

El cuadro de abajo a la izquierda muestra el contenido de un archivo que representa un diccionario no ordenado. El cuadro de la derecha es el mismo diccionario en donde se ha *eliminado* la palabra “*alimento*”.

casa	edificacion construida para ser habitada	casa	edificacion construida para ser habitada
<b>alimento</b>	sustancia ingerida por un ser vivo	<b>radio</b>	receptor de ondas electromagneticas
celular	aparato portatil de un sistema de telefonía celular	celular	aparato portatil de un sistema de telefonía celular
gato	felido domestico	gato	felido domestico
<b>radio</b>	receptor de ondas electromagneticas		

Observe que la definición de radio se movió a la línea que ocupaba la definición de alimento. La definición de radio se reemplazó por una línea en blanco. Cada línea del archivo es de 80 caracteres y almacena una definición. Los primeros 10 caracteres corresponden a la palabra definida, luego vienen 69 caracteres para su definición y se termina la línea con un `\n` para completar los 80 caracteres. El archivo podría contener 0 o más líneas en blanco al final (y solo al final).

Escriba el programa *eliminar.c* de manera que se invoque por ejemplo:

```
$ ./eliminar dict.txt alimento
```

Este programa debe eliminar la palabra *alimento* del diccionario almacenado en el archivo de nombre *dict.txt* dejando el diccionario exactamente como se muestra en el cuadro de arriba a la derecha.

**Metodología obligatoria:** Ud. debe programar la función *main* con encabezado: `int main(int argc, char *argv[])`. Verifique que el número de parámetros es el correcto, es decir *argc* debe ser 3. Si no, termine el programa con código de retorno 1 (el valor que retorna la función *main*) y escriba el modo de uso del comando en la salida estándar de errores con `fprintf(stderr, ...)` exactamente como se indica a continuación:

```
$ ./eliminar dict.txt
Uso: ./eliminar <archivo> <palabra>
$ echo $?
1
$
```

El nombre del archivo viene en `argv[1]` y la palabra en `argv[2]`. Si no puede abrir el archivo debe reportar el error usando la función estándar *perror*. Además el código de retorno del programa debe ser el que entrega la variable estándar *errno*. Por ejemplo si *duct.txt* no existe, la ejecución debe reportar:

```
$ ./eliminar duct.txt gato
duct.txt: No such file or directory
$ echo $?
2
$
```

El mensaje podría estar en español en instalaciones de Linux en español.

El archivo completo no cabe en la memoria. Solo puede almacenar en memoria un par de líneas. Lea secuencialmente el archivo con *fread* buscando las líneas que contienen la palabra y la última definición en el archivo (las líneas en blanco del final no son definiciones). Use *fseek* y *fwrite* para reescribir la línea que contenía la palabra que se elimina con una copia de la última definición. Use *fseek* y *fwrite* nuevamente para reemplazar la línea que contenía la última definición con espacios en blanco. Tenga cuidado en considerar que `strcmp("casa", "casa ")` es distinto de 0. El tamaño y número de líneas del archivo no cambia.

### Instrucciones

Baje *t4.zip* de U-cursos y descomprímalo. El directorio *T4* contiene el *Makefile* que le servirá para compilar el archivo *eliminar.c* en donde Ud. debe programar su solución. El archivo *eliminar.c.plantilla* es un molde para su solución. Otros archivos se usan para validar que su tarea funciona correctamente.

Pruebe su tarea bajo Debian 10 con los comandos:

- `make test-g`
- `make test-O`
- `make test-vg`

La ejecución de cada uno de ellos debe reportar “*Felicitaciones: todos los tests fueron aprobados*”. El comando `make test-vg` prueba su tarea con `valgrind`. No debe reportar errores ni poseer sección LEAK SUMMARY. Si no cumple estos requisitos no obtendrá puntaje. Si `valgrind` detecta errores, ejecute `make test-vg-ddd` para recibir instrucciones sobre cómo usar `ddd` para depurar los errores reportados.

Se descontará medio punto si alguna de las compilaciones arroja algún warning.

### Entrega

Ud. solo debe entregar el archivo *eliminar.c* por medio de U-cursos. Se descontará medio punto por día de atraso. No se consideran los días de vacaciones, sábado, domingo o festivos.