# Ej1- Análisis palabras fichero

Realiza un programa que analice un fichero en búsqueda de palabras y vuelque el resultado en otro fichero.

La llamada al programa se debe realizar con los siguientes parámetros:

```
./analiza.exe <Nombre de fichero> <Palabras a buscar>
```

Condiciones del programa:

- El parámetro <palabras a buscar> es una secuencia de palabras separadas por espacio, de un número indeterminado. Usar las propiedades de argc/argv para almacenarlas correctamente.
- El fichero de salida tendrá el nombre del fichero de entrada añadiendo ".out" antes de la extensión. Si el nombre no tiene punto, se añadirá al final del nombre. (Ayuda, si no se consigue crear el nombre fichero o no se puede abrir, se volcará a la salida estándar). Por ejemplo:
    - El nombre del fichero de salida de lineas.txt sería lineas.out.txt
- Los errores se deberán dirigir a la salida de error (fichero ya abierto en programa stderror).
- Para cada palabra encontrada, el programa escribirá en el archivo de resultados:
    - La palabra
    - El número de ocurrencias
    - Números de las líneas en las que se encuentra la palabra.
- Se deberá utilizar memoria dinámica tanto para leer las líneas del fichero como para almacenar las estructuras y la información que sea necesaria de estas estructuras. No hay que almacenar el contenido de todo el fichero, solo la información relevante.
- El programa deberá liberar la memoria dinámica cuando ya no la necesite.

Se pide utilizar las siguientes dos estructuras:

Estructura para almacenar la **información de las palabras** que se van a buscar (`palabraInfo_t` ):

Palabra

Número de ocurrencias (inicialmente, 0)

Lista de números de línea (array dinámico, inicialmente tamaño 0)

Estructura para almacenar la **lista de información (**`listaPalabras_t`**):**

Lista de palabras con su información (array con la estructura descrita anteriormente)

Número de palabras (tamaño útil del array almacenado en esta estructura)

Se pide utilizar las siguientes funciones:

```
//Esta función recibe una línea y el número de línea y modifica la lista de palabras para añadir la información encontrada.
```

```
void examinaLinea(char *lineaAux, int numLinea, listaPalabras_t
listaPalabras)

//Esta función obtiene la siguiente línea de un fichero ya abierto y devuelve
//un puntero a la cadena con memoria reservada dinámicamente. Utiliza fgetc

char *leeLineaDinamicaFile(FILE *fd)


//Abre el fichero de escritura obteniendo el nombre a partir del nombre del
//fichero origen y escribe para cada palabra encontrada: la palabra, el
//número de ocurrencias y las líneas en las que aparece, si una palabra
//aparece varias veces en una línea aparecerá duplicada en la lista

void escribirResultado(listaPalabras_t lista, char *nombreFichOrigen)
```

Ejemplo de ejecución:

```
./AnalisisFichero buscar.txt que palabras noestan
```

Fichero de entrada (buscar.txt)

```
Aqui un fichero que podria ser mas largo pero que no lo es
con distintas lineas
algunas mas cortas que otras
y distintas palabras para poder buscar coincidencias
podemos alargarlo pero creo que esta claro
```

Fichero de salida (buscar.out.txt)

```
que. Numero de ocurrencias: 4. Lineas: 1 1 3 5
palabras. Numero de ocurrencias: 1. Lineas: 4
```