

UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas



“ÁRBOLES Y FICHEROS”

Estudiantes:

ESTEBAN RAMOS, Ana Cecilia

RAMOS VILLANUEVA, Miranda Nailea

ALVARADO URBANO, Saul Jose

Docente:

Haydee Raquel Sisa Yataco

Curso:

Estructura de Datos

Ciclo:

III

Tacna - Perú

2024

FICHEROS

EJERCICIO 1:

El siguiente programa en C++ tiene como objetivo crear y escribir contenido en un archivo de texto llamado datos.txt. Este archivo contiene tres líneas de texto predefinidas.

```
1  #include <fstream>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      ofstream archivo;
7
8      archivo.open("datos.txt");
9
10     archivo<<"Primera línea de texto"<<endl;
11     archivo<<"Segunda línea de texto"<<endl;
12     archivo<<"Ultima línea de texto"<<endl;
13
14     archivo.close();
15     return 0;
16
17 }
```

EJECUCIÓN:



datos: Bloc de notas

Archivo Edición Formato Ver Ayuda

Primera línea de texto
Segunda línea de texto
Ultima línea de texto

EJERCICIO 2:

El siguiente programa en C++ tiene como objetivo crear y escribir datos en un archivo de texto llamado Ejemplos.txt. Este archivo contendrá registros que consisten en un número entero, un número decimal (float) y una cadena de texto que representa el estado de evaluación (como "APROBADO", "SUSPENSO" y "NOTABLE").

```
1  #include<fstream>
2  #include<iostream>
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;
10         fichout<< 2 << " " << 1.1 << "SUSPENSO" <<endl;
11         fichout<< 3 << " " << 8.0 << "NOTABLE" <<endl;
12         fichout.close();
13     }
14 }
```

EJECUCIÓN:



*EJEMPLOS: Bloc de notas

Archivo Edición Formato Ver Ayuda

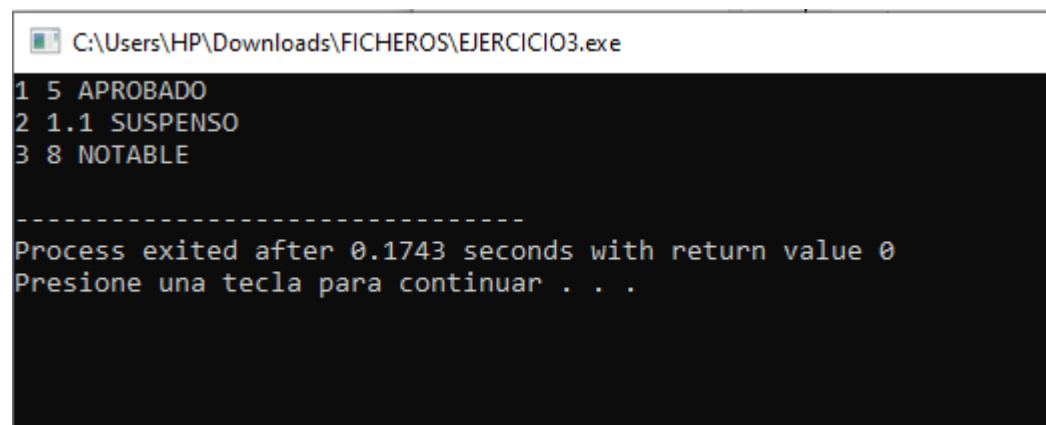
```
1 5APROBADO
2 1.1SUSPENSO
3 8NOTABLE
```

EJERCICIO 3:

El siguiente programa en C++ tiene como objetivo leer datos de un archivo de texto llamado EJEMPLOS.TXT Este archivo debe contener registros donde cada registro consiste en un número entero, un número decimal (float) y una cadena de caracteres (string). El programa lee estos datos y los imprime en la consola.

```
1  #include <fstream>
2  #include <iostream>
3  using namespace std;
4  typedef char TCadena[30];
5  int main()
6  {
7      int i;
8      float r;
9      TCadena cad;
10     ifstream fichin("EJEMPLOS.TXT");
11     if (!fichin)
12         cout << "\ Incapaz de crear o abrir el fichero ";
13     else{
14         fichin >> i;
15         while (!fichin.eof()){
16             cout << i << " ";
17             fichin >> r;
18             cout << r << " ";
19             fichin >> cad;
20             cout << cad << "\n";
21             fichin >> i;
22         }
23         fichin.close();
24     }
25 }
26 }
```

EJECUCIÓN:



```
C:\Users\HP\Downloads\FICHEROS\EJERCICIO3.exe
1 5 APROBADO
2 1.1 SUSPENSO
3 8 NOTABLE

-----
Process exited after 0.1743 seconds with return value 0
Presione una tecla para continuar . . .
```

EJERCICIO 4:

El siguiente programa en C tiene como objetivo permitir al usuario escribir frases que se guardarán en un archivo de texto llamado **registroDeUsuario.txt**. El programa continuará solicitando frases hasta que el usuario ingrese la palabra "fin", momento en el cual se detendrá y cerrará el archivo.

```
1  #include <stdio.h>
2  #include <string.h>
3  using namespace std;
4  int main()
5  {
6      FILE* ptFichero;
7      char fin[]="fin";
8      char frase[60];
9
10     ptFichero = fopen("registroDeUsuario.txt", "wt");
11     printf(" PROGRAMA para ESCRIBIR FRASES. \nCuando quiera salir,""escriba la palabra fin.\n\n");
12
13     do
14     {
15         puts("\nEscriba una FRASE:\n(o fin). \n");
16         gets(frase);
17         if (strcmp(frase, fin) == 0)
18             break;
19         fprintf(ptFichero, "%\n", frase);
20     }
21     while(strcmp(frase,fin)!= 0);
22
23     fclose(ptFichero);
24     return 0;
25 }
26 }
```

EJECUCIÓN:

C:\Users\HP\Downloads\FICHEROS\EJERCICIO4.exe

```
PROGRAMA para ESCRIBIR FRASES.
Cuando quiera salir,escriba la palabra fin.

Escriba una FRASE:
(o fin).

Hola mundo

Escriba una FRASE:
(o fin).

fin

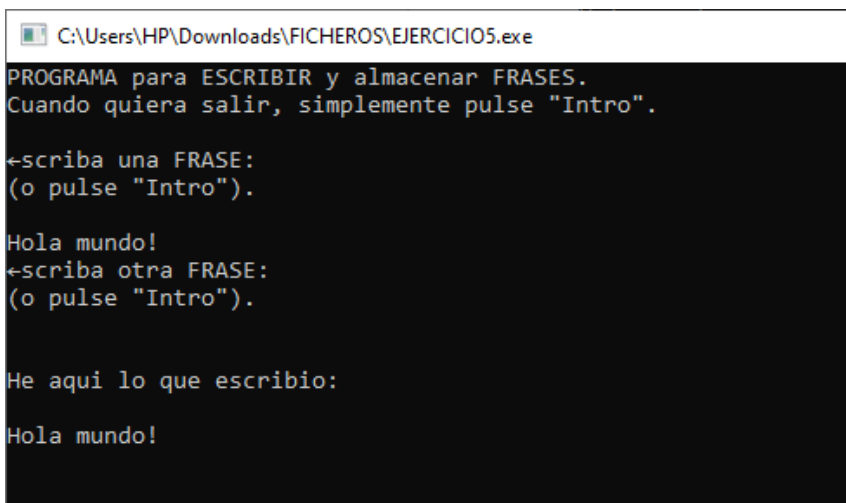
-----
Process exited after 79.71 seconds with return value 0
Presione una tecla para continuar . . .
```

EJERCICIO 5:

El siguiente programa en C tiene como objetivo permitir al usuario escribir frases que se almacenarán en un archivo de texto llamado `frases.txt`. El programa continuará solicitando frases hasta que el usuario presione la tecla "Intro" (sin ingresar texto), momento en el cual se detendrá, mostrará las frases ingresadas y cerrará el archivo.

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      FILE* ficheroU;
6      char frase [61];
7      int i=0;
8
9      ficheroU = fopen ("frases.txt", "wt");
10     printf("PROGRAMA para ESCRIBIR y almacenar FRASES. \n"
11            "Cuando quiera salir, simplemente pulse \"Intro\".\n\n");
12     do
13     {
14         if (i==0)
15             puts("\Escriba una FRASE:\n(o pulse \"Intro\"). \n");
16         else
17             puts("\Escriba otra FRASE:\n(o pulse \"Intro\"). \n");
18         gets(frase);
19         fprintf(ficheroU, "%s\n", frase);
20         i++;
21     }
22     while (strcmp(frase, "") !=0);
23     printf("He aqui lo que escribio:\n\n");
24     fclose(ficheroU);
25
26     ficheroU = fopen("frases.txt", "rt");
27     do
28     {
29         fgets (frase, 60, ficheroU);
30         puts(frase);
31     }
32     while (!feof(ficheroU));
33     getchar();
34     printf("...Hasta luego!");
35     getchar();
36     fclose(ficheroU);
37     return 0;
38
39 }
```

EJECUCIÓN:



```
C:\Users\HP\Downloads\FICHEROS\EJERCICIO5.exe
PROGRAMA para ESCRIBIR y almacenar FRASES.
Cuando quiera salir, simplemente pulse "Intro".

←scriba una FRASE:
(o pulse "Intro").

Hola mundo!
←scriba otra FRASE:
(o pulse "Intro").

He aqui lo que escribio:

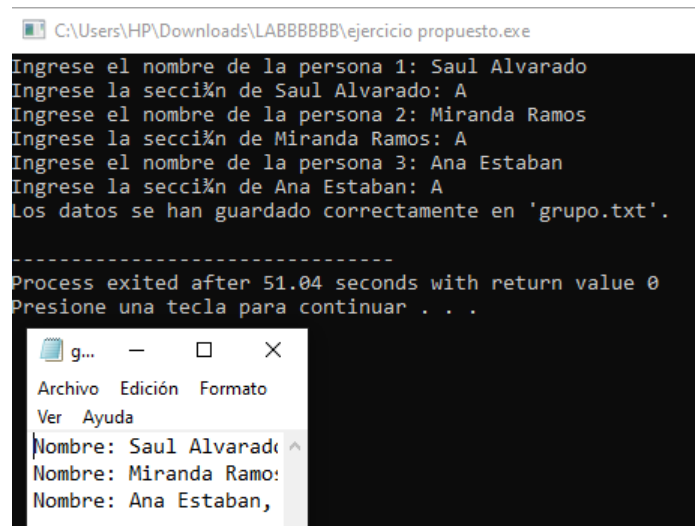
Hola mundo!
```

EJERCICIO PROPUESTO:

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

```
1  #include <fstream>
2  #include <iostream>
3  using namespace std;
4  typedef char TCadena[30];
5  int main()
6  {
7      int i;
8      float r;
9      TCadena cad;
10     ifstream fichin("EJEMPLO5.TXT");
11     if (!fichin)
12         cout << "\ Incapaz de crear o abrir el fichero ";
13     else{
14         fichin >> i;
15         while (!fichin.eof()){
16             cout << i << " ";
17             fichin >> r;
18             cout << r << " ";
19             fichin >> cad;
20             cout << cad << "\n";
21             fichin >> i;
22         }
23         fichin.close();
24     }
25 }
26 }
```

EJECUCIÓN:



```
C:\Users\HP\Downloads\LABBBBBB\ejercicio propuesto.exe
Ingrese el nombre de la persona 1: Saul Alvarado
Ingrese la secci n de Saul Alvarado: A
Ingrese el nombre de la persona 2: Miranda Ramos
Ingrese la secci n de Miranda Ramos: A
Ingrese el nombre de la persona 3: Ana Estaban
Ingrese la secci n de Ana Estaban: A
Los datos se han guardado correctamente en 'grupo.txt'.

-----
Process exited after 51.04 seconds with return value 0
Presione una tecla para continuar . . .

g...
Archivo Edici n Formato
Ver Ayuda
Nombre: Saul Alvarado
Nombre: Miranda Ramo:
Nombre: Ana Estaban,
```

EJERCICIO DE ÁRBOLES

Crear un programa c++ que implemente un Árbol Binario de Búsqueda (ABB). El programa debe permitir la inserción de nodos en un árbol binario y luego mostrar el árbol y sus recorridos en inorden, preorden y postorden

```
#include <iostream>
#include <stdlib.h>
using namespace std;
struct nodo{
    int nro;
    struct nodo *izq, *der;
};
typedef struct nodo *ABB;
/* es un puntero de tipo nodo que hemos llamado ABB, que utilizaremos
para mayor facilidad de creacion de variables */
ABB crearNodo(int x)
{
    ABB nuevoNodo = new(struct nodo);
    nuevoNodo->nro = x;
    nuevoNodo->izq = NULL;
    nuevoNodo->der = NULL;
    return nuevoNodo;
}
void insertar(ABB &arbol, int x)
{
    if(arbol==NULL)
    {
        arbol = crearNodo(x);
    }
    else if(x < arbol->nro)
        insertar(arbol->izq, x);
    else if(x > arbol->nro)
        insertar(arbol->der, x);
}
void preOrden(ABB arbol)
{
    if(arbol!=NULL)
    {
        cout << arbol->nro << " ";
        preOrden(arbol->izq);
        preOrden(arbol->der);
    }
}
void enOrden(ABB arbol)
{
    if(arbol!=NULL)
    {
        enOrden(arbol->izq);
        cout << arbol->nro << " ";
        enOrden(arbol->der);
    }
}
void postOrden(ABB arbol)
{
    if(arbol!=NULL)
    {
        postOrden(arbol->izq);
        postOrden(arbol->der);
        cout << arbol->nro << " ";
    }
}
void verArbol(ABB arbol, int n)
{
    if(arbol==NULL)
        return;
    verArbol(arbol->der, n+1);
    for(int i=0; i<n; i++)
        cout<<" ";
    cout<< arbol->nro <<endl;
}
```



```

        verArbol(arbol->izq, n+1);
    }
}
int main()
{
    ABB arbol = NULL; // creado Arbol
    int n; // numero de nodos del arbol
    int x; // elemento a insertar en cada nodo
    cout << "\n\t\t ..[ ARBOL BINARIO DE BUSQUEDA ].. \n\n";
    cout << " Numero de nodos del arbol: ";
    cin >> n;
    cout << endl;

    for(int i=0; i<n; i++)
    {
        cout << " Numero del nodo " << i+1 << ": ";
        cin >> x;
        insertar( arbol, x);
    }

    cout << "\n Mostrando ABB \n\n";
    verArbol( arbol, 0);
    cout << "\n Recorridos del ABB";
    cout << "\n\n En orden : "; enOrden(arbol);
    cout << "\n\n Pre Orden : "; preOrden(arbol);
    cout << "\n\n Post Orden : "; postOrden(arbol);
    cout << endl << endl;
    system("pause");
    return 0;
}

```

FASE DE EJECUCIÓN:

```

        ..[ ARBOL BINARIO DE BUSQUEDA ]..

Numero de nodos del arbol: 10

Numero del nodo 1: 5
Numero del nodo 2: 10
Numero del nodo 3: 20
Numero del nodo 4: 30
Numero del nodo 5: 12
Numero del nodo 6: 22
Numero del nodo 7: 16
Numero del nodo 8: 18
Numero del nodo 9: 23
Numero del nodo 10: 21

Mostrando ABB

    30
   23
  22
 21
20
 18
 16
 12
10
5

Recorridos del ABB

En orden : 5 10 12 16 18 20 21 22 23 30

Pre Orden : 5 10 20 12 16 18 30 22 21 23

Post Orden : 18 16 12 21 23 22 30 20 10 5

Presione una tecla para continuar . . .

```