CS1112: Programación II

Unidad 9: Archivos

Sesión de Laboratorio - 14A

Profesores:
María Hilda Bermejo mbermejo@utec.edu.pe
Estanislao Contreras econtreras@utec.edu.pe
Jorge Villavicencio jvillavicencio@utec.edu.pe
Edson Mendiolaza emendiolaza@utec.edu.pe
Ian Paul Brossard ibrossard@utec.edu.pe
Jose Chavez jchaveza@utec.edu.pe
Julio Yarasca jyarascam@utec.edu.pe
Percy Quevedo pquevedo@utec.edu.pe
Wilder Nina wnina@utec.edu.pe
José Fiestas jfiestas@utec.edu.pe

Material elaborado por:
Maria Hilda Bermejo

Índice:

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

9 Unidad 9: Archivos

Logro de la sesión:

Al finalizar la sesión, los alumnos desarrollan sus programas utilizando archivos.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejercicio 1:

Crear un programa que solicite el nombre de archivo y que cuente las vocales del archivo y genere otro archivo cuyo nombre es ***vocales.txt***.

El archivo debe mostrar (en cada línea):

Por ejemplo:

```
Created by Jose Fiestas on 6/12/20.
#include <iostream>
Vocal y frecuencia en el texto:
a: 3
e: 6
i: 3
o: 3
u: 1
```

Ejercicio 2:

Crear un programa que solicite un número n, nombre de archivo y los datos de "*n*" personas (dni, nombre, edad, salario).
El programa debe grabar esos datos en el archivo.
Genere el archivo con la fecha y hora de registro y los datos de las personas

Por ejemplo:

```
----------------------------------------
Registro del Thu Nov 16 09:10:22 2023
----------------------------------------
    DNI   Nombre      edad   salario
```

Ejercicio 3:

Lea un archivo con una lista de números enteros (el primer número del archivo es la cantidad total de números)
Almacene la lista de números en un vector
Ordene el vector y muestre la lista ordenada
Implemente la búsqueda de un elemento en el vector

Por ejemplo:

```
10, 11, 12, 13, 21, 21, 22, 24, 31, 33, 35, 40, 43, 44, 49, 53, 53, 58, 62, 75, 83, 83, 92, 97,
Ingrese numero a buscar: 31
31 pertenece a la lista
```

```
10, 11, 12, 13, 21, 21, 22, 24, 31, 33, 35, 40, 43, 44, 49, 53, 53, 58, 62, 75, 83, 83, 92, 97,
Ingrese numero a buscar: 30
30 no pertenece a la lista
```

Ejercicio 4:

Escriba un programa que permita realizar lo siguiente:

1. Modifique el programa crear una clase CAleatorio e implemente los siguientes métodos en la clase: LoadData(), InsertionSort(), QuickSort(), BinarySearch().
   Modifique la función **std::fstream ordenada(nombreArchivoOrdenado, std::ios::out)** Para que no utilice un mapa al ordenar los elementos del archivo sino por el contrario que utilice dos métodos privados que apliquen InsertionSort o QuickSort al vector que contiene los elementos leídos del archivo. Agregue un método a la clase CAleatorio para generar una búsqueda binaria sobre los elementos del archivo cargados en un vector.

Ejercicio 4:

Main.cpp

```
#include <iostream>
#include "CAleatorio.h"

int main() {
    CAleatorio *ptr = new CAleatorio();
    ptr->LoadData();
    return 0;
}
```

Ejercicio 4:

CAleatorio.cpp

```
#include <iostream>
#include <Vector>
#include "Tipos.h"
using namespace std;
class CAleatorio {
   private:
        vector<Entero> numeros;
        Boleano loaded;
        void InsertSort();
        void QuickSort();
   public:
        CAleatorio();
        void LoadData();
        void Sort();
        void Show();
        Entero BinarySearch();

};
```

Ejercicio 5:

Crear un programa que crea un clase que se denominará **contador_palabras_t** y que cuente las palabras de un archivo y lo devuelva a través de un método, la lectura del archivo deberá realizarse usando el operador **<<** que debe ser sobrecarga de modo que:

```
contador_palabras_t contador;
contador << "origen.txt";
```

Nota: Como primera aproximación al problema considere el espacio como el carácter que define el fin o inicio de una palabra.

Ejercicios Templates

Ejercicio 1

Escriba un programa que implemente un algoritmo de ordenación para números enteros y números de punto flotante.

a) Implemente la solución con una clase **COrdenar** y con dos hijas **COrdenarEnteros**, **COrdenarFlotantes**.
b) Implemente la solución usando **template**.
c) Escriba un programa main que use las dos implementaciones.
d) Discuta con sus compañeros ventajas y desventajas de ambas soluciones.

Ejercicio 2

Escriba un programa que implemente la función genérica **input** similar a la función input de de python:

```python
# python
entero = int(input("Ingrese dato"))
print (entero)
```

```cpp
// C++
auto entero = input<int>("Ingrese dato");
cout << entero << endl;
```

Ejercicio 3

Escriba un programa que implemente la clase genérica **rango** que permita devolver un rango de valores:

```
rango<int> r1(1, 20, 2); // start = 1, stop = 20, step = 2
cout << r1++ << endl;
while (!r1.stop())
   cout << r1++ << " "; // 1 3 5 7 9 11 13 15 17 19

rango<char> r2('A', 'J'); // start = 1, stop = 20
while (!r2.stop())
   cout << r2++ << endl; // 'A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I'
```

Explorando lo aprendido

- ¿Qué diferencia existe entre ofstream y ifstream?
- ¿Qué diferencia existe entre ofstream y fstream?
- ¿Cuáles son las formas de abrir un archivo?
- Considerando que el objeto **archivo** es **std::fstream** y **texto** es **std::string** ¿Que valor retorna **archivo >> texto** si se alcanzó el final del archivo?
- ¿Qué tipos de templates existen?
- ¿Qué ventajas tiene programar con templates?
- ¿Qué desventajas encuentra?
- Dentro de los parámetros de un template ¿Existe alguna diferencia utilizar class en vez de typename o viceversa?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Bibliografía:

Deitel. P.J. and Deitel. H. M. (2016) C++ How to Program, Prentice Hall.

Stroustrup, Bjarne (2013). The C++ Programming Language, 4th Addison-Wesley.

Eckel, Bruce, 2000. Thinking in C++, Vol 1: Introduction to Standard C++, 2nd Edition, Prentice Hall

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

¡Nos vemos en la siguiente clase!

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA