

**Algoritmos y Estructuras de Datos – Práctica Primer Parcial 2020**

Prof: Marcelo Lipkin – Director de Cátedra: Dr. Oscar Bruno

**Temas evaluados:** Interpretación de enunciados; manejo de variables, registros y vectores, ciclos repetitivos y módulos. Resolución de problemas mediante Algoritmos y lenguaje C++.

**Enunciado:**

Para una locadora de automotores se necesita desarrollar un programa para llevar un registro de los alquileres de los autos, donde conste si el auto está alquilado, la cantidad de kilómetros reales (la suma de los kilómetros recorridos en los alquileres).

Los datos a procesar son:

**Autos para alquilar**, se disponen de una flota de hasta 100 autos con los siguientes datos: **patente** (6 caracteres), **marca del auto**; **año de fabricación**; **modelo** (chico, mediano, grande); **cantidad de kilómetros**, **valor diario de alquiler**, **fecha de último alquiler/devolución**, **estado** (A – alquilado, D- depósito). Esta información está desordenada y finaliza con una patente en "0" (en caso que se procese de un archivo, finaliza cuando termina el archivo)

**Alquileres/devoluciones**, por cada alquiler o devolución se dispone de los siguientes datos: **patente** (6 caracteres), **tipo de operación** (A – alquiler, D – devolución), **cantidad de kilómetros** (0 para alquiler, número entero para devolución), **fecha del alquiler o devolución**. Esta información está desordenada y finaliza con una patente en "0" (en caso que se procese de un archivo, finaliza cuando termina el archivo)

Se pide diseñar el algoritmo que procesando los datos ingresados:

**A – (2pts)** Desarrollar la estructura de datos para todo el problema y completar la función main principal (presentada más abajo) que invoca a las distintas funciones solicitadas/agregadas para resolver el problema. Desarrollar la función de ingreso de datos de **autos para alquilar**.

```
Int main () {  
    // definir las variables para autos y aquellas que considere necesarias  
    inicializarAutos(); // completar parámetros  
    ingresarAutos(); // completar parámetros  
    procesarAlquileres(); // completar parámetros  
    mostrarAutosPtoC(); // completar parámetros  
    listadoOrdenadoAutosPtoD(); // completar parámetros  
}
```

**B – (3 pts)** Desarrollar una función para procesar los **alquileres/devoluciones** que mantenga actualizada la cantidad de kilómetros recorridos, muestre para cada devolución el valor a cobrar al cliente y al final de proceso muestre el total facturado por la empresa. Para calcular el monto a cobrar se debe multiplicar el valor diario del auto por la cantidad de días alquilados (calcular mediante la función diferencia –ver al final). Tener en cuenta actualizar fecha de alquiler para los alquileres y actualizar kilómetros recorridos en las devoluciones

**C – (2 pts)** Desarrollar una función muestre en la pantalla todos los autos con la cantidad de kilómetros recorridos y al final del listado el total por cada modelo (chico, mediano, grande).

**D – (3 pts p/posibilidad de promocionar)** Desarrollar una función que emita un listado ordenado en forma decreciente por modelo que contenga solamente aquellos autos (en depósito) que **NO ESTÁN ALQUILADOS**.

Resuelva favoreciendo uso de módulos (además de los solicitados) con los parámetros que corresponda en cada caso.

En caso de necesitar utilice las funciones desarrolladas en la clase sin hacerlas nuevamente.

Para resolver en computadora se adjunta archivos con datos y la estructura general del programa con las funciones para leer los datos de archivos de texto y búsqueda y ordenamiento mediante "templates".

```
struct tFecha {  
    int dia;  
    int mes;  
    int anio; } ;  
  
int diferenciaFechas(tFecha f1, tFecha f2) {  
    int abs((f1.anio-f2.anio)*12 +(f1.mes-f2.mes)*30 + (f1.dia-f2.dia));  
    return dif;  
}  
  
Fecha f1 = {16,7,2020};  
tFecha f2 = {5,5,2020};  
int cantDias = diferenciaFechas(f1, f2);
```