

# Ejercicios del Tema 4

## Estructuras

### 1. Distancia entre puntos

Escribe un programa que calcule la distancia entre dos puntos introducidos por el usuario, y que decida el cuadrante en el que está localizado cada punto. Este programa debe emplear la estructura punto capaz de almacenar dos coordenadas x e y de tipo float, y debe utilizar dos funciones, cuadrante y distancia. La función cuadrante devuelve un número entero indicando el cuadrante del punto, y la función distancia devuelve un número real con la distancia entre dos puntos.

```
int cuadrante(punto p);  
float distancia(punto p1, punto p2);
```

### 2. Vector de puntos

Realiza un programa que pida al usuario un número n de puntos indicando sus coordenadas (estructura punto definida en el ejercicio anterior) y los introduzca en un vector. El programa debe indicar el total de puntos que hay en cada uno de los cuadrantes.

Ejemplo de funcionamiento:

```
Indique el numero de puntos a introducir:  
3  
Introduzca x1, y1:  
3 6  
Introduzca x2, y2:  
-1 8  
Introduzca x3, y3:  
-3 3  
El total de puntos por cuadrante es:  
Cuadrante 1 = 1  
Cuadrante 2 = 2  
Cuadrante 3 = 0  
Cuadrante 4 = 0
```

### 3. Distancia entre dos instantes temporales

Escribe un programa que calcule la distancia temporal entre dos instantes. Este programa debe emplear la estructura tiempo (horas, minutos, segundos) para almacenar la información de cada instante. Asimismo debe emplear la función dTiempo que admite como argumentos dos estructuras tiempo y devuelve una estructura tiempo. Por ejemplo, si la estructura t1 representa el instante 3:45:15 y la estructura t2 representa el instante 9:44:03, la función dTiempo debe devolver la estructura 5:58:48. Es importante tener en cuenta la posibilidad de que entre t1 y t2 se encuentre la medianoche.

```
tiempo dTiempo(tiempo t1, tiempo t2);
```

### 4. Contacto

Define la estructura contacto con los campos nombre, primer apellido, segundo apellido, y fecha de nacimiento. Esta estructura emplea internamente la estructura fecha para almacenar la fecha de nacimiento (día, mes y año).

Escribe un programa que lea los datos de dos contactos, y los almacene en un vector de estructuras. A continuación debe imprimir los datos del que sea más mayor, o ambos si tienen la misma edad.

El programa debe estar construido en base a dos funciones: `printContacto` debe imprimir el contenido de una estructura `contacto`; `compFecha` compara dos estructuras `fecha` y devuelve 1 cuando la primera es mayor, -1 cuando es menor, 0 cuando son iguales.

```
void printContacto(contacto persona);  
int compFecha(fecha f1, fecha f2);
```