



# Fundamentos de Programación Grado en Ingeniería Informática

- Septiembre 2018 -

Dpto. Tecnologías de la Información  
Escuela Técnica Superior de Ingeniería - Universidad de Huelva

## NOTA

La puntuación máxima de este examen es de 5 puntos

### Problema 1.-

2.5 puntos

La empresa de gestión de puertos de montaña Onucheck desea desarrollar un software que, mediante un conjunto de cámaras automatizadas, controle y detecte los vehículos cuando pasan por un tramo de carretera que contenga un puerto de montaña peligroso. Cada cámara actualiza la siguiente información de los vehículos cuando pasan delante de la misma.

```
typedef char Cadena[50];  
struct Vehiculo  
{  
    Cadena Matricula;        //Matricula de vehículo  
    float hora;              //Hora de pasada  
    int Kilometros;          //Kilómetro de la carretera.  
    bool Cadenas;            //true si lleva cadenas puestas.  
};
```

En esta estructura se almacena la **matrícula** del coche, la **hora** de pasada delante de la cámara, el **kilómetro** de la carretera en el que fue detectado y si llevaba puestas o no **cadenas** de nieve.

Para almacenar y gestionar todos los datos tomados por las cámaras del tramo de carretera, se dispone de la siguiente clase que define los atributos y métodos para realizar una funcionalidad básica.

```
class PuertoMontana  
{  
    Vehiculo Trafico[10000];  
    int NVehiculos;  
    //Número de vehículos detectados, desde Trafico[0] hasta Trafico[NVehiculos-1]  
    Cadena Nombre;        //Nombre del puerto de montaña  
    bool Abierto;          //true si el puerto está abierto  
public:  
    bool ActualizarPuerto(float pTemperaturas[3]);  
    void Dame_Nombre(Cadena pNombre);  
    int VehiculosEnPeligro(int KmInicial, int KmFinal);  
};
```

### **\*\* Código a implementar \*\***

Se pide implementar los métodos **ActualizarPuerto**, **Dame\_Nombre** y **VehiculosEnPeligro**.

Los métodos de la class PuertoMontana son:

**bool ActualizarPuerto(float pTemperaturas[3]):** Si la media aritmética de las tres temperaturas que se le pasan en **pTemperaturas** supera los 0°C pondrá **Abierto** a true, en caso contrario a false. El método devuelve el estado final del puerto si está abierto o cerrado. **(1 punto)**

**void Dame\_Nombre(Cadena pNombre):** Devuelve el nombre del puerto de montaña en el parámetro **pNombre**. **(0,3 puntos)**

**int VehiculosEnPeligro(int KmInicial, int KmFinal):** Método que devuelve el número de vehículos que pueden estar en peligro. Para determinar si un vehículo está en peligro se tiene que dar la condición de que el puerto esté cerrado, el vehículo esté entre los kilómetros dados por parámetro (**KmInicial** y **KmFinal**) y que no lleve cadenas. En caso de que el puerto esté abierto siempre devolverá **-1**. **(1,2 puntos)**

## Problema 2.-

2.5 puntos

Dada las siguientes estructuras de datos para la gestión de un edificio compuesto de hasta veinte plantas, cada planta se compone de hasta diez viviendas, y **suponiendo todos los métodos implementados**, diseñe un main que **muestre por pantalla el dni y el nombre del propietario de la vivienda cuyo precio de adquisición sea el más alto**.

```
struct VIVIENDA {
    float precio;          /* Precio de adquisición de la vivienda*/
    char dni[10];          /* DNI del propietario de la vivienda */
    char nombre[30];       /* Nombre del propietario de la vivienda*/
};

class PLANTA {
    VIVIENDA viviendas[10];
    int nc; /* Número de viviendas almacenadas en la tabla desde la posición 0 hasta nc-1 */
public:
    int num_viviendas();
    /* Número de viviendas almacenadas*/
    VIVIENDA devolver_vivienda (int n);
    /* Devuelve la vivienda que está en la posición n de la tabla viviendas */
    void agregar (VIVIENDA c);
    /* Añade la vivienda c en la tabla viviendas*/
};

class EDIFICIO {
    PLANTA plantas[20];
    int np; /*Número de plantas almacenadas en la tabla desde la posición 0 hasta np-1 */
public:
    int num_plantas();
    /*Número de plantas almacenadas*/
    PLANTA devolver_planta (int n);
    /* Devuelve la PLANTA almacenada en la posición n de la tabla plantas*/
    void cargar();
    /* Carga desde teclado los datos del edificio*/
};

int main () {
    EDIFICIO x;
    x.cargar();
    ...
    /*Muestre por pantalla el dni y el nombre del propietario de la vivienda que tiene el precio de
    adquisición más alto*/
    ...
    return 0;
}
```