

**Objetivo de la práctica:**

Practicar uso de ficheros: abrir, cerrar y tratamiento de información contenida en el fichero

**Uso de Ficheros**

1. No olvidar incluir la cabecera: `#include <fstream.h>`
2. Los ficheros se pueden abrir para leer su contenido o para escribir en ellos. Nunca leeremos y escribiremos simultáneamente (nota: *fichero* es solamente el nombre de la variable):
 

Clase para definición de un objeto fichero de escritura:	<code>ofstream fichero;</code>
Clase para definición de un objeto fichero de lectura:	<code>ifstream fichero;</code>
Apertura del fichero:	<code>fichero.open ("nombre_fichero");</code>
Cierre del fichero:	<code>fichero.close ();</code>

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

```
fichero.open("c:\\tmp\\hola.txt"); // deja en fichero hola.txt en el directorio c:\tmp (notad la doble \)
```

3. 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`):

```
fichero << dato; //donde dato es cualquier tipo de dato convertido a carácter. Escritura en fichero
fichero >> dato; // lectura desde fichero
```

4. Fin de fichero. Para leer un fichero hasta su fin se empleará:

```
while (fichero >> dato) // esta condición proporcionará FALSE cuando se llegue al final.
```

5. Estructuras. No se puede leer ni almacenar una estructura directamente en un fichero, sino que debe procederse por separado con cada uno de sus campos:

```
ejemplo: struct complejo
{
    int real;
    int imag;
}
```

```
Bien:   salidafich << complejo.real << " " << complejo.imag;
Mal:    salidafich << complejo
```



## PROBLEMAS

(\*)1. Hacer un programa que permita leer y escribir matrices de tamaño variable (máximo 10x10) de un fichero texto. El formato de este fichero es el siguiente:

- ✓ primera línea: un entero que representa el número de filas de la matriz
- ✓ segunda línea: un entero que representa el número de columnas de la matriz
- ✓ siguientes líneas: una fila de la matriz en cada línea, donde cada elemento está separado del siguiente por un espacio

**Ejemplo de matriz1.dat:**

```
3
2
4 -34
4 2
0 7
```

Ese programa incluirá una función que permita visualizar la información leída en forma de matriz de la siguiente forma:

```
4 -34
4 2
0 7
```

(\*)2. Hacer un programa similar al anterior pero en este caso los ficheros con los que se trabajará serán ficheros binarios.

3. Hacer un programa que lea de 2 ficheros “matriz1.dat” y “matriz2.dat” (2 matrices con el formato del ejercicio 1) y después de sumarlas (si es posible) muestre el resultado y lo deje en otro fichero “matriz3.dat”, con el mismo formato que las anteriores matrices.

(\*\*)4. Hacer un programa que cree un fichero de alumnos, que contendrá la siguiente información: nombre, primer apellido y nota. El programa deberá permitir desde un menú, las siguientes opciones:

- 1.- Ver todo el fichero
- 2.- Ver alumnos aprobados
- 3.- Añadir alumno
- 4.- Calcular la nota media
- 5.- Salir