

Universidad Privada de Tacna

Facultad de ingeniería



Curso: Estructura de datos

Docente: Haydee Sisa Yataco

Integrantes:

- Milton H Flores Chino
- Gustavo Vargas Calizaya
- Antony Solórzano Zegarra

Tacna, 27 de Noviembre de 2024

Problema 1:

Escribe un programa en C++ que realice las siguientes acciones:

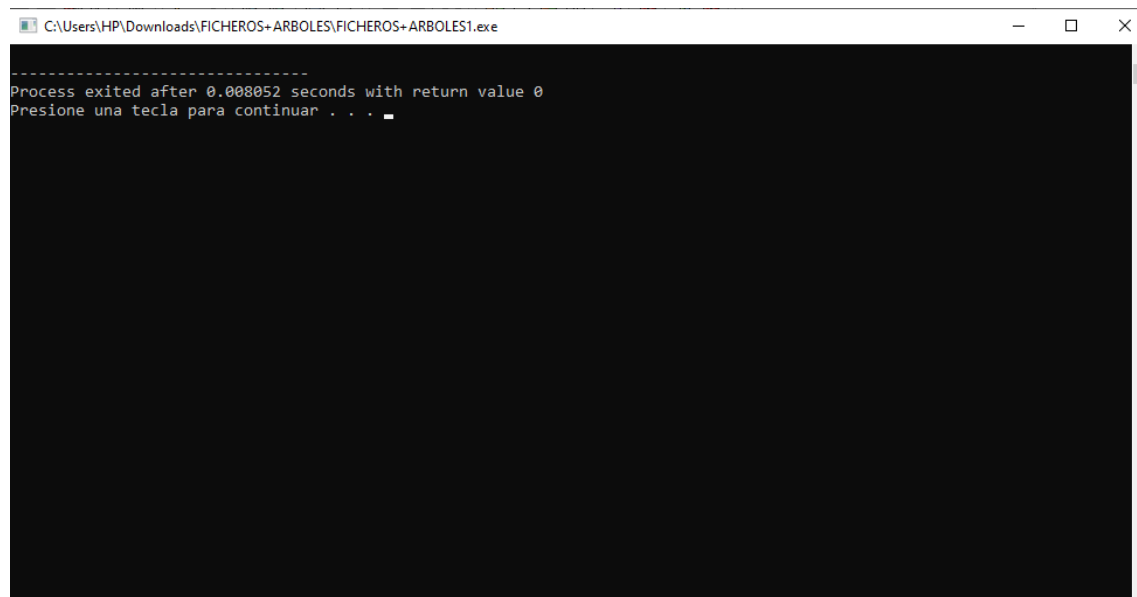
1. Cree un archivo de texto llamado datos.txt.
2. Escriba en el archivo tres líneas de texto:

Primera línea: "Primera línea de texto"

Segunda línea: "Segunda línea de texto"

Tercera línea: "Última línea de texto"

```
1  #include <fstream>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      ofstream archivo; // objeto de la clase ofstream
7
8      archivo.open("datos.txt");
9
10     archivo << "Primera linea de texto" << endl;
11     archivo << "Segunda linea de texto" << endl;
12     archivo << "Ultima linea de texto" << endl;
13
14     archivo.close();
15     return 0;
16 }
```



Problema 2:

Escribe un programa en C++ que haga lo siguiente:

1. Intente crear un archivo llamado EJEMPLOS.TXT en modo de escritura (ios::out).
2. Verifique si el archivo se ha creado correctamente. Si no es posible crearlo, debe mostrar el mensaje:
"Incapaz de crear este o abrir el fichero".
3. Si el archivo se creó exitosamente, escriba en el archivo las siguientes líneas:
"1 5.0 APROBADO"
"2 1.1 SUSPENSO"
"3 8.0 NOTABLE"
4. Cierre el archivo correctamente al terminar.

```
1 #include <fstream> // Biblioteca para el manejo de ficheros
2 #include <iostream> // Biblioteca para la entrada-salida estándar
3 using namespace std;
4 int main(){
5     ofstream fichout("EJEMPLOS.TXT", ios::out);
6     if (!fichout)
7         cout << "\n Incapaz de crear este o abrir el fichero \n";
8     else {
9         fichout << 1 << " " << 5.0 << " APROBADO" << endl; // Escritura en el fichero
10        fichout << 2 << " " << 1.1 << " SUSPENSO" << endl;
11        fichout << 3 << " " << 8.0 << " NOTABLE " << endl;
12        fichout.close();
13    }
14 } // Fin del main
```

Problema 3:

Escribe un programa en C++ que lea un archivo de texto llamado EJEMPLO5.TXT, el cual contiene líneas con un entero, un número flotante y una cadena de texto. El programa debe:

1. Abrir el archivo y verificar si se pudo abrir correctamente.
2. Leer los datos de cada línea y mostrarlos en el siguiente formato:

<Número> <Valor flotante> <Cadena de texto>

3. Continuar leyendo hasta el final del archivo y luego cerrarlo.

```
1  #include <fstream> // Biblioteca para el manejo de ficheros
2  #include <iostream> // Biblioteca para la entrada-salida estándar
3  using namespace std;
4  typedef char TCadena[30];
5
6  int main() {
7      int i;
8      float r;
9      TCadena cad;
10
11     ifstream fichin("EJEMPLO5.TXT"); // Declaración y apertura del fichero
12     if (!fichin) {
13         cout << "\n Incapaz de crear o abrir el fichero ";
14     } else {
15         cout << "\n Exito. Se logro la Lectura adelantada!!! ";
16         while (!fichin.eof()) {
17             cout << i << " "; // Lectura de valores en el fichero
18             fichin >> r;
19             cout << r << " "; // Lectura de valores en el fichero
20             fichin >> cad;
21             cout << cad << "\n"; // Lectura de valores en el fichero
22             fichin >> i;
23         }
24         fichin.close();
25     }
26     return 0;
27 }
```

```
Exito. Se logro la Lectura adelantada!!! 0 1 2
3 4 5
6 7 8
9 10 11

-----
Process exited after 0.009197 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 4 :

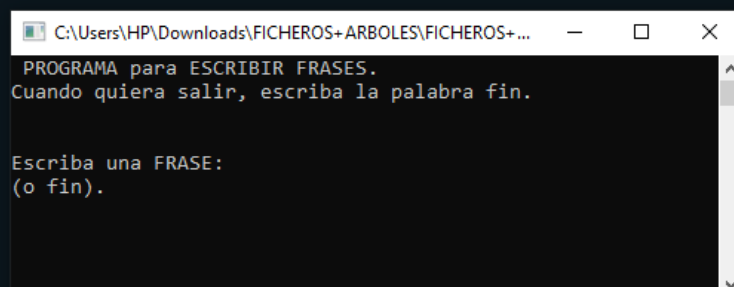
Crea un programa en lenguaje C que permita al usuario escribir frases y almacenarlas en un archivo de texto denominado registroDeUsuario.txt. El programa debe funcionar de la siguiente manera:

1. El usuario debe ingresar frases en el programa.
2. Cada frase que ingrese el usuario será escrita en el archivo registroDeUsuario.txt, una por línea.
3. El programa continuará pidiendo frases al usuario hasta que este ingrese la palabra "fin", la cual indicará el fin del proceso de escritura.
4. Cuando se ingrese "fin", el programa debe detenerse y cerrar el archivo.

Requisitos adicionales:

- El programa debe mostrar un mensaje de bienvenida explicando el funcionamiento del programa.
- Asegúrese de manejar adecuadamente la apertura y cierre del archivo.
- El programa debe usar funciones estándar de C como fopen(), gets(), fprintf(), y fclose().
- Asegúrese de que el código funcione de manera continua hasta que el usuario decida salir con la palabra "fin"

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      FILE *ptfichero;
6      char fin[] = "fin";
7      char frase[60];
8
9      ptfichero = fopen("registroDeUsuario.txt", "wt");
10     printf(" PROGRAMA para ESCRIBIR FRASES.\nCuando quiera salir, escriba la palabra fin.\n\n");
11
12     do {
13         puts("\nEscriba una FRASE:\n(o fin). \n");
14         gets(frase);
15         if (strcmp(frase, fin) == 0) {
16             break;
17         }
18         fprintf(ptfichero, "%s\n", frase);
19     } while (strcmp(frase, fin) != 0);
20
21     fclose(ptfichero);
22     return 0;
23 }
```



Ejercicio 5:

Realiza un programa en C que permita al usuario escribir frases y almacenarlas en un archivo de texto llamado frases.txt. El programa debe solicitar al usuario que ingrese frases una por una, guardarlas en el archivo y finalizar la escritura cuando el usuario presione "Enter" sin ingresar texto. Después de almacenar las frases, el programa debe leerlas desde el archivo y mostrarlas en pantalla. Al final, mostrará un mensaje de despedida.

```
#include <stdio.h>
#include <string.h>

int main() {
    FILE* ficheroU;
    char frase[61];
    int i = 0;

    ficheroU = fopen("frases.txt", "wt");
    printf(" PROGRAMA para ESCRIBIR y almacenar FRASES.\n");
    printf("Cuando quiera salir, simplemente pulse \"Intro\".\n\n");

    do {
        if (i == 0)
            puts("\nEscriba una FRASE:\n(o pulse \"Intro\"). \n");
        else
            puts("\nEscriba otra FRASE:\n(o pulse \"Intro\"). \n");

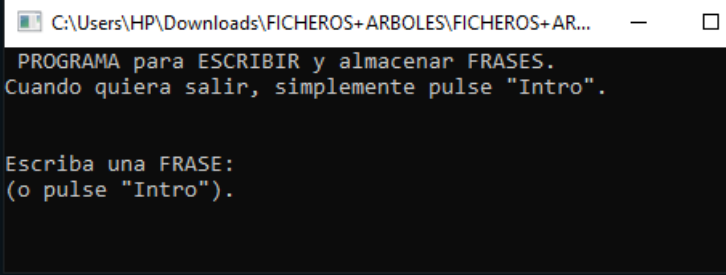
        gets(frase); // Se recomienda reemplazar `gets` por `fgets` por seguridad
        fprintf(ficheroU, "%s\n", frase);
        i++;
    } while (strcmp(frase, "") != 0);

    printf("He aquí lo que escribió:\n\n");
    fclose(ficheroU);

    ficheroU = fopen("frases.txt", "rt");
    do {
        fgets(frase, 60, ficheroU);
        puts(frase);
    } while (!feof(ficheroU));

    getchar();
    printf("...Hasta luego!");
    getchar();

    fclose(ficheroU);
    return 0;
}
```



Ejercicio 6

Crear un programa que, guarde en un archivo el nombre y sección de las 3 personas del grupo que hacen esta experiencia práctica.

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    string nombres[3] = {
        "Antony Solorzano",
        "Gustavo Vargas",
        "Miltom Flores"
    };

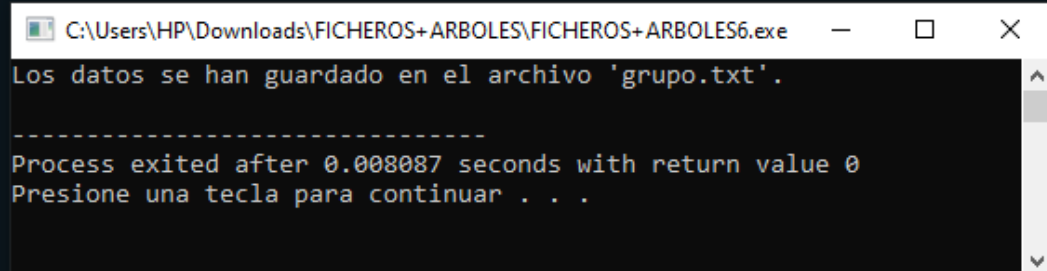
    string seccion = "Sección A";
    ofstream archivo("grupo.txt");
    if (!archivo) {
        cout << "No se pudo crear el archivo." << endl;
        return 1;
    }

    archivo << "Nombres y Secciones del Grupo:" << endl;
    for (int i = 0; i < 3; i++) {
        archivo << i + 1 << ". Nombre: " << nombres[i]
            << " - Sección: " << seccion << endl;
    }

    archivo.close();

    cout << "Los datos se han guardado en el archivo 'grupo.txt'." << endl;

    return 0;
}
```



```
C:\Users\HP\Downloads\FICHEROS+ARBOLES\FICHEROS+ARBOLES6.exe
Los datos se han guardado en el archivo 'grupo.txt'.
-----
Process exited after 0.008087 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 7

Implementa un programa en C++ que permita gestionar un Árbol Binario de Búsqueda (ABB). El programa debe realizar las siguientes funciones:

1. **Crear el árbol binario de búsqueda (ABB):** Permite insertar nodos con valores enteros proporcionados por el usuario, respetando la propiedad de orden del ABB:
 - Los valores menores que el nodo raíz se colocan en el subárbol izquierdo.
 - Los valores mayores que el nodo raíz se colocan en el subárbol derecho.
2. **Mostrar el árbol en estructura visual:** Representa gráficamente el árbol con niveles para que el usuario pueda visualizar su estructura jerárquica.
3. **Recorridos del árbol:** Implementa y muestra los resultados de los tres tipos básicos de recorridos de un ABB:
 - **En orden (In-Order):** Recorre primero el subárbol izquierdo, luego la raíz, y después el subárbol derecho.
 - **Preorden (Pre-Order):** Recorre primero la raíz, luego el subárbol izquierdo y, por último, el subárbol derecho.
 - **Postorden (Post-Order):** Recorre primero el subárbol izquierdo, luego el subárbol derecho y, por último, la raíz.
4. **Interacción con el usuario:** Solicita al usuario el número de nodos que desea insertar en el árbol y el valor de cada nodo.

```
1 #include <iostream>
2 #include <stdlib.h>
3 using namespace std;
4
5 struct nodo{
6     int nro;
7     struct nodo *izq, *der;
8 };
9
10 typedef struct nodo *ABB;
11 /* Es un puntero de tipo nodo que hemos llamado ABB, que utilizaremos
12    para mayor facilidad de creación de variables */
13
14 ABB crearNodo(int x)
15 {
16     ABB nuevoNodo = new(struct nodo);
17     nuevoNodo->nro = x;
18     nuevoNodo->izq = NULL;
19     nuevoNodo->der = NULL;
20
21     return nuevoNodo;
22 }
23
24 void insertar(ABB &arbol, int x)
25 {
26     if(arbol == NULL)
27     {
28         arbol = crearNodo(x);
29     }
30     else if(x < arbol->nro)
31         insertar(arbol->izq, x);
32     else if(x > arbol->nro)
33         insertar(arbol->der, x);
34 }
35
36 void preOrden(ABB arbol)
37 {
38     if(arbol != NULL)
39     {
40         cout << arbol->nro << " ";
41         preOrden(arbol->izq);
42         preOrden(arbol->der);
43     }
44 }
45
46 void enOrden(ABB arbol)
47 {
```

```
C:\Users\HP\Downloads\FICHEROS+ARBOLES\FICHEROS+ARBOLES7.exe

.: [ ÁRBOL BINARIO DE BÚSQUEDA ]:.

N-mero de nodos del Árbol: 6

N-mero del nodo 1: 1
N-mero del nodo 2: 2
N-mero del nodo 3: 3
N-mero del nodo 4: 4
N-mero del nodo 5: 5
N-mero del nodo 6: 6

Mostrando ABB

      6
     / \
    5   6
   / \
  4   6
 / \
3   6
/ \
2   6
/ \
1   6

Recorridos del ABB:

En orden :
-----
Process exited after 9.111 seconds with return value 3221225477
Presione una tecla para continuar . . .
```