grande. A estas cadenas contenidas en otras mayores se les llaman **subcadenas**.

El siguiente programa te presenta todas las formas de troceado que yo admito.

Programa	Comentarios
10 LET A\$ = "ROMAR-BELLAVE"	La variable A\$ es valorada.
20 PRINT A\$ (TO 4)	Imprimo todos los caracteres desde el principio hasta el cuarto, de la A\$.
30 PRINT A\$ (3 TO 10)	Imprimo todos los caracteres desde el tercero al décimo, ambos incluidos, de la A\$.
40 PRINT A\$ (8 TO)	Imprimo todos los caracteres del octavo al final de la A\$.
50 PRINT A\$ (6 TO 6)	Imprimo el sexto caracter exclusivamente.
60 PRINT A\$ (1 TO 12)	Imprimo la cadena del principio al final. Igual resultado obtendrías con PRINT A\$.
70 LET B\$ = A\$ (6 TO 10)	Dejamos la variable B\$ igual a la subcadena A\$ indicada.

10 PRINT Apretar P
 R
 I
 N
 T MAL

Hasta aquí te he explicado la parte del lenguaje BASIC que podría ser más difícil de interpretar por tu cerebro.

En este epígrafe tienes una relación de ciertas instrucciones que son útiles según el tipo de aplicación que vayas a desarrollar.

Cada uno de estos comandos o funciones va acompañado de un texto aclaratorio y un ejemplo para fijar las ideas y está agrupado con otros de su mismo tipo.

Funciones aritméticas

Funciones	Comentarios	Ejemplo
ABS	Respondo con el valor absoluto del número o del valor de la variable que sigue a la instrucción.	PRINT ABS C
EXP	Respondo con el valor del antilogaritmo natural del número o del valor de variable que sigue a la instrucción.	PRINT EXP 0.69314718
LN	Respondo con el valor del logaritmo natural del número o del valor de la variable que sigue a la instrucción.	PRINT LN 2
INT	Respondo con el valor entero del número o del valor de la variable que sigue a la instrucción.	INT 10.9 = 10 INT -10.9 = -11
RND	Respondo un valor menor que 1 e igual o mayor que 0. Esta instrucción no va seguida de ningún valor.	RND * 10
SGN	Respondo -1, ϕ ó 1 según que el número o el valor de la variable numérica sea menor que ϕ , cero propiamente o mayor que ϕ .	LETC = (A+B)*SGN B

Funciones aplicadas a cadenas de caracteres

Función	Concepto	Ejemplo
CODE	Respondo al código numérico que tiene para mí el primer carácter de la cadena de caracteres que siga a la instrucción.	PRINT CODE "A"

Funciones	Comentarios	Ejemplo
CHR\$	Respondo el carácter correspondiente al código numérico que siga a la instrucción.	PRINT CHR\$ 38
IN KEY\$	Esta instrucción no va seguida de ningún valor. Interpreto la tecla con el cursor en L.	IF IN KEY\$ = " " THEN STOP
LEN	Respondo el número total de caracteres que componen una cadena de caracteres o una variable de caracteres	10 LET A\$ = "ABCD" 20 PRINT LEN A\$
STR\$	Sólo puede ir seguida de un número o una variable numérica, transformando ese valor en una cadena de caracteres.	STR\$ 20 = "20"
VAL	Sólo puede ir seguida de una cadena o variable de caracteres; transformando esa cadena en un número, siempre y cuando la cadena previa esté formada por caracteres que represente dígitos.	VAL "78" = 78 10 LET A\$ = "10" 20 LET B = VAL A\$ 30 PRINT B
FAST	Cuando leo esta instrucción trabajo cuatro veces más rápido.	20 FAST
SLOW	Cuando leo SLOW vuelvo a la velocidad normal.	30 SLOW
SCROLL	Mueve el contenido de la pantalla una línea hacia arriba.	10 LET A = 1 20 SCROLL 30 PRINT A 40 LET A = A + 1 50 GOTO 20
STOP	Paro la ejecución de un programa.	
CONT	Un programa parado con un STOP puede arrancar de nuevo con esta instrucción.	

SEGUNDA PARTE

**Practicando el BASIC
con el ZX81**

Comentarios a los ejercicios	82
Ejercicios de cálculo	83
<i>Soluciones</i>	84
Ejercicios de modificación	88
<i>Soluciones</i>	89
Ejercicios con textos	92
<i>Soluciones</i>	93
Ejercicios con variables de caracteres	95
<i>Soluciones</i>	96
Ejercicios de impresión en pantalla	100
<i>Soluciones</i>	101
Ejercicios con Goto	104
<i>Soluciones</i>	105
Ejercicios de decisión (IF/THEN)	108
<i>Soluciones</i>	110
Ejercicios de repetición	111
<i>Soluciones</i>	113
Ejercicios de subrutinas	116
<i>Soluciones</i>	117
Ejercicios con ARRAYS de caracteres	120
<i>Soluciones</i>	121
Ejercicios con ARRAYS numéricos	123

Como complemento a la primera parte de "La pequeña gran puerta", damos esta colección de ejercicios con el objetivo de afianzar los conocimientos adquiridos.

Todos los problemas están agrupados en bloques correspondientes con los epígrafes tratados en la teoría y van acompañados de sus correspondientes soluciones.

Las soluciones que proponemos no son, por supuesto, únicas o excluyentes, siendo probable que usted desarrolle ideas perfectamente válidas y aceptables por su ZX81 en su configuración mínima de 1K RAM.

En cualquier caso vea y analice los comentarios que acompañan a algunos programas, ya que le sugerirán nuevos caminos y/o le aclararán algunos conceptos.

Le recomendamos que se esfuerce en buscar la respuesta a los ejercicios antes de dirigirse a la solución de los mismos, los cuales, por otra parte, han sido pensados de acuerdo con el grado de dificultad que el estudiante de este nivel puede resolver.

La práctica hará más por su aprendizaje que muchas horas de teoría.

Haz los siguientes programas que te permitan respectivamente:

- 1º Sumar hasta diez números que tu des a través del teclado.
- 2º Sumar los dos primeros números y restar el tercero.
- 3º Sumar los dos primeros números y restar el tercero, multiplicando el resultado anterior por un cuarto número.
- 4º Elevar cualquier número a cualquier potencia.
- 5º Hallar la raíz cuadrada de un número.

Intenta razonar como "entendería" yo cada cuestión. ¿Una ayudita?

Ayuda al ejercicio 3

"Espera la entrada (INPUT) de un número que llamarás A. Después espera la entrada (INPUT) de otro número que llamarás B. Después espera la entrada (INPUT) de otro número que llamarás C. Después espera la entrada de otro número que llamarás D.

Después deja (LET) E igual a la suma (+) de A y B.

Después deja (LET) F igual a la resta (-) de E y C.

Después deja (LET) G igual al producto (*) de F por D.

Te aconsejo que antes de empezar a programar escribas el programa en términos parecidos a los ejemplos anteriores.

Una solución a estos ejercicios la tienes a continuación.

Siempre que empieces a teclear un programa no olvides limpiar mi memoria. Para ello tienes dos caminos, o bien desconectas el jack de entrada de corriente, o bien aprietas NEW, NEWLINE.

Ayuda al ejercicio 5

"Espera la entrada (INPUT) de un número que llamarás A. Después deja (LET) C igual a la raíz cuadrada (SQR) de C. Después imprime el valor de C.

*Ejercicio 1***Programa****Comentarios**

1 REM "EJERCICIO 1"

La instrucción REM no afecta al programa. Sólo sirve para recordarte aquello que tú quieras resaltar.

10 INPUT N1

Estas instrucciones me ordenan esperar la entrada de 10 números.

20 INPUT N2

30 INPUT N3

Repasa en la página número 18 y recordarás lo que es una variable numérica.

40 INPUT N4

50 INPUT N5

También podrías haber usado las variables numéricas A, B, C, D, etc., en vez de N1, N2, N3, etc.

60 INPUT N6

70 INPUT N7

80 INPUT N8

90 INPUT N9

100 INPUT N10

110 LET S = N1 + N2 +
+ N3 + N4 + N5 + N6 +
+ N8 + N9 + N10

Me ordena "dejar" la variable numérica S igual a la suma de las variables N1, N2, etc.

120 PRINT S

Me ordena "imprimir" el valor de S.

RECUERDA:

- 1º No olvides apretar NEWLINE después de completar cada línea del programa. P. ej.

Escribe 10

Aprieta INPUT

Escribe N1

Aprieta NEWLINE

- 2° Para correr (RUN) el programa, es decir, para hacerlo funcionar, aprieta RUN, NEWLINE.

Ejercicio 2

Programa

Comentarios

1 REM "EJERCICIO 2"	REM no afecta al programa, sólo sirve para "recordarte" textos.
10 INPUT A	Espero la entrada de la variable numérica A.
20 INPUT B	Espero la entrada de la variable numérica B.
30 INPUT C	Espero la entrada de la variable numérica C.
40 LET D = A + B	Dejo D igual a A + B.
50 LET E = D - C	Dejo E igual a D - C.
60 PRINT E	Imprimo el valor de E.

Ejercicio 3

(Hay una solución alternativa en la página 87. (Ejercicio 3A)).

Añade las siguientes líneas de programa al programa del ejercicio 2.

35 INPUT F	Espero la entrada de la variable numérica F.
60 LET G = E * F	Al escribir esta línea de programa (y apretar NEWLINE por supuesto) borras la anterior y "deja" la variable G igual a E * F.
70 PRINT G	Imprimo el valor de G.

Si quieres puedes ir grabando (SAVE) estos programas en un casete y así tendrás una colección de ejercicios grabados y, en todo caso, practicas la grabación (SAVE) y carga (LOAD) de programas.

*Ejercicio 4***Programa****Comentarios**

1 REM "EJERCICIO 4"

10 INPUT X

X es la variable que representa la base.

20 INPUT Y

Y es la variable que representa el exponente.

30 LET Z = X**Y

** (SHIFT, H) me ordena elevar X a la potencia Y.

40 PRINT Z

*Ejercicio 5***Programa****Comentarios**

1 REM "EJERCICIO 5"

10 INPUT A

20 LET B = SQR A

SQR me ordena extraer la raíz cuadrada de A.
SQR se obtiene así:

1º SHIFT, NEWLINE simultáneamente (el cursor cambia a F)

2º SQR (debajo de la tecla H).

30 PRINT B

*Ejercicio 3A***Programa****Comentarios**

1 REM "EJERCICIO 3A"

10 INPUT A

20 INPUT B

30 INPUT C

40 INPUT D

50 LET E = (A + B - C) * D

60 PRINT E

Ejercicio 6. Si tienes el ejercicio 1 grabado en un casete, cárgalo (LOAD) en mi memoria, si no teclea el programa de nuevo.

Cuando lo tengas listado (LIST, NEWLINE) en la pantalla conviértelo de forma que sirva para sumar sólo dos números.

Ejercicio 7. Convierte este último en el programa del ejercicio 3.

Ejercicio 8. Convierte este último en el programa del ejercicio 3A.

Ejercicio 6

Proceso	Comentarios
30 NEWLINE	Cualquier número de línea de programa seguido de NEWLINE borra el contenido que pudiera tener esa línea de programa.
40 NEWLINE	
50 NEWLINE	
60 NEWLINE	
70 NEWLINE	
80 NEWLINE	
90 NEWLINE	
100 NEWLINE	

Ahora desplaza el puntero hasta la línea 110 apretando SHIFT y 6 simultáneamente.

Edita (EDIT) esta línea apretando SHIFT y 1.

La K está entre 110 y LET. Aprieta SHIFT y 8.

La K cambió a L. Sigue apretando SHIFT y 8 hasta que la L esté situada a la derecha de N10.

Empieza a borrar (RUBOUT) apretando SHIFT y ϕ hasta que la L se sitúe a la derecha de N2.

Aprieta NEWLINE.

El programa está modificado. Ya puedes correrlo (RUN, NEWLINE).

Sólo podrás sumar dos números.

Ejercicio 7

Proceso	Comentarios
30 INPUT C	Introduce esta línea (no olvides NEWLINE).
115 LET E = S - C	Introduce esta línea de programa.

Desplaza el cursor hasta la línea 120 mediante SHIFT y 6.

Edita (EDIT) la línea 120 (SHIFT y 1).

Borra la S y sustitúyela por E.

En este punto el programa funciona exactamente igual que el del ejercicio 2, aunque las variables usadas sean distintas.

35 INPUT F

Introduce esta línea de programa.

120 LET G = E * C

Sustituimos la línea 120 (que ya no será útil) por esta otra que cumple la función de multiplicar E por C. (La variable F del ejercicio 3 es aquí la C).

130 PRINT G

Imprime el valor de G.

En este ejercicio debes concluir que no importa el número de línea de programación que utilices con tal de que sea mayor que el número de la línea anterior y, menor que el de la línea siguiente: lo único importante es que yo sepa el orden con que debo realizar mis operaciones!

Graba en tu casete el programa modificado.

Ejercicio 8

Carga (LOAD) en mi memoria el programa anterior.

Proceso

Comentarios

Te aconsejo que cambies todas las variables a las que figura en el ejercicio 3A.

110 LET E = (A +
+ B - C) * D

Sustituye la línea 110 por esta otra.

115 NEWLINE

Borra las instrucciones contenidas en las líneas de programación 115 y 120 dejándolas "vacías".

120 NEWLINE

130 PRINT E

Es más rápido escribir esta línea (por su longitud) que editarla.

Corre (RUN, NEWLINE) el programa y haz algunas pruebas.

Para acabar estas prácticas sobre modificaciones y cambios te enseñaré un buen truco.

Si quieres “editar” una determinada línea de programación puedes evitarte la rutina de bajar o subir el “puntero” simplemente apretando LIST seguido del número de la línea de programa que te interese modificar (y NEWLINE, claro). De esta forma el puntero saldrá en la línea deseada y sólo tendrás que editar (SHIFT, 1).

Prueba LIST 110.

¿Vale? . . . de nada, de nada.

Ejercicio 9. Lista el programa del ejercicio 8.

¿Cómo harías para que la pantalla te pregunte?:

NUMERO 1? (antes de darme el primer número)

NUMERO 2? (antes de darme el segundo número)

NUMERO 3? (antes de darme el tercer número)

NUMERO 4? (antes de darme el cuarto número)

Ejercicio 10. Sobre el resultado del ejercicio 9. ¿Cómo harías para que después de darme cada número yo lo imprima a continuación del NUMERO? correspondiente

Ejercicio 11. Sobre el resultado del ejercicio 11. ¿Cómo harías para que te imprima "EL RESULTADO =" y a continuación el valor de E?

¡No mires las soluciones!

Primero inténtalo tú. Estos humanos . . .

*Ejercicio 9***Proceso****Comentarios**

5 PRINT "NUMERO 1?" Añade esta línea
15 PRINT "NUMERO 2?" Añade esta línea
25 PRINT "NUMERO 3?" Añade esta línea
32 PRINT "NUMERO 4?" Añade esta línea

Deja un espacio (tecla SPACE) entre NUMERO y el número y entre ? y las ".

Córrelo (RUN, NEWLINE). Mi respuesta te la doy justo debajo de NUMERO 4?".

*Ejercicio 10***Proceso****Comentarios**

Lista el programa (LIST, NEWLINE)

Añade un punto y coma (;) al final de las líneas de programa 5; 15; 25 y 32.

12 PRINT A Añade esta línea
22 PRINT B Añade esta línea
31 PRINT C Añade esta línea
37 PRINT D Añade esta línea

Corre (RUN, NEWLINE) el programa modificado con algunos números y compruébalo.

Prueba a quitar los ; de las líneas 25 y 32. ¿Dónde pongo los números con respecto a los dos primeros?

... ¡Ah! Los puntos y comas me atraen ...

*Ejercicio 11***Proceso****Comentarios**

Lista el programa (LIST, NEWLINE)

120 PRINT "EL RESULTADO I"?

Añade esta línea

Corre (RUN, NEWLINE) el programa con cuatro números cualquiera y prueba esta otra posibilidad:

Proceso**Comentarios**

120 PRINT "EL RESULTADO ="; E

130 NEWLINE

Borro la línea 130

Corre (RUN, NEWLINE) esta variación con los mismos (u otros) cuatro números.

¡El resultado es igual!

La Sentencia PRINT te acepta la posibilidad de concatenar textos y variables numéricas por medio de comas (,) y puntos y comas (;). Así puedes eliminar líneas de programa innecesarias.

Prueba a sustituir el punto y coma de la línea 120 por una coma y corre el programa de nuevo.

Verás que el resultado se ha separado un poco. Tiene una explicación: "La coma me obliga a empezar a imprimir a partir de la columna 16".

Ejercicio 12

Escribe un programa que te pregunte tu nombre y que, al dárselo, imprima:

HOLA tu nombre

RECUERDA:

PROGRAMAR SERA TAN NECESARIO COMO LEER

Ejercicio 13

Haz un programa que te pregunte primero tu nombre, después tu apellido, tu teléfono y por último tu dirección, y que cada respuesta quede registrada en una variable de caracteres.

NOMBRE?

APELLIDO?

TELEFONO?

DIRECCION?

Ejercicio 14

Haz un programa que te pregunte tu profesión y tu edad y que cada respuesta quede registrada en una variable de caracteres.

Ejercicio 15

Ir directamente a su solución.

Ejercicio 12

```
1  REM "EJERCICIO 13"
10 PRINT "TU NOMBRE"
20 INPUT N$
30 PRINT "HOLA"; N$
40 PRINT "RECUERDA:"
50 PRINT "PROGRAMAR SERA TAN NECESARIO COMO LEER"
```

*Ejercicio 13***Programa****Comentarios**

```
1 REM "EJERCICIO 13"
10 PRINT "NOMBRE?"
20 INPUT N$
30 PRINT "APELLIDO?"
40 INPUT A$
50 PRINT "TELEFONO?"
60 INPUT T$
70 PRINT "DIRECCION?"
80 INPUT D$
```

Aunque la entrada sea un número yo lo voy a considerar como una cadena de caracteres, ya que lo asocio a una variable de caracteres denominada T\$

Sigue los pasos que te indico a continuación y aprenderás algo realmente interesante.

Proceso**Comentarios**

1° RUN, NEWLINE

Con esta instrucción me ordenas que corra el programa, es decir (como ya sabes), que lo haga funcionar.

PERO ADEMAS PONGO A CERO (DEJO VACIAS) TODAS LAS VARIABLES NUMERICAS Y DE CARACTERES.

2°

Contesta a las preguntas: nombre, apellido, etc.

3° PRINT A\$,
NEWLINE

Con esta instrucción me ordenas que imprima el valor de A\$

Te daré tu nombre en la esquina superior derecha.

Prueba con las otras variables.

AHORA NO BORRES ESTE PROGRAMA. *No aprietes RUN*. INTRODUCE A CONTINUACION EL PROGRAMA DEL PROXIMO EJERCICIO

Ejercicio 14.

Tal como sigue:

Proceso**Comentarios**

1 REM "EJERCICIO 14"

10 PRINT "PROFESION?"

20 INPUT Z\$

Observa que utilizo una variable de caracteres (Z\$) distinta a las usadas en el ejercicio 13.

30 PRINT "EDAD?"

40 INPUT Y\$

Vale el comentario de la línea 20.

50 PRINT "NOMBRE:";
N\$

Yo mantengo en mi memoria al valor de N\$ del programa anterior al margen de que haya en mi memoria un programa nuevo . . . siempre que no me hayas desconectado o apretado RUN o NEW o uses las mismas variables en el programa nuevo.

60 PRINT "APELLI-
DOS:"; A\$

Yo mantengo el valor de las variables mientras que tu no las cambies.

70 PRINT "TELEFO-
NO:"; T\$

80 PRINT "DIREC-
CION:"; D\$

90 PRINT "PROFE-
SION:"; Z\$

100 PRINT "EDAD:"; Y\$

ATENCION: No corras el programa con RUN.

A partir de ahora usa GOTO (seguido del primer número de línea de programa), y NEWLINE como siempre.

RECUERDA: Cuando usas RUN, NEWLINE yo correré el programa, pero ADEMÁS TE BORRO EL CONTENIDO DE LAS VARIABLES.

Por tanto aprieta, en este caso, GOTO 1 (o cualquier número anterior a 10, ya que 10 es la primera línea de programa útil) y a continuación NEWLINE.

Contesta a las preguntas del programa.

Cuando lo hayas hecho, e inmediatamente, imprimiré toda la información que contienen las variables de caracteres que has utilizado y que tengo en memoria.

Me puedes listar como siempre (LIST, NEWLINE).

Tendrás oportunidad de apreciar las ventajas de todo esto.

De momento te aconsejo que guardes (SAVE) este programa en un casete, con los valores que tengan las variables. Lo llamaremos "EJERCICIO 14".

Lo utilizaremos en el ejercicio 15.

Ejercicio 15

Supongo que has guardado el programa anterior.

Habrás notado lo práctico que es el cuenta vueltas de tu magnetofón para saber dónde empieza cada programa . . .

Bien, carga (LOAD) el ejercicio 14.

¿Ya lo has hecho?

RECUERDA: Si aprietas RUN, NEWLINE te quedas con el programa en mi memoria, pero "borro" el contenido de las variables.

Vamos a hacer pruebas:

Prueba	Resultado
1º Aprieta GOTO 50	Impresión en pantalla de tus datos
2º Aprieta GOTO 1	Te pregunto tu profesión. Si me das otra distinta, sustituiré el valor de la variable Z\$. Igual sucederá con Y\$ Varía estos datos. Observa el resultado.

No grabes en el casete esta modificación.

Este ejercicio ha tenido por objeto mostrarte que las variables también pueden ser "guardadas" en el casete.

Ejercicio 16

Haz un programa que te imprima automáticamente un * en la esquina superior izquierda de la pantalla, otro en el centro y otro en la esquina inferior derecha. (Usando PRINT AT).

(No olvides que las dos últimas líneas no están disponibles para tí).

Repite el programa poniendo a continuación del * las vocales AEIOU.

Ejercicio 17

Sabrías hacer el programa anterior usando PRINT y PRINT TAB.

Ejercicio 18

Carga (LOAD) el "EJERCICIO 13" en mi memoria.

Cambia el programa (SIN ALTERAR EL CONTENIDO DE LAS VARIABLES) de forma que imprima los textos a la izquierda de la pantalla y el contenido de las variables de caracteres a partir de la columna 20 y en la misma línea (Usa PRINT TAB).

*Ejercicio 16***Programa****Comentarios**

1 REM "EJERCICIO 16"

10 PRINT "***"

También vale PRINT TAB ϕ ; "***" o PRINT AT ϕ, ϕ ; "***" con una columna de variación.

20 PRINT AT 10,16; "***"

También podrías haber escrito:

20 PRINT, "***"

30 PRINT AT 21, 32; "***"

*Ejercicio 17***Programa****Com****Comentarios**

1 REM "EJERCICIO 17"

10 PRINT "***"

20 PRINT

Una instrucción PRINT sin nada a continuación me obliga a dejar una línea en blanco.

30 PRINT

40 PRINT

50 PRINT

60 PRINT

70 PRINT

Este procedimiento de espaciar las líneas de impresión en pantalla es incómodo y consume una gran capacidad de memoria.

80 PRINT

90 PRINT

100 PRINT

110 PRINT TAB 16; "***"

120 PRINT

Programa**Comentarios**

130 PRINT

140 PRINT

150 PRINT

Acostúmbrate a usar PRINT TAB, PRINT AT y siempre que puedas punto y coma (;) y coma (,)

160 PRINT

170 PRINT

180 PRINT

190 PRINT

200 PRINT

210 PRINT

220 PRINT

230 PRINT TAB 31; "*"

*Ejercicio 18***Programa****Comentarios**50 PRINT "NOMBRE";
TAB 20; N\$

Cambia la línea 50 por ésta.

60 PRINT "APELLIDOS:";
TAB 20; A\$

Fíjate como se ha colocado aquí TAB.

70

80

Cambia tú estas líneas

90

100

Cuando hayas acabado las modificaciones y quieras comprobar el resultado, deberás apretar GOTO 50, de esta forma yo te correré el programa desde la línea 50 sin borrar las variables y sin hacerte las preguntas que van en las líneas de programa anteriores.

¿Has seguido mis instrucciones?

Bueno, pues ahora aprieta GOTO 1 y ¿a ver cómo te las ingenias para que te dé la impresión por pantalla ... sin alterar el contenido de las variables?

Nada, nada ¡Estás en un bucle!

Ejercicio 19

Haz un programa que imprima las vocales (A, E, I, O, U) una debajo de otra en la pantalla y hasta que esta se llene.

Usa instrucciones PRINT y GOTO.

Ejercicio 20

¿Se puede cambiar el valor numérico de la instrucción GOTO por una variable numérica?

Ejercicio 21

Haz el ejercicio 19 usando PRINT AT, GOTO y,,

*Ejercicio 19***Programa****Comentarios**

10 PRINT "A"

A

20 PRINT "E"

E

30 PRINT "I"

I

40 PRINT "O"

O

50 PRINT "U"

U

60 GOTO 10

A

E

I

O

U

A

E

I

O

U

A

E

I

O

Hasta que te dé un error
de pantalla llena

Ejercicio 20

Si se puede. Cambia la línea 60 del programa anterior por 60 GOTO X y ahora introduce LET X = 10 (sin número de línea) y aprieta NEWLINE.

El programa correrá igual que antes ya que a X le hemos dado el valor 10.

Prueba haciendo:

LET X = 20

LET X = 30

LET X = 40

LET X = 50

*Ejercicio 21***Programa****Comentarios**

5 LET C = 0

5. Se define el valor de la variable C.

10 PRINT AT C, 0; "A", ,
"E", , "I", , "O", , "U"

10. La doble coma obliga a un salto de línea imprimiendo por tanto para cada valor de C:

A

E

I

O

U

20 LET C = C + 5

Esta línea deja a C igual al valor anterior que tuviera más cinco y, así, obtenemos.

A 1

E 2

Programa

Comentarios

I 3

O 4

U 5

A 6

E 7

I 8

O 9

U 10

" "

" "

" "

30 GOTO 10

Ejercicio 22

Desarrolla un programa que te diga:

PASE o ESPERE, según el color Verde o Azul de un semáforo. El color lo introduces tú por el teclado.

Ejercicio 23

Para ser candidata a Mis Universo se debe medir más de 1,7 metros y pesar menos de 60 kilos.

¿Podrías hacer un programa en que dando el nombre, estatura y peso de cada aspirante te diga si reúne o no los requisitos mínimos?

Ejercicio 24

¿Qué tendrías que hacer en el programa anterior para que la Srta. Juanita del Enchufe y Alto Voltaje fuera admitida, sea cual fuere su talla y peso, en el concurso de belleza anterior?

Ejercicio 25

Qué instrucciones pondrías al principio de un programa para que sólo funcionara con las personas que conozcan la "clave secreta"? En el ejercicio anterior por ejemplo.

*Ejercicio 22***Programa****Comentarios**

10 PRINT "COLOR?"

Para obtener caracteres en negativo debes trabajar en modo gráfico:

15 INPUT C\$

1º Aprieta

20 PRINT C\$

SHIFT y 9 (GRAPHICS)

25 IF C\$="ROJO" THEN
GOTO 40

(el cursor cambia a G)

30 PRINT AT 14,10 "PASE"

2º Teclea tu texto

Escribe PASE Y ESPERE en negativo.

35 GOTO 50

3º Para volver al modo normal repite la operación primera.

40 PRINT AT 14,10;
"ESPERE"50 PRINT AT 21,0; "OTRO
COLOR?"

55 INPUT R\$

60 IF R\$ = "NO" THEN
STOP

65 CLS

70 GOTO 10

*Ejercicio 23***Programa****Comentarios**

5 PRINT "NOMBRE?",

10 INPUT N\$

11 PRINT N\$

15 PRINT "TALLA?",

20 INPUT T

21 PRINT T

25 PRINT "PESO?",

30 INPUT P

31 PRINT P

35 IF T <> 1,7 OR P <> 60 THEN GOTO 70

Observa cómo la operación lógica OR implica que la instrucción THEN se cumpla cuando alguna de las condiciones IF es cierta.

40 PRINT N\$, "ADMITIDA"

45 PRINT AT 21,0; "OTRA CANDIDATA?"

50 INPUT O\$

No confundas la letra O con el número ϕ

55 IF O\$ = "NO" THEN STOP

60 CLS

65 GOTO 5

70 PRINT N\$, "RECHAZADA"

75 GOTO 45

Ejercicio 24

Modifica la línea 35:

35 IF T < 1,7 OR P > 60 THEN IF N\$ <> "JUANITA DEL ENCHUFE"
60 THEN GOTO 70

Ejercicio 25

Programa

Comentarios

1 PRINT "CLAVE?"

2 INPUT C\$

3 IF C\$ <> "CUALQUIER COMBINACION DE CARACTERES" THEN STOP

5 PRINT "NOMBRE?"

10 INPUT N\$

11 PRINT N\$

¿Sabes cual es la clave secreta?

(¿Sabes de qué color es el caballo blanco de . . .?)

Ejercicio 26

Usando el ciclo FOR/NEXT haz un programa que te imprima la columna cero con la palabra "columna".

Ejercicio 27

El mismo pero que se imprima en la columna 12.

Ejercicio 28

El mismo, pero que llene toda la pantalla.

Ejercicio 29

Añade las líneas de programa suficientes como para conocer el número de veces que se ha imprimido la palabra "columna" antes del error 5/26.

Ejercicio 30

Escribe un programa que te de en pantalla los números impares del 1 al 50, evitando el error de "pantalla llena".

Ejercicio 31

Modifica el programa anterior de forma que sólo responda la suma de los números impares de 1 a 50.

Ejercicio 32

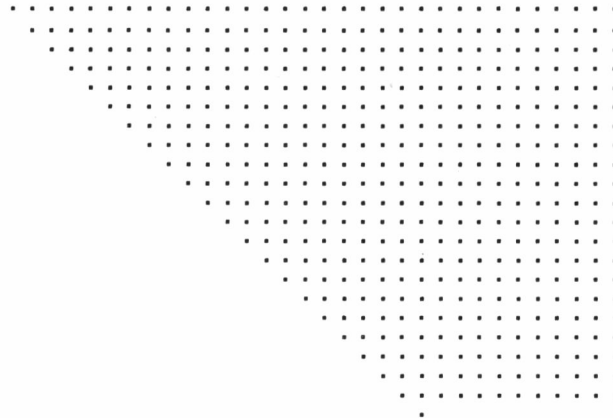
Escribe un programa que te imprima la palabra BASIC "parpadeando". Utiliza el modo gráfico y un bucle FOR/NEXT.

Ejercicio 33

Modifica el programa anterior de modo que el "parpadeo" sea más lento.

Ejercicio 34

Con dos bucles FOR/NEXT enlazados crea una imagen de puntos en pantalla similar a esta:

*Ejercicio de interpretación*

Cómo interpretas este programa:

```
5 FOR E = 1 TO 2
10 LET E = 1

20 LET A = 2

30 LET C = E + A

40 PRINT C

50 NEXT E
```

*Ejercicio 26***Programa**

```
5 FOR X = 0 TO 21
10 PRINT "COLUMNA"
20 NEXT X
```

Comentarios

Se imprime la palabra "columna" 22 veces a partir de la columna 0 y línea tras línea.

*Ejercicio 27***Programa**

```
5 FOR X = 0 TO 21
10 PRINT TAB 12;
  "COLUMNA"
20 NEXT X
```

Comentarios

TAB 12 obliga a comenzar en la columna 12

*Ejercicio 28***Programa**

```
5 FOR X = 0 TO 21
10 PRINT "COLUMNA";
20 NEXT X
30 GOTO 5
```

Comentarios

El ; obliga a imprimir una palabra tras otra. La línea 30 obliga a repetir el ciclo FOR/NEXT hasta obtener un error de "pantalla sin espacio de impresión".

*Ejercicio 29***Programa**

```
4 LET C = 0
5 FOR X = 0 TO 21
10 PRINT "COLUMNA";
11 LET C = C + 1
20 NEXT X
30 GOTO 5
```

Comentarios

Hemos añadido un contador.

Cada vez que el bucle pasa por aquí sumo una unidad.

Cuando la pantalla se haya llenado, aprieta PRINT C y obtendrás el contenido de la variable C que es el contador.

*Ejercicio 30***Programa****Comentarios**

```
10 FOR X = 1 TO 50 STEP 2
20 PRINT X,
30 NEXT X
```

*Ejercicio 31***Programa****Comentarios**

```
5 LET C = 0
10 FOR X = 1 TO 50 STEP 2
11 LET C = C + X
30 NEXT X
35 PRINT AT 21, 0; "VALOR
DE LA SUMA . . ."; C
```

*Ejercicio 32***Programa****Comentarios**

```
10 FOR X = 0 TO 100
20 PRINT AT 10, 15;
"BASIC"
30 CLS
40 PRINT AT 10, 15;
"BASIC"
50 NEXT X
```

Estos ejercicios pueden servirte a la hora de resaltar ciertos resultados en tus futuros programas.

Escribe la palabra BASIC de la línea 20 en negativo (modo gráfico)

Escribe la palabra BASIC de la línea 40 normalmente.

*Ejercicio 33***Programa****Comentarios**

10 FOR X = 0 TO 10

Los bucles FOR/NEXT no se pueden cruzar, pero, evidentemente, si se pueden suceder.

20 PRINT AT 13,15;
"BASIC"

Escribe BASIC en negativo (línea 20)

50 NEXT X

60 FOR X = 0 TO 10

70 PRINT AT 13,15;
"BASIC"

Escribe BASIC normalmente (línea 70)

80 NEXT X

90 GOTO 10

*Ejercicio 34***Programa****Comentarios**

10 FOR F = 0 TO 21

20 FOR C = F TO 31

30 PRINT AT F, C; "."

40 NEXT C

50 NEXT F

Observa cómo varía el valor de entrada del segundo bucle (C) con el valor de entrada del primero (F) y consiguientemente, el lugar de arranque de PRINT.

Ejercicio 35

¿Se te ocurre alguna forma de combinar el programa del ejercicio 34, con el del ejercicio 32, para que, convirtiendo éste en una subrutina de aquél, la palabra BASIC cruce parpadeando, la pantalla de izquierda a derecha?

Ejercicio 36

Escribe un programa para obtener la tabla de multiplicar por 5, separando cada número por una línea de asteriscos.

5 POR 0 = 0

5 POR 1 = 5

5 POR 2=10

5 POR 3=15

5 POR 4=20

5 POR 5=25

5 POR 6=30

5 POR 7=35

5 POR 8=40

5 POR 9=45

5 POR 10=50

Ejercicio 37

Rediseña el ejercicio 1 con una subrutina de forma que te pregunte por pantalla: "NUMERO?" antes de la entrada de cada número y te de las "GRACIAS" después. Límitalo a cinco sumandos.

Ejercicio 38

Como aprovecharías las subrutinas del ejercicio anterior para usarlas en las del ejercicio 3A.

Nota: Pon las subrutinas al final de los programas. En la práctica usa las subrutinas sólo si te ahorra trabajo personal e instrucciones.

*Ejercicio 35***Programa****Comentarios**10 FOR F = ϕ TO 21

En este ejercicio se anidan dos bucles FOR/NEXT con una subrutina.

20 FOR C = F TO 31

30 GOSUB 90

40 NEXT C

50 NEXT F

90 FOR Y = ϕ TO 2100 PRINT AT F, C;
"BASIC"

Escribe BASIC en negativo.

110 CLS

140 PRINT AT F, C;
"BASIC"

160 NEXT Y

170 RETURN

*Ejercicio 36***Programa****Comentarios**10 FOR X = ϕ TO 10

20 LET A = X * 5

30 PRINT "5 POR "; X; "="; A

40 GOSUB 100

50 NEXT X

100 PRINT "*****"

110 RETURN

*Ejercicio 37***Programa****Comentarios**

5 GOSUB 100	Las subrutinas conviene ponerlas después de la última instrucción del programa.
10 INPUT N1	
15 GOSUB 200	
16 GOSUB 100	
20 INPUT N2	
25 GOSUB 200	
26 GOSUB 100	
30 INPUT N3	
35 GOSUB 200	
36 GOSUB 100	
40 INPUT N4	
45 GOSUB 200	
46 GOSUB 100	
50 INPUT N5	
55 GOSUB 200	
70 LET S = N1 + N2 + N3 + N4 + N5	
70 PRINT "LA SUMA ES "; S; ". DE	
NADA"	
80 STOP	
100 PRINT "NUMERO?"	
110 RETURN	
200 PRINT "GRACIAS"	
210 RETURN	

Ejercicio 38

En general puedes aprovechar cualquier programa como una subrutina de cualquier otro. Basta con que no olvides la instrucción RETURN y no duplicar las variables, con lo cual perderías información o provocarías errores.

Ejercicio 39

Dimensiona un array de caracteres que permita alojar 4 variables de caracteres de hasta 5 caracteres cada una.

Ejercicio 40

Dimensiona otro array para 5 variables de hasta cuatro caracteres.

Ejercicio 41

¿Qué significan estas líneas de programa?

10 DIM A\$ (15,5)

15 DIM B\$ (15,5)

Ejercicio 42

Haz un programa que te permita dar valores automáticamente a las 4 variables del array DIM A\$ (4,7).

Ejercicio 43

Interpreta este programa:

10 DIM A\$ (4,7)

20 DIM B\$ (4,7)

30 FOR X = 1 TO 4

40 LET B\$ (5-X) = A\$ (X)

50 NEXT X

Ejercicio 44

Escribe un programa que ponga en orden alfabético no más de 4 palabras de hasta 7 letras cada una.

Ejercicio 39

DIM A\$ (4,5)

Ejercicio 40

DIM A\$ (5,4)

Ejercicio 41

Los arrays en general se dimensionan al principio de los programas.

En este caso tenemos dos arrays de caracteres (símbolo \$) denominados A y B pudiendo contener cada uno de ellos 15 variables de hasta 5 caracteres.

Ejercicio 42

10 DIM A\$ (4,7)

Si alguna de las 4 variables tuviera más de 7 caracteres, yo ignoraría el exceso.

20 FOR X = 1 TO 4

30 INPUT A\$ (X)

40 NEXT X

Ejercicio 43

Se transfieren los valores de las variables contenidas en el array A\$ al B\$ de tal forma que A\$ (1) = B\$ (4), A\$ (2) = B\$ (3) y así sucesivamente.

*Ejercicio 44***Programa****Comentarios**

10 DIM A\$ (4,7)

Con las cuatro primeras instrucciones introducimos las palabras que valoran las variables de array.

20 FOR X = 1 TO 4

Programa**Comentarios**

30 INPUT A\$ (X)

40 NEXT X

50 IF A\$ (1) < = A\$ (2) THEN GOTO 90

60 LET B\$ = A\$ (1)

70 LET A\$ (1) = A\$ (2)

80 LET A\$ (2) = B\$

90 IF A\$ (2) < = A\$ (3) THEN GOTO 1 40

100 LET B\$ = A\$ (2)

110 LET A\$ (2) = A\$ (3)

120 LET A\$ (3) = B\$

130 GOTO 50

140 IF A\$ (3) < = A\$ (4) THEN GOTO 1 90

150 LET B\$ = A\$ (3)

160 LET A\$ (3) = A\$ (4)

170 LET A\$ (4) = B\$

180 GOTO 90

190 PRINT A\$ (1)

200 PRINT A\$ (2)

210 PRINT A\$ (3)

220 PRINT A\$ (4)

Con el resto de las instrucciones se comparan todas las variables de array ordenándolas, según el "peso" relativo que yo tengo de ellas y que corresponde a su orden alfabético.

Ejercicio 45

Dimensiona un array numérico que pueda contener hasta 10 variables numéricas.

Ejercicio 46

Escribe un programa que te permita valorar las 10 variables numéricas del array anterior.

Ejercicio 47

Dimensiona un array numérico bidimensional que pueda contener hasta 10 variables numéricas.

Ejercicio 48

Escribe un programa que te permita valorar las 10 variables numéricas del array anterior. Guarda en un casete el programa y el array valorado.

Ejercicio 49

Añade las instrucciones oportunas al programa anterior para que te pregunte si quieres conocer el valor de alguna de las variables del array, y, en caso afirmativo, te pregunte la fila y la columna, imprimiendo el contenido de esa variable en la pantalla.

Ejercicio 50

Carga el programa del ejercicio 48, y, sin modificar el valor de las variables, adapta el listado de forma que, automáticamente, se valoren las variables del array DIM W (2,5) con la condición de que los sucesivos valores del array A se multipliquen por 2 si son mayores que 50 y se eleven al cuadrado si son menores y se transfieran, con esta condición, al array W.

Ejercicio 45

DIM A (10)

Ejercicio 46

10 DIM A (10)

20 FOR X = 1 TO 10

30 INPUT A (X)

40 NEXT X

Ejercicio 47

DIM A (2,5)

Ejercicio 48

10 DIM A (2,5)

20 FOR X = 1 TO 2

30 FOR Y = 1 TO 5

40 PRINT "VALOR "; X; ", "; Y; " DEL

ARRAY NUMERICO A"

50 INPUT A (X, Y)

60 NEXT Y

70 NEXT X

Ejercicio 49

```
10 DIM A (2,5)
20 FOR X = 1 TO 2
30 FOR Y = 1 TO 5
40 PRINT "VALOR ";X;"",", Y, " DEL
  ARRAY NUMERICO A"
50 INPUT A (X, Y)
60 NEXT Y
70 NEXT X
100 PRINT AT 20, 0; "QUIERES CONOCER
EL VALOR DE ALGUNA VARIABLE?
110 INPUT R$
115 CLS
120 IF R$ = "N" THEN STOP
130 PRINT "FILA?",
140 INPUT F
145 PRINT F
150 PRINT "COLUMNA?",
160 PRINT C
165 PRINT C
170 PRINT A (F.C)
180 GOTO 100
```

Ejercicio 50

5 DIM W (2,5)

10 DIM A (2,5)

20 FOR X = 1 TO 2

30 FOR Y = 1 TO 5

52 IF A (X, Y) <= 50 THEN GOTO 57

53 LET W (X, Y) = A (X, Y) *2

57 LET W (X, Y) = A (X, Y) **2

60 NEXT Y

70 NEXT X

Trabajo:

Referencias:

Hoja número:

NUMERO LINEA	INSTRUCCION	COMENTARIOS

Trabajo:

Referencias:

Hoja número:

NUMERO LINEA	INSTRUCCION	COMENTARIOS

ZX81

Al atravesar "La pequeña gran puerta" habrá entrado en el apasionante mundo de la Informática y estará en condiciones de escribir programas bastante sofisticados, y estudiar cualquier otra versión de BASIC que un nuevo y distinto ordenador al ZX81 pudiera exigirle.

La pretensión de este libro es "hacerle ver" cómo "razona" un computador. Va dirigido a todas aquellas personas que deseen aprender a manejar un microcomputador sin tener conocimientos previos de programación ni de la lengua inglesa.

Ideal para que los padres aprendan con sus hijos a comunicarse con la máquina que más le ayudará en el futuro: EL ORDENADOR.



ISBN: 84-283-1277-X