

8.- En una tienda, los vendedores trabajan por sueldo base más una comisión.
 10.- Dados dos números, si ambos son mayores que 10, que imprima la suma de ellos; si uno de ellos no es mayor que 10 que imprima el producto.
 base a la venta realizada por semana.

UNIDAD IV

PROGRAMACIÓN BASIC

1.- INTRODUCCIÓN

A.- CONCEPTOS DE BASIC



En este capítulo de nuestro texto, emplearemos exclusivamente el lenguaje BASIC. Fundamental por la que decidimos usar este lenguaje es su sencillez en su manejo, ya que las instrucciones son palabras técnicas del Inglés, que permite al alumno un aprendizaje inmediato.



BASIC fue desarrollado en 1963 por John Kemeny y Thomas Kurtz en Estados Unidos, su característica principal es la facilidad que ofrece al programador.

Las siglas de BASIC significan:

- Beginner's (de principiantes)
- All-purpose (todo propósito)
- Symbolic (simbólico)
- Instruction (instrucción)
- Code (código)



PROGRAMACIÓN BASIC

10.- Dados dos números, si ambos son mayores que 10, que imprima la suma de ellos; si uno de ellos no es mayor que 10 que imprima el producto.



PROGRAMACIÓN BASIC

UNIDAD IV

PROGRAMACIÓN BASIC

1.- INTRODUCCIÓN.

A.- CONCEPTOS DE BASIC

En este cuarto capítulo de nuestro texto, emplearemos exclusivamente el lenguaje BASIC. La razón fundamental por la que decidimos utilizar este tipo de lenguaje es su operatividad, pues facilita la comunicación entre hombre-máquina, debido a la sencillez de su manejo, ya que las instrucciones se dan en palabras técnicas del inglés, que permite al alumno un acercamiento inmediato.

BASIC fue desarrollado en 1963 por John Kemeny y Thomas Kurtz en Estados Unidos, su característica principal es la facilidad que ofrece al programar.

Las siglas de **BASIC** significan:

- B**eginner's (de principiante)
- A**ll-purpose (todo propósito)
- S**ymbolic (simbólico)
- I**nstruction (instrucción)
- C**ode (código)

Traducida resultaría: Código de instrucciones simbólicas para todos los propósitos para principiantes.

2.- ESTRUCTURA DE UN PROGRAMA

Un programa en BASIC está formado por:

- A.- Número de línea.
- B.- Instrucciones.
- C.- Documentación del programa.
- D.- Uso de constantes.
- E.- Asignación de variables.
- F.- Terminación de línea.

A.- NÚMERO DE LÍNEA.

Son números naturales usados como referencia para la secuencia del programa, generalmente marcados de diez en diez para dejar la opción de intercalar instrucciones posteriores, sirven además como etiqueta informativa al hacer transferencia de control.

Si después de haber realizar el programa se requiere introducir otra instrucción **12 instrucción**, se puede teclear al final del programa y BASIC lo reacomodará de acuerdo a su número de línea.

Ejemplo	Ejemplo modificado
10 instrucción	10 instrucción
20 instrucción	12 instrucción
30 instrucción	20 instrucción
40 instrucción	30 instrucción
... ..	40 instrucción

B.- INSTRUCCIONES.

Las instrucciones para realizar un programa en lenguaje BASIC pueden ser de dos tipos: **Operaciones y Estatutos.**

OPERACIONES.- Pueden ser **Aritméticas y Lógicas.**

Operaciones aritméticas.- Parte del programa donde se evalúan todas las expresiones matemáticas que requieren una operación. Los operadores aritméticos son:

Operador	Operación
+	suma
-	resta
*	multiplicación
/	división
\	división entera
^	exponenciación
MOD	el residuo de la división

La división entera se representa por la diagonal invertida \, al contrario de la barra inclinada / que representa la división de dos números. Esta operación convierte al dividendo y divisor en enteros, redondeándolos si fuera necesario, antes de efectuar la división propiamente dicha y, tras su ejecución, el cociente se redondea de igual manera.

Ejemplos: $323.89 \setminus 9 = 36$
 $427.70 \setminus 52 = 8$

2.- El operador **MOD** proporciona el residuo de una división, pero antes de realizar la división, tanto el dividendo como el divisor se redondean como en la división entera.

Formato : Dividendo MOD Divisor

- Ejemplos:
- 7 MOD 4 La respuesta es: 3
 - 18.6 MOD 3 La respuesta es: 1
 - 75 MOD 8.9 La respuesta es: 3
 - 8 MOD 2 La respuesta es: 0
 - 750 MOD 25 La respuesta es: 0

Los operadores aritméticos son ejecutados bajo la siguiente jerarquía:

Jerarquía	Operación
1 ^a . ^	exponenciación
2 ^a . *	multiplicación
/	división
\	división entera
MOD	residuo de la división
3 ^a . + -	suma y resta

Los operadores de igual jerarquía se evalúan de izquierda a derecha. En una expresión que incluya dos o más operadores aritméticos, los de jerarquía más alta se ejecutan antes de que los de menor jerarquía. Este orden de evaluación se puede anular por el uso de **paréntesis**, de modo que cualquier operador dentro de ellos se ejecuta primero.

Si la expresión tiene paréntesis, las operaciones contenidas en ella se calcula en primer lugar; por lo tanto, se puede cambiar la secuencia de evaluación o el valor de la expresión aritmética, encerrando parte de las expresiones en paréntesis.

- Ejemplos:
- a) $2 + 5 * 3 + 7 - 12 / 4 + 3 * 3 ^ 2$
 $2 + 5 * 3 + 7 - 12 / 4 + 3 * 9$
 $2 + 15 + 7 - 3 + 27$
48
 - b) $(2 + 5) * 3 + (7 - 12) / 4 + (3 * 3) ^ 2$
 $7 * 3 - 5 / 4 + 9 ^ 2$
 $7 * 3 - 5 / 4 + 81$
 $21 - 1.25 + 81$
100.75

Nota: La existencia de paréntesis anidados implica iniciar la solución de los más internos hacia fuera.

Operadores Lógicos .- Son los relacionales y los booleanos.

Los operadores relacionales son:

=	Igual	>=	ó	=>	Mayor o igual que
>	Mayor que	<=	ó	<	Menor o igual que
<	Menor que	<>	ó	><	Diferente

- Ejemplos:
- A < 100
 - NUM >= 70
 - AB1 <> 500

Los operadores booleanos son:

NOT	complemento lógico
AND	conjunción, "Y"
OR	disyunción, "O"

Ejemplos: A > 100 AND NUM >= 70
AB1 <> 500 OR CD2 <= 800

Los operadores lógicos, tanto los relacionales como los booleanos, tienen igual prioridad; si varios de ellos están presentes en la misma expresión, se evalúan de izquierda a derecha. En el caso de que en una expresión existan operadores aritméticos y lógicos, los operadores lógicos se ejecutan después de los aritméticos.

ESTATUTOS. - Son palabras reservadas en el lenguaje BASIC que indican una orden específica. Los estatutos o sentencias los mencionaremos más adelante, daremos su sintaxis y las formas de como utilizarlos.

Ejemplos de estatutos: REM, PRINT, INPUT, LET, CLS, GOTO, etc.

C.- DOCUMENTACIÓN DEL PROGRAMA.

Este aspecto no es ejecutado por la computadora; sólo permite al programador introducir comentarios al programa tales como: especificar objetivos, quien lo elaboró, inicio de un proceso determinado, etc.

Ejemplo: **REM** Este programa determina como aprobar el curso.

D.- USO DE CONSTANTES.

Las constantes, como ya lo tratamos en el capítulo anterior, son valores declarados en los programas y que no cambian durante el proceso. Existen dos tipos: **Alfanuméricas** y **Numéricas**.

Ejemplo: DOMS = "Secretaría Académica"

Constantes alfanuméricas o strings. - En el lenguaje BASIC éstas constantes pueden contener hasta 255 caracteres: alfabéticos, combinación de letras y números o números solamente. Deberán encerrarse entre comillas y no es posible realizar operaciones aritméticas o cálculos con ellas.

Ejemplos: "UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN"

"Col. Roble."

"Madero 2315 Pte."

"Sierra de Santa. Clara"

Constantes numéricas. - Representan valores numéricos que pueden ser negativas o positivas; a las constantes numéricas positivas, se les puede suprimir el signo, pero la negativas siempre deben especificarse.

Ejemplos: 10 - 60.25
- 22 37.12
1563 - 53.295
- 135 874.395

E.- ASIGNACIÓN DE VARIABLES.

Nombre simbólico que permite almacenar información en un espacio de la memoria, la cual puede representar valores alfanuméricos o numéricos.

Las reglas para la formación de variables son:

- ▣ Longitud máxima de 40 caracteres.
- ▣ Siempre debe empezar con un carácter alfabético.
- ▣ Sólo puede almacenar valores de su mismo tipo.
- ▣ No pueden emplearse palabras reservadas como: RUN, TO, IF, etc.

En el lenguaje BASIC, utilizaremos tres tipos de variables: **Reales, Enteras y Alfanuméricas (Strings o Cadenas)**

▣ **Variables reales.**- Su valor puede contener valores enteros y decimales.

Ejemplos: B = -587.734
DOM = 41416794456
AB2 = 9

▣ **Variables enteras.**- Son aquellas que utilizamos para guardar números enteros y deben ir acompañadas del signo %.

Ejemplos: B% = -699
N% = 12
Ejemplo: R% = 167

▣ **Variables alfanuméricas o strings.**- Son todas aquellas variables que contengan una cadena de caracteres, pueden ser letras, números o ambos; aunque contengan un número, éste no representa un valor numérico. Las variables strings deberán acompañarse al final con un signo \$ y su valor tendrá que encerrarse entre comillas.

Ejemplo: DOM\$ = "Secretaría Académica"

A\$ = "34"

B\$ = "50"

SUM\$ = A\$ + B\$

En el último ejemplo, una variable strings puede estar en función de otra variable alfanumérica; y tal como se mencionó en el párrafo anterior, el valor de SUM\$ sería igual a "3450". El espacio en blanco dentro de las comillas también se toma como un carácter.

Nota: BASIC respeta todo aquello escrito entre comillas, dejándolo tal como se escribió.

F.- TERMINACIÓN DE LÍNEA.

En este lenguaje todos los estatutos, órdenes y nombres de los programas pueden escribirse con mayúsculas o minúsculas, pero BASIC los cambia a mayúsculas. Al realizar una corrección y al finalizar una línea del programa se debe pulsar Intro, Return o Enter.