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

NOTA 1: Durante la práctica todos los ejercicios deberán ser guardados ***temporalmente*** *en el directorio \tmp. Una vez finalizada la misma y transferidos los ficheros a un disquete, se deberá eliminar dicho directorio.*

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:	`ofstream fichero;`
Clase para definición de un objeto fichero de lectura:	`ifstream fichero;`
Apertura del fichero:	`fichero.open ("nombre_fichero");`
Cierre del fichero:	`fichero.close ();`

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. Escribe un programa que lea desde teclado la información correspondiente a cada alumno de la asignatura de "Fundamentos de Programación" y escriba dicha información en un fichero de texto y en un fichero binario. Esta información es el identificador del alumno, la nota de prácticas y la nota del examen. Comprueba las diferencias entre los dos ficheros (tamaño, formato, ...)

2. Supongamos que tenemos dos ficheros binarios productos1.dat y productos2.dat que contienen un vector de registros de productos ordenadas por el número de referencia del producto. Cada registro tiene asociada una estructura como esta:

```
struct producto
{
int referencia;
float cantidad;
};
typedef producto catálogo[MAX_PROD];
```

   Escribe una función para realizar cada una de las siguientes tareas:

   2.1. Una función que a partir de los dos ficheros originales escriba un nuevo fichero binario que contenga una única lista de productos también ordenada por el número de referencia del producto.

   2.2. Una función que escriba en un nuevo fichero binario un listado de aquellos 10 productos del catálogo que más existencias de cada producto tienen.

   2.3. Una función que escriba en un nuevo fichero binario un listado de aquellos elementos que están por arriba de la media de existencias del catálogo de productos.

3. Escribe una función que simule la función "reemplazar" que disponen los editores de texto. A dicha función le pasamos el nombre del a fichero donde se encuentra el texto, la cadena que queremos buscar, y la cadena que utilizamos para reemplazar. La función nos debe regresar además el número total de reemplazos realizados en tal texto.

4. Una empresa de plásticos piensa que la empresa competidora está espiándole interceptando los ficheros de información que se transmiten mediando la red. Esta empresa nos pide que le desarrollemos un programa de encriptación para poder enviar los ficheros con seguridad por la red, y poder descifrarlos una vez recibidos. Así un fichero de texto antes de ser enviado se deberá encriptar con una clave que solo sea conocida por los trabajadores de la empresa.

   El sistema de encriptación que nos proponen consiste al sumar el código ASCII de cada carácter al código ASCII de la clave. El carácter correspondiente al código ASCII de la suma es el carácter encriptado. Queremos que los código encriptados que generemos sean imprimibles por lo tanto las caracteres encriptados deben estar entre 32 y 127. Aquellos caracteres que correspondan a un código ASCII superior a 127 simplemente no lo codificaremos (por ejemplo la ñ, ç, o, etc.)

   Ejemplo:

La clave está en Rebeca	Frase original
rebecarebecarebecarebe	Clave repetida
lb"fpfvf"hwyá!gq$wecgfe	Frase encriptada