

RESUMEN PROGRAMACIÓN Q-BASIC

TEMA 2: INSTRUCCIONES GRÁFICAS Y OTRAS

2.1.- RESOLUCIÓN DE LA PANTALLA → Los modos de pantalla más utilizados son:

SCREEN 12: Tiene una resolución de 640 x 480 píxeles. Admite 16 colores elegidos de una paleta de 256.

SCREEN 13: Tiene una resolución de 320 x 200 píxeles. Admite 256 colores.

2.2- COLORES

0 → Negro

1 → Azul

2 → Verde

3 → Cian

4 → Rojo

5 → Magenta

6 → Amarillo

7 → Blanco mate

8 → Gris

9 → Azul claro

10 → Verde claro

11 → Cian claro

12 → Rojo claro

13 → Magenta claro

14 → Amarillo claro

15 → Blanco brillante

2.3.- PUNTO → Sintaxis: **PSET (X, Y), número del color**

2.4.- LÍNEA → Sintaxis: **LINE (X1, Y1) – (X2, Y2), número del color**

2.5.- RECTÁNGULO → Sintaxis: **LINE (X1, Y1) – (X2, Y2), número del color, B**

2.6.- RECTÁNGULO RELLENO DE COLOR

Sintaxis: **LINE (X1, Y1) – (X2, Y2), número del color, BF**

2.7.- CÍRCULO → Sintaxis: **CIRCLE (X, Y), valor del radio, número del color del perímetro**

2.8.- CAMBIO DEL COLOR DE FONDO DE LA PANTALLA

Sintaxis: **PAINT (X, Y), color de relleno**

2.9.- COLOREADO DE UNA ZONA

Sintaxis: **PAINT (X, Y), color de relleno, color del límite o borde**

2.10.- MULTICOMANDO DE DIBUJO → Sintaxis: **DRAW “{parámetros}”**

Parámetros:

Cx → Color x, indica que se va a empezar a dibujar con el color x.

Dx → Down x, dibuja x píxeles hacia abajo

Ux → Up x, dibuja x píxeles hacia arriba

Lx → Left x, dibuja x píxeles hacia la izquierda

Rx → Right x, dibuja x píxeles hacia la derecha

Ex → dibuja x píxeles hacia arriba-derecha

Fx → dibuja x píxeles hacia abajo-derecha

Gx → dibuja x píxeles hacia abajo-izquierda

Hx → dibuja x píxeles hacia arriba-izquierda

M x,y → Move x,y , se mueve al punto x, y dibujando píxeles.

Hace una línea desde donde esté al punto x, y

BM x,y → se mueve al punto x, y sin dibujar.

Ax → Gira una figura hecha con DRAW:

- ♦ x = 1 → gira 90°
- ♦ x = 2 → gira 180°
- ♦ x = 3 → gira 270°

Ejemplo: DRAW “ C10 BM 150,140 D20 L30 U40 R60”

2.11.- INSTRUCCIONES VARIAS

- **PAUSA** → Sintaxis: **SLEEP t**
- **BORRADO DE PANTALLA** → Sintaxis: **CLS**
- **INTRODUCCIÓN DE INFORMACIÓN SOBRE EL PROGRAMA**
Sintaxis: **REM: “ Información que deseemos introducir ”**
- **SONIDO** → Sintaxis: **BEEP**
- **DETECCIÓN DE PULSACIÓN SOBRE EL TECLADO**
Sintaxis: **a\$ = INKEY \$** → Detecta si se pulsa una tecla y la almacena en la variable a\$
- **CLEAR** → Sintaxis: **CLEAR** Hace que todas las variables de un programa queden a cero.
- **IMPRESIÓN EN PANTALLA** → Sintaxis: **PRINT “ texto que se desea imprimir ”**

TEMA 3: BUCLES

3.1.- BUCLES CON CONTADOR: CICLOS FOR / NEXT

3.1.1.- BUCLES CRECIENTES

Sintaxis:

FOR i = inicio TO fin

Instrucciones que
queremos repetir

NEXT i

También podemos poner:

FOR i % = inicio TO fin

Instrucciones que
queremos repetir

NEXT i

3.1.2.- BUCLES DECRECIENTES

Sintaxis:

FOR I = fin TO inicio STEP -1

Instrucciones que queremos repetir

NEXT I

3.2.- BUCLES CONDICIONALES

3.2.1.- Comandos DO, LOOP, UNTIL {condición}

Sintaxis:

DO

Instrucciones que queremos repetir

LOOP UNTIL “Condición”

Ejemplo: LOOP UNTIL a\$ <> “ ”

También podemos poner:

DO UNTIL “Condición”

Instrucciones que queremos repetir

LOOP

Ejemplo: DO UNTIL a\$ = “A ”

3.2.2.- Comandos WILE {condición}, WEND

WHILE {condición}

Instrucciones que queremos repetir

WEND

La condición ha de ser una expresión numérica → Ejemplo: WHILE i < 10

3.2.3.- Comandos DO, LOOP, WILE {condición}

El comando WHILE se puede utilizar tanto con el comando DO (DO WHILE) como con el comando LOOP (LOOP WHILE)

La sintaxis puede ser de 2 formas:

1ª Forma:

DO WHILE {condición}

Instrucciones
que queremos
repetir

LOOP

Ejemplo : DO WHILE i < 10

2ª Forma:

DO

Instrucciones
que queremos
repetir

LOOP WHILE {condición }

Ejemplo : LOOP WHILE i < 10

TEMA 4: IMPRESIÓN DE TEXTOS EN PANTALLA

4.1.- BORRADO DE PANTALLA → Sintaxis: CLS

4.2.- IMPRESIÓN EN PANTALLA → Sintaxis: PRINT “ texto que se desea imprimir ”

Ejemplo: PRINT “Hola, Ø”; nombre\$; “Øtienes Ø”; edad; “Øaños”

4.3.- FILA Y COLUMNA DEL COMIENZO DE IMPRESIÓN

Sintaxis: LOCATE fila, columna

Valores máximos	FILA	COLUMNA
SCREEN 12	29	75
SCREEN 13	23	40

4.4.- IMPRESIÓN EN PANTALLA DE LA FECHA ACTUAL

Sintaxis: **PRINT DATES\$** ; Ejemplo: PRINT “ La fecha de hoy es “ ; DATES\$

4.5.- IMPRESIÓN EN PANTALLA DE LA HORA ACTUAL

Sintaxis: **PRINT TIMES\$** ; Ejemplo: PRINT “ La hora actual es “ ; TIMES\$

4.6.- COLORES DE LOS CARACTERES Y FONDO DE PANTALLA

Sintaxis: **COLOR , número color letra, número color fondo**

Por omisión toma como color de fondo el negro

TEMA 5: VARIABLES E INTRODUCCIÓN DE DATOS

5.1.- NOMBRE DE LAS VARIABLES

El nombre de las variables es libre , pero siempre conviene hacerlo descriptivo.

Las variables pueden tener cualquier forma y tamaño respetando las siguientes reglas:

- 1) El primer carácter debe ser siempre una letra: a, b, c.....A, B, C,....., no importa si el nombre está en mayúsculas o minúsculas.
- 2) Después de la primera letra podemos poner letras , números o subrayados (_)
- 3) El carácter final:
Para variable numéricas → puede ser : %, ! , #, δ, o nada.
Para variables de cadena → ha de ser : \$
- 4) No puede ser una palabra reservada para un comando de Q Basic, por ejemplo PRINT.

5.2.- VARIABLES DE CADENA

Una cadena es una forma de guardar información en Q BASIC

El nombre de la variables de cadena es libre , pero SIEMPRE el símbolo \$ debe ir al final del nombre de dicha variable. El contenido de la variable de cadena debe de ir entre comillas.

Ejemplos: **nombre1\$** = “ Mª Dolores Villalobos ” ; **nombreuno\$** = “Daniel Ventura ”

Las variables de cadena pueden tener caracteres numéricos, pero estos no tienen valor numérico.

Ejemplo: nombre1\$ = “3215” → 3215 no tiene valor numérico

5.3.- PARTE ENTERA DE UN NÚMERO

Sintaxis: **INT { Variable numérica }** ; Ejemplo: B = INT A

5.4.- PARTE DECIMAL DE UN NÚMERO

Sintaxis: **FRAC { Variable numérica }** ; Ejemplo: B = FRAC A

5.5.- GENERACIÓN DE NÚMEROS ALEATORIOS

- **Comando RND** → genera un número cuasialeatorio con un valor comprendido entre cero y uno.; Ejemplo: X = RND

- **Comando RANDOMIZE** (es más utilizada la siguiente)

La instrucción RANDOMIZE debe preceder a la instrucción RND

- **Comando RANDOMIZE TIMER**

Si la instrucción RANDOMIZE TIMER precede a la instrucción RND se generarán números totalmente aleatorios. Ejemplo;

```
RANDOMIZER TIMER
x = INT (3 + 5 * RND)
```

5.6.- INTRODUCCIÓN DE DATOS MEDIANTE EL TECLADO

Sintaxis: **INPUT ; { nombre de la variable numérica o de cadena }**

Sintaxis: **INPUT “ Texto deseado” ; { nombre de la variable numérica o de cadena }**

Otra forma de utilizar el comando INPUT con las variables de cadena es la siguiente:

Sintaxis: **Variable\$ = INPUT\$ (cantidad de teclas a recibir)**

Ejemplo: nombre\$ = INPUT\$ (6)

El llegar a esta línea el programa se detiene a la espera de que introduzcamos la información. El (6) indica el número de teclas que debemos pulsar

TEMA 6: TOMA DE DECISIONES

6.1.- NOTACIONES

es igual a (=) ; no es igual a (<>) ; es mayor que (>) ; es mayor o igual que (>=)
es menor que (<) ; es menor o igual que (<=).

6.2.- COMANDOS PARA LA TOMA DE DECISIONES

Comandos: IF.....THEN.....ELSE.....END IF

Sintaxis: **IF { condición } THEN { ejecuta la próxima sentencia } ELSE END IF**

- ♦ IF → si, se cumple una condición
- ♦ THEN → entonces ejecuta la próxima sentencia
Si no se cumple la condición continuara la ejecución del programa desde ELSE o desde END IF
- ♦ ELSE → en caso contrario (todo lo demás).
- ♦ END IF → fin de la toma de decisión

Primera forma:

```
CLS
REM "Mayoría de edad"
INPUT “¿Cuántos años tienes?:”
, edad
IF edad < 18 THEN
PRINT “Eres menor de edad”
ELSE
PRINT “Eres mayor de edad”
END IF
```

Segunda forma:

```
CLS
INPUT “¿Cuántos años tienes?:”
, edad
IF edad < 18 THEN
PRINT “Eres menor de edad”
IF edad > 18 THEN
PRINT “Eres mayor de edad”
END IF
```

TEMA 7: SALTOS

I.- SALTOS INCONDICIONALES

1.- COMANDO “GOTO” → Salta incondicionalmente al destino de salto especificado.

Sintaxis: **GOTO { Etiqueta de destino o N° de línea de destino }** ; Ejemplo: GOTO 40

2.- SUBROUTINAS

Una subrutina es una parte del programa que se ha de repetir en muchas ocasiones. Se colocan después del fin (END) del programa principal y se accede a ellos mediante un comando GOSUB.

Las subrutinas:

- ♦ Necesitan de una etiqueta (nombre) que las identifique (ejemplo: “PALOMA”, “450”, etc.)
- ♦ Finalizan siempre con el comando RETURN

3.- GOSUB → Sintaxis: **GOSUB { etiqueta }**

4.- RETURN → Sintaxis: **RETURN**

El comando RETURN retorna la ejecución del programa desde una subrutina al programa principal

II.- SALTOS CONDINCIONALES

1.- Comandos IF / THEN / GOTO / GOSUB

Ejemplo: IF A = 20 THEN GOTO 35

Ejemplo: IF A > 33 THEN GOTO 20

Ejemplo: IF A < 56 THEN GOTO 20

Ejemplo: IF A >= 61 THEN GOTO 100

Ejemplo: IF A\$ = “Paloma” THEN GOTO 35

Ejemplo: IF A = 20 THEN GOSUB Canario

Ejemplo: IF A > 33 THEN GOSUB Telefono

Ejemplo: IF A < 56 THEN GOSUB 20

Ejemplo: IF A >= 61 THEN GOSUB Andres

Ejemplo: IF A\$ = “Paloma” THEN GOSUB 35

2.- BIFURCACIÓN MÚLTIPLE

1) Comando ON-GOTO

Sintaxis: **ON { Variable } GOTO , dato1, dato2, dato3,**

Ejemplo: ON A GOTO 15, 40, 25, 40, 60

Si la variable A = 1 entonces el control se transfiere a la etiqueta 15, si A = 2 a la etiqueta 40, si A = 3 a la etiqueta 25, etc.

2) Comando ON-GOSUB

Sintaxis: **ON { Variable } GOSUB , dato1, dato2, dato3,**

Ejemplo 1: ON A GOSUB 15, 40, 25, 40, 60

Si la variable A = 1 entonces el control se transfiere a la etiqueta 15, si A = 2 a la etiqueta 40, si A = 3 a la etiqueta 25, etc.

Ejemplo 2: ON A GOSUB Paloma, Canario, Lechuza, Loro, Águila

Si la variable A = 1 entonces el control se transfiere a la etiqueta Paloma, si A = 2 a la etiqueta Canario, si A = 3 a la etiqueta Lechuza, etc.

3.- SALTO SELECTIVO

Comandos: **SELECT CASE número ; CASE ELSE ; END SELECT**

Este comando nos permite ejecutar una sola sentencia de los múltiples sentencias que puede tener un determinado bloque del programa.

Ejemplo: desarrolla un programa que realice las cuatro operaciones matemáticas básicas

```
CLS ; SCREEN 12
PRINT " MENU PRINCIPAL "
PRINT " 1. SUMA "
PRINT " 2. RESTA "
PRINT " 3 MULTIPLICACIÓN "
PRINT " 4. DIVISIÓN "
REM: aquí comienza el bloque select-case
INPUT " INTRODUCES NÚMERO ( 1 - 4 ): " ; número
SELECT CASE número
CASE 1
PRINT " Has seleccionado el número 1 "
CASE 2
PRINT " Has seleccionado el número 2 "
CASE 3
PRINT " Has seleccionado el número 3 "
CASE 4
PRINT " Has seleccionado el número 4 "
CASE 5
PRINT " Has seleccionado el número 5 "
CASE ELSE
PRINT " El número que has seleccionado no era de 1 a 4 "
END SELECT
```

Previamente al comando **SELECT CASE** se nos pregunta, mediante el comando **INPUT**, por el valor de una variable llamada "número".

A continuación de " **SELECT CASE número** " viene un bloque constituido por **CASE 1**, **CASE 2**, **CASE 3**, **CASE 4** y **CASE 5**. El programa ejecutará una sola línea de este bloque en función del número introducido. Si introducimos, por ejemplo un 2, el programa saltará a la línea **CASE 2** y ejecutará todas las líneas que están a continuación antes de que aparezca **CASE 3**. Cuando alcance **CASE 3** el programa saltará a **END SELECT**

CASE ELSE solamente es leído en el caso de que no introduzcamos un número comprendido entre 1 y 4.

Ejemplo:

```
CLS
INPUT " Introduce número de 1 a 8 " ; número
SELECT CASE número
CASE 1 TO 2
PRINT " Has seleccionado el número 1 ó 2 "
CASE 3 TO 4
PRINT " Has seleccionado el número 3 ó 4 "
CASE 5 TO 6
PRINT " Has seleccionado el número 5 ó 6 "
CASE 7 TO 8
PRINT " Has seleccionado el número 7 u 8 "
END SELECT
```

El nuevo comando **TO** que hemos agregado nos permite introducir un rango dentro de la sentencia **SELECT CASE**.