

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática



Ingeniería Informática

PROGRAMACIÓN ORIENTADA A OBJETOS

UNIDAD 06
Flujos de Entrada/Salida

Guía de trabajos prácticos
2016

UNIDAD 06

Flujos de Entrada/Salida

Ejercicio 1

Escriba a través de un programa C++ un archivo binario llamado *grupo.dat*, formado por un conjunto de 200 pares de números generados aleatoriamente. Cada par de datos se conforma por un entero y un flotante.

Ejercicio 2

Escriba un programa abra el archivo generado en el ejercicio anterior y solicite al usuario que ingrese un entero, un flotante y una posición. El programa debe sobrescribir el par en la posición ingresada por el usuario por el nuevo par. Luego muestre la lista de datos en consola mostrando un par por línea.

Ejercicio 3

Escriba un programa que genere un archivo binario con una lista de 100 enteros ordenados de forma creciente (por ejemplo: 0, 5, 10, 15, 20, 25...). Luego escriba otro programa que permita insertar (no reemplazar) un nuevo dato de tipo entero, ingresado por el usuario, en el archivo manteniendo el orden creciente. Evite utilizar vectores auxiliares.

Ejercicio 4

Con un software de tratamiento de imagenes llamado *WellDesign* se han generado 78 archivos con los nombres "design1.wds", "design2.wds", ... "design78.wds". El software fue creado para *Windows XP*. Estos archivos presentan el problema que en el nuevo sistema operativo *Windows 10* al tratar de abrirlos con *WellDesign* presentan un molesto mensaje. Al consultar al soporte tecnico de la empresa que desarrollo *WellDesign* nos mencionan que si se modifica la cabecera de cada archivo se soluciona el problema. A partir del byte 178 se debe reemplazar "mswxp" por "msw10", y a partir del byte 239 reemplazar el valor entero 2003 por el valor 2016. Escriba un programa en C++ que realice esas correcciones en los 78 archivos almacenados.

Ejercicio 5

Escriba una clase para gestionar la tabla de las N mejores puntuaciones de un juego. La clase debe permitir ingresar un nuevo puntaje (y encargarse de ubicarlo en la posición de la tabla que corresponda, o descartarlo), consultar los datos, y almacenar y recuperar los mismos mediante un archivo binario.

Cuestionario

- ♦ ¿Qué entendemos por archivo?
- ♦ ¿Cuál es la importancia de utilizar archivos?
- ♦ ¿Cuál es la diferencia entre archivos de texto y archivos binarios?
- ♦ ¿Puede abrirse un archivo de texto como binario? ¿Y un archivo binario como archivo de texto?
- ♦ ¿Puede guardarse un objeto de tipo *std::string* en un archivo de texto? ¿Y en un archivo binario? Justifique sus respuestas.

Ejercicios Adicionales

Ejercicio 1

Escriba una función *CreateDummyFile(...)* que reciba el nombre de un archivo y un entero con su tamaño en bytes. La función debe crear un archivo binario del tamaño especificado y llenarlo con valores aleatorios (basura).

Escriba una sobrecarga de la función en la que se especifique el tamaño del archivo mediante un *std::string*, por ejemplo: "100Mb", "10Kb" ó "893b".

Ejercicio 2

Proponga una clase *Persona* para representar a una persona a partir de su nombre, apellido y fecha de nacimiento con día, mes y año. Diseñe una clase llamada *RecursosHumanos* que permita manejar los datos de un grupo de personas. La clase debe poseer métodos para:

- a) Guardar los datos de las personas en un archivo binario.
- b) Cargar los datos de las personas desde un archivo binario.
- c) Agregar una persona.
- d) Eliminar una persona a partir de su nombre o apellido.
- e) Indicar la cantidad de personas en la base de datos.