

# INTRODUCCIÓN A LA PROGRAMACIÓN

Parte II

- Septiembre 1999 -

**APELLIDOS:**

**NOMBRE:**

**GRUPO:**

## PROBLEMA 2 (3.5 puntos)

Disponemos de unas tablas declaradas de la siguiente forma:

```
tipo
    tipo_tabla = tabla [1..10,1..5,1..3] de real;
...
ftipo
...
var
    matriz_entrada, matriz_salida:tipo_tabla;
    la_media:real;
fvar
```

La tabla "matriz\_entrada" se encuentra completamente cargada de datos. Tenemos también las dos siguientes sentencias en el código principal:

- (1) Media(matriz\_entrada,la\_media);
- (2) Distancia(Maximo(matriz\_entrada),matriz\_entrada,matrizsalida);

La primera sentencia llama a la acción "Media" y calcula la media de todos los valores de la tabla, devolviendo su valor en la variable "la\_media".

La segunda sentencia llama a la acción "Distancia", la cual calcula la diferencia entre cada elemento de la tabla de "matriz entrada" y el valor máximo de la propia tabla "matriz entrada" (calculado por la función "Maximo") y lo almacena en "matriz-salida".

Realice la **declaración** e **implementación** de los 3 subprogramas "Media", "Maximo" y "Distancia".

### PROBLEMA 3 (3.5 puntos)

Se desea tener un cierto control del número de coches de que dispone un concesionario de automóviles de segunda mano, para lo cual podrá hacer uso del módulo COC. Complete el módulo CONCHLV que viene a continuación:

Módulo COC;

exporta

tipo tcoche;

(\* para contener información sobre un coche: modelo, marca, número de serie, precio... \*)

tipo tproveedor,

(\* información relativa sobre un proveedor de coches\*)

funcion modelocoche (c : tcoche) : tabla[1..20] de carácter;

(\* devuelve el modelo del coche c \*)

funcion stock (c : tcoche) : entero;

(\* devuelve el número de unidades que hay del coche c \*)

funcion nombrecoche (c: tcoche): tabla [1..20] de carácter;

(\* devuelve el nombre del coche c \*)

accion proveedores ( c : tcoche; var prov: tabla [1..3] de tproveedor, var n: entero);

(\* guarda en prov en sus primeras n posiciones los proveedores del coche c\*)

funcion nombreproveedor (a: tproveedor) : tabla[1..30] de carácter;

(\* devuelve el nombre del proveedor a \*)

función comparar (a,b : tabla[1..20] de carácter) : booleano;

(\* devuelve cierto si a y b son idénticas, en caso contrario falso \*)

fexporta

fmódulo

Módulo CONCHLV;

...

exporta

tipo tcoches;

(\* información relativa a un conjunto de coches ... \*)

funcion cocheunico ( a : tcoche; n : entero) : booleano;

(\* devolverá cierto si no existe ningún coche con el mismo modelo dentro de los n coches que hay en a, en caso contrario devolverá falso \*) (\* esquema de búsqueda \*)

accion listado (cc : tcoches; n : entero);

(\* sacará por pantalla para los n coches que hay en cc cuyo stock sea superior a 10 el modelo del coche y el nombre de los proveedores de dicho coche \*) (\* esquema de recorrido \*)

fexporta

implementacion

tipo tcoches = tabla[1..200] de tcoche;

(\* si hay m coches estarán en la tabla desde la posición 1 hasta la m \*)

...

fimplementacion

fmódulo