

Nombre y Apellidos:
Especialidad y Grupo:

FUNDAMENTOS DE INFORMÁTICA

20-junio-2003

PROGRAMA de Estructuras repetitivas y alternativas (2.5 puntos)

1. Dado un mes del año y el día de la semana en que comienza, escribir un programa en QB que muestre por pantalla la representación del calendario correspondiente a dicho mes. Para ello:

- Introducir los datos del número del mes y del día de la semana. Repetir su entrada hasta que sean correctos, visualizando el mensaje correspondiente en caso de error.
- Por ejemplo, si el mes es 5 y el día de la semana es 4, entonces la representación del mes de mayo debe mostrar en pantalla la siguiente información:

```
L M X J V S D
      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
```

VERSIÓN con bucles FOR

```
CLS
DIM numMes AS INTEGER, diaSemana AS INTEGER
DIM totalDiasMes AS INTEGER
DIM fil AS INTEGER, numFilas AS INTEGER
DIM col AS INTEGER
DIM numeroDia AS INTEGER

DO
    INPUT "Número de mes: ", numMes
    INPUT "Día de la semana: ", diaSemana
    IF numMes < 1 OR numMes > 12 THEN
        PRINT "Mes incorrecto. Repetir"
    END IF
    IF diaSemana < 1 OR diaSemana > 7 THEN
        PRINT "Día de la semana incorrecto. Repetir"
    END IF
LOOP UNTIL numMes >= 1 AND numMes <= 12 AND diaSemana >= 1 AND
diaSemana <= 7

SELECT CASE numMes
    CASE IS = 1, 3, 5, 7, 8, 10, 12
        totalDiasMes = 31
    CASE IS = 2
        totalDiasMes = 28
    CASE IS = 4, 6, 9, 11
        totalDiasMes = 30
END SELECT

PRINT
PRINT " L M X J V S D"
numFilas = (totalDiasMes + diaSemana) \ 7 + 1
numeroDia = 1
```

```

FOR fil = 1 TO numFilas STEP 1
  FOR col = 1 TO 7 STEP 1
    IF col < diaSemana OR numeroDia > totalDiasMes THEN
      PRINT SPC(3);
    ELSE
      PRINT USING "###"; numeroDia;
      numeroDia = numeroDia + 1
    END IF
  NEXT col
  PRINT
  diaSemana = 1
NEXT fil

```

VERSIÓN con bucles DO ... LOOP

```

CLS
DIM numMes AS INTEGER, diaSemana AS INTEGER
DIM totalDiasMes AS INTEGER
DIM fil AS INTEGER, numFilas AS INTEGER
DIM col AS INTEGER
DIM numeroDia AS INTEGER

DO
  INPUT "Número de mes: ", numMes
  INPUT "Día de la semana: ", diaSemana
  IF numMes < 1 OR numMes > 12 THEN
    PRINT "Mes incorrecto. Repetir"
  END IF
  IF diaSemana < 1 OR diaSemana > 7 THEN
    PRINT "Día de la semana incorrecto. Repetir"
  END IF
LOOP UNTIL numMes >= 1 AND numMes <= 12 AND diaSemana >= 1 AND
diaSemana <= 7

SELECT CASE numMes
  CASE IS = 1, 3, 5, 7, 8, 10, 12
    totalDiasMes = 31
  CASE IS = 2
    totalDiasMes = 28
  CASE IS = 4, 6, 9, 11
    totalDiasMes = 30
END SELECT

PRINT
PRINT " L M X J V S D"
LOCATE 5, 1 + (diaSemana - 1) * 3
numeroDia = 1

DO
  DO
    PRINT USING "###"; numeroDia;
    numeroDia = numeroDia + 1
    diaSemana = diaSemana + 1
  LOOP UNTIL numeroDia > totalDiasMes OR diaSemana > 7
  PRINT
  diaSemana = 1
LOOP UNTIL numeroDia > totalDiasMes

```

PROGRAMA de Arrays y Registros (5 puntos)

2. Se trata de realizar un programa en QB para confeccionar una sopa de letras. Para ello, se dispone de los siguientes arrays:

- Array bidimensional dinámico, llamado *arraySopa*, de $m \times m$ elementos de tipo alfanumérico. Cada elemento contiene 1 sola letra (carácter alfanumérico).
- Array unidimensional dinámico, llamado *arrayPalabras*, de n elementos, que contiene la información de las palabras que se van a cargar en el array bidimensional anterior. Cada elemento de este array de registro tiene la siguiente estructura:

registroPalabra

palabra	fila	col	orientación
0	2425	2627	28← 29 →

Para la confección de la sopa de letras se deben seguir estos pasos:

- Teclear la dimensión m y declarar el array bidimensional.
- Teclear la dimensión n y declarar el array unidimensional.
- Cargar la lista *arrayPalabras*, sólo con aquellas palabras que luego se puedan cargar en la tabla *arraySopa*. El formato de pantalla es el siguiente:

Palabra: _____

Fila inicio: ____

Colu inicio: ____

Orientación: ____

La entrada de los cuatro datos anteriores se supone correcta. La orientación puede ser H (Horizontal), V (Vertical) o D (Diagonal). Sólo hay que comprobar que la última letra de cada palabra se pueda cargar después en el array bidimensional, teniendo en cuenta la posición de inicio de la fila, de la columna y su orientación. Si la palabra es demasiado larga, se debe visualizar un mensaje de error y teclear de nuevo los cuatro datos.

- Cargar la tabla *arraySopa* con las letras de las palabras que están almacenadas en la lista *arrayPalabras*, a partir de la posición y según la orientación indicadas.
- Por ejemplo:

arrayPalabras con 2 elementos (válidos para luego cargarlos en la sopa de letras)

GARBANZOS	1	1	D	ARROZ	4	5	H
-----------	---	---	---	-------	---	---	---

arraySopa 10 filas x 10 columnas

G									
	A								
		R							
			B	A	R	R	O	Z	
				A					
					N				
						Z			
							O		
								S	

- Rellenar el resto de los elementos de la tabla anterior con una letra cualquiera, pero de la siguiente forma:
 - cadena = "ZYXWVUTSRQPOÑNMLKJIHGFEDCBA"
 - Generar un número al azar entre [1,27].
 - Este número será la posición que ocupa la letra en la cadena anterior = "ZYXWVUTSRQPOÑNMLKJIHGFEDCBA". Dicha letra es la que se debe cargar en la fila y en la columna que corresponda de la tabla.

'Ejemplo de uso de un array de registro y de
'un array bidimensional alfanumérico

```
CLS
DIM m AS INTEGER
INPUT "Teclea la dimensión del arraySopa: ", m
DIM arraySopa(1 TO m, 1 TO m) AS STRING

TYPE registroPalabra
    palabra AS STRING * 25
    fila AS INTEGER
    colu AS INTEGER
    ori AS STRING * 1
END TYPE
DIM n AS INTEGER
INPUT "Teclea el número de palabras: ", n
DIM arrayPalabras(1 TO n) AS registroPalabra
DIM i AS INTEGER
DIM posiFinalFila AS INTEGER, posiFinalColu AS INTEGER

'Cargar el arrayPalabras con palabras que
'luego puedan entrar en la sopa de letras.

FOR i = 1 TO n
    DO
        INPUT "Palabra: ", arrayPalabras(i).palabra
        INPUT "Fila inicio: ", arrayPalabras(i).fila
        INPUT "Colu inicio: ", arrayPalabras(i).colu
        INPUT "Orientación: ", arrayPalabras(i).ori
        arrayPalabras(i).ori = UCASE$(arrayPalabras(i).ori)
        posiFinalFila = 0
        posiFinalColu = 0
        IF arrayPalabras(i).ori = "H" THEN
            posiFinalFila = arrayPalabras(i).fila
            posiFinalColu = arrayPalabras(i).colu +
LEN(RTRIM$(arrayPalabras(i).palabra)) - 1
        ELSEIF arrayPalabras(i).ori = "V" THEN
            posiFinalFila = arrayPalabras(i).fila +
LEN(RTRIM$(arrayPalabras(i).palabra)) - 1
            posiFinalColu = arrayPalabras(i).colu
        ELSEIF arrayPalabras(i).ori = "D" THEN
            posiFinalFila = arrayPalabras(i).fila +
LEN(RTRIM$(arrayPalabras(i).palabra)) - 1
            posiFinalColu = arrayPalabras(i).colu +
LEN(RTRIM$(arrayPalabras(i).palabra)) - 1
        END IF
        IF posiFinalFila > m OR posiFinalColu > m THEN
            COLOR 5
            PRINT "*** Palabra no entra. Repetir ***"
            COLOR 7
        END IF
    LOOP UNTIL posiFinalFila <= m AND posiFinalColu <= m
NEXT i
```

```

DIM fila AS INTEGER, colu AS INTEGER
DIM longi AS INTEGER
DIM posi AS INTEGER

'Cargar el arraySopa con las letras de las palabras que
'están almacenadas en el arrayPalabras, a partir de
'la posición y según la orientación indicadas.

FOR i = 1 TO n
    fila = arrayPalabras(i).fila
    colu = arrayPalabras(i).colu
    longi = LEN(RTRIM$(arrayPalabras(i).palabra))
    FOR posi = 1 TO longi
        arraySopa(fila, colu) = MID$(arrayPalabras(i).palabra, posi,
1)
        IF arrayPalabras(i).ori = "H" THEN
            colu = colu + 1
        ELSEIF arrayPalabras(i).ori = "V" THEN
            fila = fila + 1
        ELSEIF arrayPalabras(i).ori = "D" THEN
            fila = fila + 1
            colu = colu + 1
        END IF
    NEXT posi
NEXT i

DIM j AS INTEGER

'Rellenar el resto de las posiciones
'del arraySopa con letras al azar

DIM cadena AS STRING * 27
cadena = "ZYXWVUTSRQPONMLKJIHGFEDCBA"
RANDOMIZE TIMER

FOR i = 1 TO m
    FOR j = 1 TO m

        IF arraySopa(i, j) = "" THEN
            num = INT(RND * 27) + 1
            arraySopa(i, j) = MID$(cadena, num, 1)
        END IF
    NEXT j
NEXT i

'Visualizar el contenido del arraySopa

CLS
FOR i = 1 TO m
    FOR j = 1 TO m
        PRINT arraySopa(i, j);
    NEXT j
    PRINT
NEXT i

```

PROGRAMA de Funciones y Procedimientos (2.5 puntos)

3. Visualizar por pantalla el acrónimo de un organismo. Para ello:

- Programa Principal
 - Teclear el nombre del organismo en letras minúsculas.
 - Llamar a la función *enMayusculas\$*, con el fin de que devuelva el nombre del organismo con el primer carácter de cada palabra en mayúsculas, almacenando el resultado en una nueva variable.
 - Utilizar un procedimiento llamado *acronimoOrga*, para que devuelva el acrónimo que corresponda.
 - Visualizar en pantalla en una línea el nombre del organismo y su acrónimo.
Por ejemplo:
Euskal Herriko Unibertsitatea E.H.U.
- Función *enMayusculas\$*
 - El parámetro formal de entrada es el nombre del organismo en minúsculas.
- Procedimiento *acronimoOrga*
 - El parámetro formal de entrada es el nombre del organismo en formato correcto y el de salida es su acrónimo.

```
REM Visualizar por pantalla el acrónimo de un organismo.
```

```
DECLARE FUNCTION enMayusculas$ (orga AS STRING)
```

```
DECLARE SUB acronimoOrga (nombreOrg AS STRING, acroOrg AS STRING)
```

```
CLS
```

```
DIM nombre AS STRING, nombreEnMayusculas AS STRING
```

```
DIM acronimo AS STRING
```

```
INPUT "Teclear nombre organismo: ", nombre  
nombreEnMayusculas = enMayusculas$(nombre)
```

```
CALL acronimoOrga(nombreEnMayusculas, acronimo)
```

```
PRINT
```

```
PRINT nombreEnMayusculas; SPC(5);
```

```
COLOR 10
```

```
PRINT acronimo
```

```
FUNCTION enMayusculas$ (orga AS STRING)
```

```
    DIM i AS INTEGER
```

```
    DIM letra AS STRING * 1
```

```
    DIM orgaMay AS STRING
```

```
    i = 1
```

```
    DO
```

```
        letra = MID$(orga, i, 1)
```

```
        IF i = 1 THEN
```

```
            letra = UCASE$(letra)
```

```
        END IF
```

```
        IF letra <> " " THEN
```

```
            orgaMay = orgaMay + letra
```

```
        ELSEIF letra = " " THEN
```

```
            orgaMay = orgaMay + letra
```

```
            i = i + 1
```

```
            orgaMay = orgaMay + UCASE$(MID$(orga, i, 1))
```

```
        END IF
```

```
        i = i + 1
```

```
    LOOP UNTIL i > LEN(orga)
```

```
    enMayusculas$ = orgaMay
```

```
END FUNCTION
```

```

SUB acronimoOrga (nombreOrg AS STRING, acroOrg AS STRING)
  DIM i AS INTEGER
  DIM letra AS STRING * 1
  FOR i = 1 TO LEN(nombreOrg)
    letra = MID$(nombreOrg, i, 1)
    IF ASC(letra) >= 65 AND ASC(letra) <= 90 OR ASC(letra) = 165
  THEN
      acroOrg = acroOrg + letra + "."
    END IF
  NEXT i
END SUB

```