

Ficheros: texto y binarios

Fundamentos de Programación
Fundamentos de Programación I

Trabajar con ficheros texto

- No olvidar incluir la cabecera `#include <fstream.h>`
- Clase para definición de un objeto fichero de escritura `ofstream fichero;`
- Clase para definición de un objeto fichero de lectura `ifstream fichero;`
- Apertura del fichero: `fichero.open("nombrefichero");`
- Cierre del fichero: `fichero.close();`
- Escritura y lectura. La lectura y escritura de información se realiza a través los operadores `<<` y `>>` (de forma equivalente al uso de `cin` y `cout`):

Nota: por defecto el Dev C++ deja el fichero creado en el mismo directorio donde está el compilador.

```
fichero << dato; //escritura en fichero
fichero >> dato; // lectura desde fichero
```

Trabajar con ficheros binarios

- Leer fichero binario

```
f.read((char *) (& dato), sizeof(dato))
```

Número de bytes a leer

Posición inicial de memoria,

- Escribir fichero binario

```
f.write((char *) (& dato), sizeof(dato) );
```

Número de bytes a escribir

Posición donde se encuentra el dato a escribir

Trabajar con ficheros texto

Realiza un programa que calcule la media de un conjunto de números reales positivos almacenados en un fichero. Los datos leídos se guardan en un fichero después de haberles restado la media. También se guarda la media.

```

//Programa que lee y escribe vectores desde fichero (texto) y
//calcula su media

#include <iostream.h>
#include <stdlib.h>
#include <fstream.h>
#include <string>

const int MAXIMO = 200;
typedef float Vector[MAXIMO];

//Prototipos
int F_LeerVector(ifstream &f, Vector v);
void F_EscribirVector(ofstream &f, Vector v, int tam);
void MostrarVector(Vector v, int tam);
void RestarMediaVector(Vector v, int tam, float &media);

//Leer Vector del fichero
int F_LeerVector(ifstream &f, Vector v)
{
    int i = 0;

    while( f >> v[i] && i < MAXIMO )
        i++;

    return i;
}
//Escribir Vector en fichero
void F_EscribirVector(ofstream &f, Vector v, int tam)
{
    int i;

    for (i = 0; i < tam ; i++)
        f << v[i] << ' ';

    return;
}

```

```

//Mostrar Vector por pantalla
void MostrarVector(Vector v, int tam)
{
    int i;

    cout << "El vector es:\n";
    for (i = 0; i < tam ; i++)
        cout << v[i] << ' ';

    return;
}
//Resto la media al vector (modifica los elementos del vector)
void RestarMediaVector(Vector v, int tam, float &media)
{
    int i;

    //Calculo la media
    media = 0;
    for ( i = 0; i < tam ; i++)
        media += v[i];
    media = media / tam;
    //Modifico vector
    for ( i = 0; i < tam ; i++)
        v[i] = v[i] - media;

    return;
}

```

```

int main()
{
    Vector v;
    int tam;
    ifstream f;
    ofstream fout;
    string nombrefichero;

    cout << " Este programa lee un vector de fichero\n";
    cout << "Introduce el nombre del fichero\n";
    cin >> nombrefichero;

    //Abrir fichero texto
    f.open(nombrefichero.c_str());
    if(!f)
        cout << "Error abriendo fichero\n";
    else
    {
        tam = F_LeerVector(f,v);
        f.close();
        MostrarVector(v,tam);
        CalcularMediaVector(v,media);
        cout << "Nombre de fichero para guardar resultado:\n";
        cin >> nombrefichero;
        //Abrir fichero
        fout.open(nombrefichero.c_str());
        if ( !fout)
            cout << "Error abriendo fichero de resultados:\n";
        else
        {
            F_EscribirVector(fout,v,tam);
            fout.close();
        }
    }

    return 0;
}

```

Trabajar con ficheros binarios

Realiza un programa que calcule la media de un conjunto de números reales positivos almacenados en un fichero. Todos los datos se guardan en un fichero binario junto con la media calculada

```

//Programa que lee y escribe vectores desde fichero y
//calcula la media (ficheros binarios)

#include <iostream.h>
#include <stdlib.h>
#include <fstream.h>
#include <string>

const int MAXIMO=200;
typedef float Vector[MAXIMO];

int F_LeerVectorBinario(ifstream &f, Vector v);
void F_EscribirVectorBinario(ofstream &f, Vector v, int tam);
void MostrarVector(Vector v, int tam);
void ProcesarDatos(Vector v, int tam, float &media)

int F_LeerVectorBinario(ifstream &f, Vector v)
{
    int i=0;

    while( f.read((char *)(&v[i]),sizeof(v[i])) && i < MAXIMO)
    {
        i++;
    }

    return(i);
}
void F_EscribirVectorBinario(ofstream &f, Vector v, int tam)
{
    int i;

    for (i = 0; i < tam ; i++)
        f.write((char *)(&v[i]),sizeof(v[i]));
    return;
}
}

```

```

void MostrarVector(Vector v, int tam)
{
    int i;

    cout << "El vector es:\n";
    for (i = 0; i < tam ; i++)
        cout << v[i] << ' ';

    return;
}
void ProcesarDatos(Vector v, int tam, float &media)
{
    int i;

    media = 0;
    for (i = 0; i < tam ; i++)
        media += v[i];
    media = media / tam;
    return;
}
//Funciön que lee datos desde teclado y los guarda en un vector
int LeerDatos(Vector v)
{
    bool fin=false;
    int i=0;
    float a;

    do
    {
        cin >> a;
        if ( a < 0 || (i >= MAXIMO))
            fin = true;
        else
        {
            v[i] = a;
            i++;
        }
    }while(fin != true);

    return i;
}

```

```

int main()
{
    Vector v;

    int tam;
    int opcion;
    ifstream f;
    ofstream fout;
    float media;
    string nombrefichero;

    cout << " Este programa lee un vector de...\n";
    cout << "Introduce opcion a realizar\n";
    cout << "0. Leer datos desde teclado" << endl;
    cout << "1. Leer desde fichero" << endl;
    cin >> opcion;
    switch(opcion)
    {
        case 0:
            cout << "Introduce los elementos del
vector";
            cout << "Numero negativo para terminar";
            tam=LeerDatos(v);
            MostrarVector(v, tam);
            break;

        case 1:
            cout << "Nombre del fichero a leer\n";
            cin >> nombrefichero;
            f.open(nombrefichero.c_str());
            if (!f)
                cout << "Error abriendo fichero\n";
            else {
                tam=F_LeerVectorBinario(f,v);
                MostrarVector(v,tam);
            }
            break;

        default: cout << "opcion no valida";
    }
}

```

```

ProcesarDatos(v,tam,media);

endl; cout << "La media calculada es: " << media <<
cout << "Nombre de fichero de salida\n";
cin >> nombrefichero;
fout.open(nombrefichero.c_str());
if (!fout)
    cout << "Error creando fichero\n";
else
{
    F_EscribirVectorBinario(fout,v,tam);
    fout.close();
}
system("PAUSE");
return 0;
}

```

Calculo de calificaciones escolares

Escribe un programa para calcular la nota de un grupo escolar que siga la siguiente política de calificación:
 Hay 2 cuestionarios, cada uno de ellos se califica sobre 10 puntos.
 Hay un examen parcial y un examen final, cada uno de los cuales se califica sobre 100 puntos.
 El examen final representa el 50% de la nota, el parcial el 25% y los 2 cuestionarios juntos el 25% restante.

Las notas se representan de la siguiente forma:
 Sobresaliente corresponde a una nota entre [90,100].
 Notable corresponde a una nota promedio entre [70,90].
 Aprobado entre [50,70].
 Suspendido una nota promedio inferior a 50.

Define y emplea una estructura para almacenar la información de cada estudiante.
 El programa leera los datos almacenados en un fichero y el resultado lo escribirá en otro fichero con el mismo formato.

Ejemplo de fichero de entrada con las notas:

```

Pedro Lopez
10 10 100 100
Juan Martinez
9 9 90 90
Elena Ibañez
8 8 80 80

```

Ejemplo de fichero salida con la nota global calculada:

```

Pedro Lopez
10 10 100 100 Sobresaliente
Juan Martinez
9 9 90 90 Sobresaliente
Elena Ibañez
8 8 80 80 Notable

```

```

//Programa que calcula calificaciones de alumnos
#include <iostream.h>
#include <stdlib.h>
#include <string>
#include <fstream.h>

const int MAXIMO = 100 ;
//Definición de estructuras
//Definición de tipos

struct Alumno
{
    string n_alumno;
    int test1, test2;
    int parcial, final;
    string notaglobal;
};

typedef Alumno Curso[MAXIMO];

//Prototipos de funciones
bool LeerRegistro(ifstream& in_f, Alumno & alu);
void MostrarRegistro(ofstream& out_f, Alumno alu);
void CalcularNotaRegistro(Alumno & alu);

```

```

int main()
{
    //Declaro un vector para almacenar la información de los alumnos
    Curso micurso;
    int i, elementos;
    ifstream ent_f;
    ofstream sal_f;
    string nombre_in, nombre_out;

    cout << "Introduce mi nombre de fichero con los
    datos\n";
    //Leo nombre y almaceno en una variable tipo 'string'
    cin >> nombre_in;
    cout << nombre_in;
    ent_f.open( nombre_in.c_str() );
    if (!ent_f)
        cout << "Error abriendo fichero de datos."<< endl;
    else
    {
        i = 0;
        //Leo registros almacenados en el fichero
        while(LeerRegistro(ent_f, micurso[i]))
        {
            CalcularNotaRegistro(micurso[i]);
            i++;
        }
        elementos = i;
        ent_f.close(); //Cierro el fichero
        //Fichero donde guardar los resultados
        cout << "Introduce el nombre del fichero donde
        guardar el resultado\n";
        cin >> nombre_out;
        //Convierto a cadena de caracteres tipo C
        sal_f.open( nombre_out.c_str() );
        if ( !sal_f)
            cout << "Error abriendo fichero\n";
        else
            for (i = 0; i < elementos ; i++)
                MostrarRegistro(sal_f, micurso[i]);
            sal_f.close(); //Cierro el fichero
        }
    }
    return 0;
}

```

```

bool LeerRegistro(ifstream& in_f, Alumno &alu)
{
    //Leo cada elemento de la estructura de forma independiente

    //Leo el nombre
    getline(in_f, alu.n_alumno);
    //Leo las notas
    in_f >> alu.test1;
    in_f >> alu.test2;
    in_f >> alu.parcial;
    in_f >> alu.final;
    //Para saltar el salto de linea que queda en el buffer
    in_f.ignore();
    //Devuelvo true si todo ha ido bien, false en caso contrario
    return !(in_f.eof());
}

//Escribo los datos almacenados en el registro a un archivo
void MostrarRegistro(ofstream& out_f, Alumno alu)
{
    //Escribo el resultado en el fichero asociado a out_f
    out_f << alu.n_alumno << endl;
    out_f << alu.test1 << " " << alu.test2 << " " << alu.parcial << " " << alu.final << " "
    << alu.notaglobal << endl;
    return;
}

//Calculo la nota final
void CalcularNotaRegistro(Alumno & alu)
{
    float notanumerica;
    notanumerica = (alu.test1 + alu.test2) * 25 / 20 +
        (alu.final / 2) + (alu.parcial / 4);
    if (notanumerica > 90.0)
        alu.notaglobal= "Sobresaliente";
    else if (notanumerica > 70.0)
        alu.notaglobal= "Notable";
    else if (notanumerica > 50.0 )
        alu.notaglobal="Aprobado";
    else
        alu.notaglobal="Suspendido";
    return;
}

```

Devuelve true si puede leer todos los campos del registro correctamente