

**Guía 06 (Listas Ordenadas)**

1. Se tiene guardado, en un vector no ordenado, el listado de alumnos que fueron seleccionados para representar a la facultad en un congreso nacional de ingeniería. De cada alumno se tiene el nombre, el legajo (entero) y su promedio (double). Lamentablemente algunos alumnos no podrán ir al congreso por temas personales y han pedido que se los de baja. Se tiene un segundo vector no ordenado con los alumnos seleccionados que han pedido ser dados de baja.  
**Se pide:**

- a. listar los alumnos seleccionados ordenados por promedio en forma descendente y en caso de empate por nombre en forma ascendente. La idea NO es ordenar el vector, sino insertarlo en una lista ordenada.
- b. Luego mostrar sin orden los alumnos que han pedido darse de baja.
- c. Por último volver a mostrar la lista ordenada de los alumnos seleccionados sin los que hayan sido dados de baja.

**Nota:** Si usa los datos de G06E01-datos.txt para inicializar los vectores, la salida, salvo detalles de formato, debería ser la de G06E01-salida.txt

2. Se dispone del archivo binario G06E02.bin que contiene una lista no ordenada de órdenes de compras. Las mismas tienen los siguientes campos: id es un entero con un número que no se repite en ninguna otra orden. Cliente es un string que se guarda en el archivo binario con un largo de 8 caracteres. Producto también es string y se guarda con largo 10. Cantidad es un entero con la cantidad pedida de ese producto por el cliente.
  - a. Se debe leer el archivo y cargar dos listas, la primera ordenada por cliente y a igualdad de cliente por producto (siempre ascendente). La otra por id (también ascendente). Luego de cargar las listas se mostrarán ambas por pantalla.
  - b. La lista ordenada por id está pensada para consultas, por eso luego de cargar las listas el programa pedirá un id, lo buscará y mostrará todos los datos de la orden de compra correspondiente o bien informará que no hay ninguna con ese número. El ciclo se corta cuando se ingresa un id negativo, ya que todos los id correctos son positivos.
  - c. Luego se pasa al despacho, para eso se pide al operador un nombre de cliente y un producto, de haber, se eliminan todos los registros correspondientes de AMBAS listas. A medida que se borran los registros se van mostrando por pantalla los eliminados, y una vez que se eliminaron todos los que corresponde, se muestran como quedaron ambas listas. Esto se hace en un ciclo hasta que se ingresa EOF.

**Nota:** se provee el archivo G06E02.bin y la salida al mostrar las listas por primera vez.

3. Se tiene un listado de deudas no ordenado en el archivo binario G06E03.bin. Cada deuda tiene un cliente guardado con 8 caracteres, el número del mes en el que tomó la deuda y la cantidad adeudada (como número entero). Se pide primero leer el archivo y armar un lista ordenada por cliente y a igualdad de cliente por mes, luego mostrar esa lista inicial por pantalla. Luego se quiere hacer la consolidación de la deuda, esto es, para cada cliente, si tiene varios registros con el mismo mes, dejar uno solo con el total de la deuda en dicho mes. Para ello debe programar dos funciones auxiliares. La función “borrar siguiente” recibe como parámetro un puntero a un nodo de la lista y borra el nodo siguiente. La función “consolidar” recibe la lista y la modifica para dejar la versión consolidada, usando la función “borrar siguiente” como ayuda. Al finalizar se mostrará por pantalla la lista consolidada.

**Nota:** se provee el archivo G06E03.bin y G06E03-salida.txt para que puedan verificar la corrección del programa.