

Ejercicio 1 [35 pts]

- a) Escriba una función genérica *reordenar()* que reciba como argumentos un string *nombre* y dos enteros *pos1* y *pos2*. Asumiendo que el string *nombre* contiene el nombre de un archivo binario que guarda una serie de datos de un mismo tipo. La función debe modificar el archivo de la siguiente forma: Ordenar de menor a mayor desde la posición 0 del archivo hasta *pos1* (incluida), y de mayor a menor desde *pos2* hasta el final del archivo.
- b) Utilice la función desde un programa cliente para modificar 2 archivos "*datafloat.bin*" y "*dataficha.bin*". El 1er archivo tiene flotantes. El 2do archivo está compuesto por datos de un tipo *ficha*, que debe definir conteniendo los campos *dni*, *nombre*, *edad* y *peso*; donde la *edad* define qué dato es mayor o menor.

Ejercicio 2 [35 pts]

Se dispone de un archivo de textos donde cada línea contiene 3 datos separados por ';': dni, apellido, nombres. Escriba una clase que posea:

- a) un constructor que reciba el nombre de un archivo de texto, que organice los datos de cada línea en un struct y almacene el conjunto de structs en un *std::vector*.
- b) Un método *iniciales_mayusculas()* que coloque la 1er letra del apellido y de cada nombre en mayúsculas, y el resto en minúsculas.
- c) Un método *ordenar()* para ordenar la lista por apellido, y a su vez eliminar personas repetidas (igual DNI).
- d) Un método *reescribir()* que no reciba ningún parámetro y sobrescriba el archivo original con el listado depurado.

Ejercicio 3 [30 pts]

Un archivo binario contiene una base de datos de registros de tipo cliente (es un struct como el que se indica abajo) y fue generado por un programa C++.

```
struct cliente {  
    char apellido[30], nombres[40], direccion[40];  
    int dni; dd,mm,aaaa; bool borrado  
};
```

El struct tiene los datos de cada cliente y la fecha de su última compra. El miembro borrado posee el false si realizó una compra en el último año (cliente activo) y true en caso contrario.

Escriba una función C++ cuyo prototipo sea *void actualiza(string nomarchi; int dhoy, mhoy, ahoy);* donde sus argumentos corresponden al nombre del archivo con todos los clientes y los 3 enteros a la fecha actual. La función debe actualizar el archivo colocando **true** en el miembro borrado si el cliente hace más de 1 año al día de hoy que no realizó una compra. Realice la actualización modificando el archivo evitando cargar toda la lista en un contenedor auxiliar.