

**Nombre y Apellidos:**  
**Especialidad y Grupo:**

## **FUNDAMENTOS DE INFORMÁTICA**

**9-febrero-2001**

### **PROGRAMA de Estructuras repetitivas y alternativas (2.5 puntos)**

1. Los números de Fibonacci constituyen una serie en la que cada número es igual a la suma de los 2 anteriores, es decir:  $f_0 = 0$ ,  $f_1 = 1$ ,  $f_n = f_{n-2} + f_{n-1}$  ( $n \geq 2$ ). Visualizar aquellos números correspondientes a dicha serie que pertenezcan al intervalo  $[a, b]$ . Para ello:

- Teclar los valores enteros de **a** y **b**, verificando que no sean negativos ni que **a** sea mayor o igual a **b**. En caso de error, emitir el siguiente mensaje:

\*\*\* Los límites teclados a y b no son válidos \*\*\*

y repetir su entrada hasta que dichos límites sean correctos.

- Mostrar en pantalla por columnas solamente los números que cumplan la regla anterior y que además estén comprendidos dentro de  $[a, b]$ .

- Por ejemplo, si el intervalo es  $[0, 1]$ , el resultado en la pantalla muestra:

0                    1                    1

- Otro ejemplo, si el intervalo es  $[1500, 40000]$ , en la pantalla se visualiza:

1597                    2584                    4181                    6765                    10946

17711                    28657

```
REM Visualizar por pantalla los números de Fibonacci
REM comprendidos dentro de los límites del intervalo [a,b].
REM
```

```
DIM a AS LONG, b AS LONG
DIM fibn1 AS LONG
DIM fibn2 AS LONG
DIM fibn AS LONG
CLS
```

```
DO
  PRINT "Teclea los valores de los límites de [a,b]";
  INPUT ": ", a, b
  IF a < 0 OR b < 0 OR a >= b THEN
    PRINT "*** Los límites teclados a y b no son válidos ***"
  END IF
LOOP WHILE a < 0 OR b < 0 OR a >= b
```

```
fibn2 = 0
fibn1 = 1
IF fibn2 >= a AND fibn1 <= b THEN
  PRINT fibn2, fibn1,
END IF
fibn = fibn2 + fibn1
DO
  IF fibn >= a AND fibn <= b THEN
    PRINT fibn,
  END IF
  fibn2 = fibn1
  fibn1 = fibn
  fibn = fibn2 + fibn1
LOOP WHILE fibn <= b
```

**PROGRAMA de Arrays y Registros (3.5 puntos)**

2. La pescadería “PescaExpress” se diferencia de otras en que puede recibir pedidos a través de Internet y además los sirve al domicilio particular del cliente. Junto con el pedido debe ir la factura, que se va a obtener mediante un programa. Se pide:

- a. Escribir el diagrama de Warnier del apartado correspondiente a la carga del array de registro *pedidoCliente*.
- b. Codificar dicho programa en QB para emitir la factura que debe pagar el cliente.

Nuestra pescadería vende 40 clases diferentes de pescados. Para realizar el programa se dispone de los siguientes arrays de registro.

- Lista, llamada *clasesPescado*, de 40 elementos. Cada uno de ellos posee la estructura del siguiente registro:

registroPescado

nombre	precioKg	tipo
0	1920	21 ← 22 →

- Lista dinámica, llamada *pedidoCliente*, que contiene la información de los diferentes pescados solicitados. Cada elemento de este array de registro tiene la siguiente estructura:

registroPedido

nombre	Peso	importeIVAIncluido
0	1920	2324 27

- Suponer que todos los datos relativos a los pescados ya están almacenados en *clasesPescado*, incluido el campo tipo, que contiene la letra A si se trata de pescado catalogado como Azul o la letra B si es Blanco.
- Cargar la lista dinámica *pedidoCliente*, con los datos correspondientes a cada uno de los pescados comprados por un cliente, de la siguiente forma:
  - Preguntar al cliente:  
¿Cuántas clases de pescados vas a pedir? \_
  - Dimensionar ahora la lista dinámica teniendo en cuenta la respuesta anterior.
  - Para cada elemento de la lista:
    - Teclar el nombre del pescado y buscarlo en la lista *clasesPescado*.
      - Si no está, visualizar un mensaje de error y repetir su entrada.  
El pescado xxxxxxxxxxxxxx no se vende en PESCAEXPRESS.
      - Si sí está, teclar el peso en Kg. y comprobar si se trata de un pescado azul o blanco, pues el IVA que se debe aplicar al importe es del 20% en el primer caso y del 16% en el otro. Guardar la información perteneciente a dicho pescado en la lista *pedidoCliente*.
- Elaborar la factura según este formato:

**Pescadería PESCAEXPRESS**

0.500 Kg. de lubina → 4,050  
 0.625 Kg. de merluza → 2,700  
 #.### Kg. de xxxxxxxxxxxxxx → ###,###

.....  
 -----  
 Total Factura..... → #,###,### pts.

```

REM Visualizar por pantalla la factura de un cliente
REM que compre en PESCAEXPRESS.
REM
REM El programa se hace sólo para 5 clases de pescado.
REM
CLS

TYPE registroPescado
    nombre AS STRING * 20
    precioKg AS INTEGER
    tipo AS STRING * 1
END TYPE

DIM clasesPescado(1 TO 5) AS registroPescado
DIM i AS INTEGER, j AS INTEGER

TYPE registroPedido
    nombre AS STRING * 20
    peso AS SINGLE
    importeIVAIncluido AS SINGLE
END TYPE

'Carga del array de registro clasesPescado
DATA pescadilla, 900, B, anchoa, 700, A
DATA merluza, 2900, B, chicharro, 950, A
DATA perlita, 500, B

FOR i = 1 TO 5
    READ clasesPescado(i).nombre
    READ clasesPescado(i).precioKg
    READ clasesPescado(i).tipo
NEXT i

DIM n AS INTEGER
PRINT " ¿Cuántas clases de pescados vas a pedir";
INPUT n

DIM pedidoCliente(1 TO n) AS registroPedido
DIM importe AS SINGLE

'Carga del array de registro pedidoCliente
FOR i = 1 TO n
    DO
        INPUT " Nombre del pescado: ", pedidoCliente(i).nombre
        pedidoCliente(i).nombre = LCASE$(pedidoCliente(i).nombre)
        j = 0
        DO
            j = j + 1
        LOOP UNTIL pedidoCliente(i).nombre = clasesPescado(j).nombre
    OR j = 5
        IF pedidoCliente(i).nombre = clasesPescado(j).nombre THEN
            INPUT " Peso: ", pedidoCliente(i).peso
            importe = clasesPescado(j).precioKg *
pedidoCliente(i).peso
            IF clasesPescado(j).tipo = "A" THEN
                pedidoCliente(i).importeIVAIncluido = importe +
importe * 20 / 100
            ELSE
                pedidoCliente(i).importeIVAIncluido = importe +
importe * 16 / 100
            END IF
        ELSE
            PRINT " El pescado ";
            COLOR 14: PRINT RTRIM$(pedidoCliente(i).nombre);

```

```

        COLOR 7: PRINT " no se vende en PESCAEXPRESS."
    END IF
    LOOP UNTIL pedidoCliente(i).nombre = clasesPescado(j).nombre
NEXT i

'Emisión de la factura correspondiente a un cliente
DIM total AS SINGLE
total = 0
PRINT
PRINT "Pescadería PESCAEXPRESS"
PRINT STRING$(23, "-")
FOR i = 1 TO n
    PRINT SPC(10);
    PRINT USING "#.### Kg. de "; pedidoCliente(i).peso;
    PRINT pedidoCliente(i).nombre;
    PRINT " --> ";
    PRINT USING "###,###"; pedidoCliente(i).importeIVAIncluido
    total = total + pedidoCliente(i).importeIVAIncluido
NEXT i
PRINT SPC(10);
PRINT STRING$(51, "-")
PRINT SPC(10); "Total factura.....-->";
PRINT USING "#,###,### pts."; total

```

**CUESTIÓN de Funciones (1.5 puntos)**

3. Suponiendo que el usuario teclea los datos 1,2 , indicar (explicando razonadamente) los sucesivos valores de la variable total que se van a visualizar en pantalla tras la ejecución del siguiente programa XY.bas.

```
DECLARE FUNCTION ope$ (a AS STRING, b AS STRING)
DIM x AS STRING, y AS STRING
DIM i AS INTEGER
DIM total AS STRING
```

```
INPUT "Teclea valores numéricos para x,y: ", x, y
FOR i = 1 TO 3
    total = total + ope$(x, y)
    PRINT "Total"; i; " = "; total
NEXT i
```

```
FUNCTION ope$ (a AS STRING, b AS STRING)
    ope$ = a + b
    a = a + a
END FUNCTION
```

El paso de los argumentos del PP a los parámetros formales de la función se hace por referencia. Como el parámetro formal "a" de la función se modifica en su interior, este cambio afecta a su correspondiente argumento en el PP, que es "x". La siguiente vez que la función es llamada, el valor del argumento "x" no es el mismo que se había tecleado en un principio. Por lo tanto, el resultado es el siguiente:

```
Total 1 = 12
Total 2 = 12112
Total 3 = 1211211112
```

### **PROGRAMA de Procedimientos (2.5 puntos)**

4. El procesador WORD dispone de una herramienta para contar el número de palabras que hay en un fichero de texto. A partir de esta idea, vamos a codificar una utilidad que contabilice y visualice las palabras contenidas en una frase. Para ello, escribir en lenguaje QB:
- Todas las sentencias que faltan en el Programa Principal, para que:
    - La frase introducida (máximo de 100 palabras) se considere correcta:
      - Debe estar en mayúsculas.
      - Debe terminar en ".".
    - Después de la llamada al procedimiento, se visualicen únicamente los siguientes datos:
      - Primero, el número exacto de palabras.
      - Luego, cuáles son dichas palabras (una en cada línea y la última sin incluir el ".").
  - Los parámetros formales, así como las instrucciones, correspondientes al Procedimiento cargarPalabras, para que:
    - Cada palabra de la frase se guarde en el vector.
    - Se contabilice el número total de palabras o elementos cargados del array.

REM Programa Principal

```
DECLARE SUB cargarPalabras (texto AS STRING, vectorP() AS STRING, numP  
AS INTEGER)
```

CLS

```
DIM frase AS STRING  
DIM vectorPalabras(1 TO 100) AS STRING  
DIM i AS INTEGER
```

```
FOR i = 1 TO 100  
    vectorPalabras(i) = "-----"  
NEXT i
```

```
DIM numPalabras AS INTEGER  
DIM ultimaPosicion AS STRING * 1
```

```
DO  
    PRINT "Teclear frase ";  
    INPUT "(en mayúsculas y terminada en punto): ", frase  
    frase = UCASE$(frase)  
    ultimaPosicion = MID$(frase, LEN(frase), 1)  
LOOP UNTIL ultimaPosicion = "."
```

```
CALL cargarPalabras(frase, vectorPalabras(), numPalabras)
```

```
PRINT "El número de palabras de la frase: "  
COLOR 10 : PRINT frase : COLOR 7  
PRINT "es "; numPalabras; " y son:"  
FOR i = 1 TO numPalabras  
    PRINT vectorPalabras(i)  
NEXT i
```

```
SUB cargarPalabras (texto AS STRING, vectorP() AS STRING, numP AS INTEGER)
  DIM j AS INTEGER, k AS INTEGER
  DIM letra AS STRING, palabra AS STRING
  FOR j = 1 TO LEN(texto)
    letra = MID$(texto, j, 1)
    IF letra <> " " AND letra <> "." THEN
      palabra = palabra + letra
    ELSE
      k = k + 1
      vectorP(k) = palabra
      palabra = ""
    END IF
  NEXT j
  numP = k
END SUB
```