Universidad de Antioquia
Departamento de Ingeniería Electrónica y Telecomunicaciones

UNIVERSIDAD DE ANTIOQUIA
1803

Informática II - Practica No. 3
Sesiones 1 y 2

Objetivos

- Aprender a manipular cadenas de caracteres en C++.
- Aprender a gestionar archivos en C++.
- Conocer cómo manejar Excepciones.

Ejemplos

1. El profesor mostrará cómo crear un archivo y escribir en él.
    a. Lectura.
    b. Lectura/escritura.
    c. Manejo de excepciones.
2. El profesor mostrará cómo leer datos de un archivo.
    a. Carácter a carácter.
    b. Línea por línea.
3. Se realizará un ejemplo de cómo convertir caracteres leídos de un archivo a números.

Ejercicios

Tenga en cuenta que esta práctica debe ser realizada de las siguientes dos formas:

- Usando manipulación de arreglos de *char*.
- Usando la clase *string*.

1. Escriba un programa para codificar archivos de texto. El programa debe recibir un número entero n que será la semilla de codificación y un número para seleccionar el método de codificación.

Primer método:
Para codificar se convierte el archivo a binario, luego se separa en bloques de n bits. En el primer bloque se cambian todos los 1 por 0 y viceversa. Para los grupos siguientes se debe contar el número de 1s y 0s en el grupo anterior y hacer cambios según la siguiente regla:

- Si hay igual cantidad de 1s y 0s se invierte cada bit del grupo.
- Si hay mayor cantidad de 0s se invierte cada 2 bits.
- Si hay mayor cantidad de 1s se invierte cada 3 bits.

Ej: Si se ingresa 4 y el archivo contiene AbCd.

El archivo en binario sería:
`01000001011000100100001101100100`
El archivo codificado sería:
`10110100001111010001011010011011`

***Nota:** Tener en cuenta cuando se aplican las reglas a partir del segundo bloque, se debe tomar los grupos de bits del archivo binario sin codificar.

Segundo método:
Para codificar se convierte el archivo a binario, luego se separa en bloques de n bits. Cada bit dentro de un grupo se desplaza una posición, de manera que el primer bit del grupo codificado corresponde al último bit del grupo sin codificar, el segundo bit codificado corresponde al primero sin codificar y así sucesivamente hasta que el último corresponde al penúltimo sin codificar.

Ej: Si se ingresa 4 y el archivo contiene AbCd.

El archivo en binario sería:
`01000001011000100100001101100100`
El archivo codificado sería:
`00101000001100010010100100110010`

Nota: El programa debe generar un archivo de salida con la información binaria codificada. El programa debe recibir los nombres del archivo fuente y el archivo de salida por consola.

2. Escriba un programa para decodificar el archivo generado por el programa anterior. Igualmente debe recibir la semilla, el método de codificación y los nombres de los archivos de entrada y salida.

Aplicación

Modelar un sistema de registro de usuarios que tenga las siguientes funcionalidades:

1. Acceder al sistema como usuario administrador.

Nota: La validación de acceso se debe hacer abriendo el archivo sudo.txt y comparando con la clave registrada que debe estar encriptada.

2. Una vez se valide el ingreso como administrador, se podrá realizar el registro de usuarios del cajero electrónico. El formato de registro será el siguiente: cédula, clave, saldo (COP).

3. Debe permitir acceso a los usuarios del sistema y permitir realizar las siguientes operaciones:
    a. Consultar saldo.
    b. Retirar dinero especificando la cantidad deseada.

4. Se debe actualizar la información del cliente cada vez que acceda al sistema teniendo en cuenta las siguientes características:
        i. Cada ingreso al cajero por parte de los clientes para retirar dinero o consultar el saldo tiene un costo de 1000 COP.
        ii. Cuando se retire dinero del cajero se debe actualizar el saldo en la cuenta del usuario.

Nota: Todas las transacciones que se realicen en el cajero deben tener la información encriptada.