



## Programación

*Ingeniería Técnica Industrial*  
*Especialidades: Electricidad y Mecánica*

### Ejercicios del tema 3: Archivos

- 1.- Escribir un programa que lea valores de un fichero de vectores de 10 números enteros e imprima por pantalla la media de cada vector. (El fichero se ha creado utilizando `fprintf`).
- 2.- Escribir un programa que copie en un fichero otros dos ficheros, uno a continuación de otro.
- 3.- Elaborar un programa que muestre por pantalla un fichero de texto cualquiera.
- 4.- Dado un fichero de texto, escribir un programa que cuente el número de palabras que aparezcan en dicho fichero. Considerar como una palabra cualquier sucesión de caracteres separada por blancos o marca de fin de línea.

Ejemplo: (`\n` significa fin de línea)

```
Este          es          un          ejemplo para\n
probar\n
\n
  el funcionamiento de un programa \n
  \n
que cuente las palabras\n
```

deberá responder 15 palabras.

- 5.- Tenemos un fichero de registros cuyas componentes están definidas según la declaración siguiente:

```
struct registro1{
    char partido[60];
    char localidad[60];
    int candidatos;
}
```

Se quiere sustituir la estructura del registro por el siguiente:

```
struct registro1{
    char partido[60];
    char localidad[60];
    int candidatos,elegidos;
}
```



Escribir un programa que modifique la estructura y trasvase toda la información del fichero primitivo al nuevo, inicializando a 0 el nuevo campo creado.

- 6.- Dada una palabra, diremos que está “escondida” en un texto si está contenida en éste, no necesariamente de forma continua. Por ejemplo, la palabra “elefante” está escondida en la frase “El hermano de Francisco tuvo suerte”.

Elabórese un programa tal que, dada una palabra leída por teclado, la busque en un texto almacenado en un fichero de texto `texto.txt`. Como resultado, el programa debe exponer el mensaje:

```
Palabra <palabra> no encontrada en el texto
```

o bien

```
Palabra <palabra> encontrada escondida entre los <p> primeros caracteres del texto
```

En el ejemplo, el resultado debería ser:

```
Palabra elefante encontrada escondida entre los 32 primeros caracteres del texto
```

Téngase en cuenta la equivalencia entre mayúsculas y minúsculas. Supondremos que la palabra a buscar no tiene más de 20 caracteres, pero desconocemos su longitud.

- 7.- En un archivo de texto `glosario.dat` se encuentran una serie de abreviaturas en orden alfabético seguidas por su descripción en la siguiente línea.

Diseñar y elaborar un programa que dándole una abreviatura del fichero conteste con su descripción completa. Si la abreviatura pedida no apareciese en el fichero, el programa indicará la abreviatura anterior y posterior.

**Nota:** Las abreviaturas tienen una longitud máxima de 10 caracteres, pero desconocemos la longitud máxima de las descripciones.

- 8.- Dado un valor entero por teclado, elaborar un programa que busque en un fichero (`numeros.dat`) de enteros ese valor, y en caso de encontrarlo, que diga en qué posición o posiciones del fichero se encuentra. El fichero de enteros ha sido grabado haciendo uso de la función `fwrite()`, por lo que se hace necesario recuperar los valores elemento a elemento haciendo uso de la función `fread()`.

- 9.- Se define el tipo siguiente:

```
struct alumno {
    char apellidos[60];
    char nombre[20];
    float notajunio, notasept, notaprac;
}
```



En un fichero con datos de esta estructura disponemos de los alumnos matriculados en una asignatura. Se quiere obtener un fichero de texto en el que aparezca en cada línea el nombre de un alumno con el formato apellidos, nombre, nota junio. Hacer un programa que realice esta tarea.

- 10.-** En un fichero se tiene grabada una sucesión de números enteros positivos o nulos, correspondientes a las puntuaciones de varios jueces sobre un mismo ejercicio.
- Escribir un algoritmo para calcular la media aritmética de los valores estrictamente positivos, excepto el máximo y el mínimo de dichos valores positivos (suponiendo que sólo hay un máximo y un mínimo). También debe calcular el número de ceros.
  - Variar el algoritmo para el caso en el que los valores máximo y mínimo puedan aparecer varias veces.