

Sistemas Operativos

[Práctica 1 – PROGRAMACIÓN C]

LUCÍA DOMÍNGUEZ RODRIGO, MARCOS JIMÉNEZ PULIDO

Tabla de contenido

[Autores 2](#_Toc115812632)

[Descripción del Código 3](#_Toc115812633)

[Diseño del Código 3](#_Toc115812634)

[Principales Funciones 3](#_Toc115812635)

[Casos de Prueba 3](#_Toc115812636)

[Comentarios Personales 4](#_Toc115812637)

# Autores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Apellidos | Grado | Correo Institucional | Usuario Github |
| Lucía | Domínguez Rodrigo | GII + GIS | l.dominguez.2021@alumnos.urjc.es | [LuciaDominguezRodrigo](https://github.com/LuciaDominguezRodrigo) |
| Marcos | Jiménez Pulido | GII + GIS | m.jimenezp.2021@alumnos.urjc.es | [MarJ03](https://github.com/MarJ03) |
| Repositorio utilizado para la práctica | | | [CLibrary](https://github.com/LuciaDominguezRodrigo/CLibrary) | |
| Carpeta Drive con archivos de prueba | | | [Archivos de prueba](https://drive.google.com/drive/folders/17-Aqjk7uBnhNgdPc_fL4sY8LMFemg0wN?usp=drive_link) | |

# Descripción del Código

## Diseño del Código

De manera general, se ha implementado el código utilizando si es posible las funciones y librerías predeterminadas C explicadas en clase. En nuestro caso, contamos con las tres funciones principales explicadas en la práctica, y con dos funciones auxiliares para poder representar de una manera más modular y limpia el código de la función ***longlines***. Se ha definido una constante global que indica el tamaño total que puede tener una línea, ya que era utilizado en muchas partes del código (referenciado en el mismo como MAX\_LINE\_LENGTH, que en este caso se corresponderá con 1024 bytes).

***Función Head***

Para imprimir por salida estándar las primeras N líneas, primeramente se inicializa un contador a 0, y mientras ese contador no supere N, se va llenando y sobrescribiendo un buffer con el tamaño que indique MAX\_LINE\_LENGTH, el cual se va leyendo y mostrando en la salida estándar, incrementando el contador y sobrescribiendo su contenido en cada iteración.

La función devolverá un valor numérico 0 si la ejecución ha sido correcta, y un valor numérico 1 si ha habido algún error (como por ejemplo si ha habido algún error de lectura).

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

***Función Tail***

Para la implementación de esta función, primeramente se inicializa un buffer dinámico que contiene datos de tipo char\* (que apuntarán a las frases elegidas para imprimir por salida estándar), devolviendo un error numérico con valor 1 en caso de que la asignación de memoria sea errónea. Tras esto, se asigna memoria dinámica para cada línea apuntada por cada posición del buffer, es decir, las cadenas de caracteres donde irán alojadas las frases elegidas, imprimiendo un error numérico con valor 2 si alguna de las asignaciones no es correcta.

A continuación, se procede a la lectura de las líneas, inicializando un contador a 0 y aumentando su valor en 1 en cada línea correctamente leída. Cada una de las líneas leídas se almacenará en una posición del buffer, dependiendo del orden de llegada, empezando a sobrescribir las primeras líneas almacenadas en el buffer de forma ascendente cada vez que este se encuentra lleno. De esta forma, sólo se usa el tamaño de memoria requerido para mostrar las N líneas necesarias, independientemente del tamaño de la entrada.

**Diagrama

Descripción generada automáticamente**

current\_line apuntará en cada lectura a la siguiente posición de lines\_buffer, comenzando por la 0 y escribiendo en la posición line\_count mod N, siendo line\_count el número de la línea leída por la entrada estándar en esa iteración (por ejemplo, si N = 7 y se lee la 9ª línea de la entrada estándar, entonces line\_count valdrá 9 y se sobrescribirá en la cadena de caracteres apuntada por el puntero situado en la posición 9 mod 7 = 2 de lines\_buffer).

Tabla

Descripción generada automáticamente

Siguiendo con esta idea, podría darse que la última línea leída por la entrada estándar no esté apuntada por el puntero en la última posición de lines\_buffer. Para localizar la primera de las últimas N líneas se usa la variable start, calculada como line\_count mod N en el caso de que la entrada supere las N líneas y 0 en caso contrario, y teniendo line\_count recogido el número total de líneas leídas por la entrada estándar (en el caso anterior, la primera línea a imprimir en la salida estándar sería la apuntada por el puntero situado en la posición 9 mod 7 = 2 en el caso de que line\_count fuera igual a 9, y es el valor que tendría recogido la variable entera start).

A la hora de imprimir las líneas por salida estándar, se imprimen por salida N líneas empezando por la cadena de caracteres apuntada por el puntero situado en la posición (start + i) mod N, para no acceder a espacios de memoria no permitidos.

Por último, se libera la memoria reservada para almacenar las frases anteriores, y se elimina el buffer que contiene los punteros a las líneas.

***Función Longlines***

En primer lugar, la función longlines reserva memoria para almacenar una lista de N líneas, simulando un TAD (Tipo Abstracto de Datos) de lista ordenada con NNN posiciones. Si la asignación de memoria falla, se devuelve un valor de error numérico 1. La función valida que NNN sea mayor que 0; en caso contrario, muestra un mensaje de error y finaliza la ejecución.

A continuación, se configura el entorno para manejar caracteres especiales en UTF-8 utilizando setlocale(LC\_ALL, ""). Luego, se inicializa un buffer temporal (temporalLine) para almacenar cada línea leída, así como una lista de LineEntry para guardar las N líneas de mayor longitud, que se inicializa con una longitud de 0 en cada entrada para facilitar las comparaciones.

La función lee cada línea desde la entrada estándar (stdin) y procesa cada una de ellas. Para cada línea leída, se calcula la longitud en bytes, y si esta termina con un salto de línea (\n), se elimina dicho carácter para evitar que afecte la longitud de la cadena. La línea en formato char\* (ASCII o UTF-8) se convierte a wchar\_t\* utilizando mbstowcs, lo cual permite calcular la longitud en caracteres en lugar de bytes, esencial para el manejo correcto de caracteres especiales.

Luego, se llama a la función auxiliar insertLine, que inserta la línea en la lista de N líneas más largas en orden de mayor a menor longitud.

Finalmente, la función imprime las N líneas más largas en la salida estándar en el orden requerido. Mientras imprime cada línea, libera el espacio de memoria utilizado por esta. Al finalizar la impresión de todas las líneas, se libera el espacio total ocupado por la lista, asegurando una gestión adecuada de la memoria.

***Estructura del programa Test***

En la ejecución de la función main() del programa *Test*, primero se realiza un control del número de argumentos introducidos al ejecutar alguna de las funciones. En el caso de haber más de 3 argumentos, el programa lanza un error numérico de valor 1, indicando al usuario que el número de argumentos no es correcto. En el caso de haber 3 argumentos, se toma como tercer argumento el número de líneas (siendo el primero la ruta relativa del ejecutable, y el segundo la extensión de la función a ejecutar), que se convierte a un número entero y se comprueba su formato, devolviendo un valor de error numérico 2 en el caso de que el formato no sea el correcto. En caso de haber solamente 2 argumentos, se asumirá como tercer argumento el valr numérico 10.

Acto seguido, se comprueba que las extensiones sean las correctas. En caso de serlo, se procederá a la ejecución de la función correspondiente. En caso contrario, devuelve un error de tipo numérico con valor 3, indicando al usuario que la extensión introducida no es correcta.

Si la ejecución del programa *Test* ha ido bien, se devolverá como resultado la salida de la función correspondiente a la extensión, con el número de líneas indicado.

## Principales Funciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **head** | **Nombre** | **Tipo** | **Descripción** |
| **Argumentos** | Argumento 1 | **N** | **int** | **Número de líneas a imprimir** |
| **Variables Locales** | Variable 1 | **count** | **int** | **Contador de línea impresas** |
|  | Variable 2 | **buffer** | **Array de tipo char** | **Almacenamiento de las líneas de caracteres** |
| **Valor Devuelto** |  |  | **int** | **0 en caso de ejecutarse sin ningún error. 1 en caso de ocurrir un error de lectura.** |
| **Descripción de la Función** |  |  |  | **Imprime en la salida estándar las N primeras líneas procedentes de la entrada estándar.** |

## 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **tail** | **Nombre** | **Tipo** | **Descripción** |
| **Argumentos** | Argumento 1 | **N** | **int** | **Número de líneas a imprimir** |
| **Variables Locales** | Variable 1 | **lines\_buffer** | **Puntero a punteros de tipo char** | **Almacenamiento de los punteros a las líneas seleccionadas.** |
|  | Variable 2 | **lines\_count** | **int** | **Contador de líneas leídas por entrada estándar** |
|  | Variable 3 | **start** | **int** | **Posición del buffer en la que se encuentra la primera línea a imprimir de las N líneas requeridas** |
|  | Variable 4 | **total\_lines** | **int** | **Número total de líneas a leer** |
| **Valor Devuelto** |  |  | **int** | **0 en caso de ejecutarse sin ningún error. 1 en caso de ocurrir un error de asignación de memoria en el buffer de punteros a línea. 2 en caso de ocurrir un error de asignación de memoria para almacenar una de las líneas** |
| **Descripción de la Función** |  |  |  | **Imprime en la salida estándar las N últimas líneas procedentes de la entrada estándar.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **longlines** | **Nombre** | **Tipo** | **Descripción** |
| **Argumentos** | Argumento 1 | **N** | **int** | **Número de líneas a imprimir** |
| **Variables Locales** | Variable 1 | **list** | **Array de N posiciones** | **Array que almacena las líneas leídas. Cada entrada del array contiene un campo line (para la línea) y un campo length (para la longitud de la línea).** |
|  | Variable 2 | **temporalLine** | **Array de tipo char** | **Buffer temporal para almacenar cada línea leída en formato char\*, utilizado para procesar la entrada estándar y convertirla a wchar\_t\*.** |
|  | Variable 3 | **wideLine** | **Array de tipo char** | **Buffer temporal para almacenar la conversión de cada línea leída en caracteres wchar\_t\* para facilitar el manejo de caracteres especiales.** |
|  | Variable 4 | **i** | **int** | **Índice utilizado en los bucles para iterar sobre la lista de líneas y el proceso de lectura.** |
| **Valor Devuelto** |  |  | **int** | **0 en caso de ejecutarse sin ningún error. 1 en caso de ocurrir un error de asignación de memoria en el buffer de líneas** |
| **Descripción de la Función** |  |  |  | **Encuentra las N líneas más largas de la entrada estándar, y las imprime en salida estándar en orden descencente** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **insertLine** | **Nombre** | **Tipo** | **Descripción** |
| **Argumentos** | Argumento 1 | **list[]** | **Array de tipo list** | **Lista de líneas donde se insertará la nueva línea.** |
|  | Argumento 2 | **N** | **int** | **Número de elementos en la lista de líneas** |
|  | Argumento 3 | **newLine** | **Puntero a char** | **Contenido de la línea a insertar en la lista.** |
|  | Argumento 4 | **newLenght** | **int** | **Longitud de la nueva línea a insertar.** |
| **Variables Locales** | Variable 1 | **i** | **int** | **Índice utilizado para encontrar la posición correcta donde insertar la nueva línea en el arreglo.** |
|  | Variable 2 | **j** | **int** | **Índice utilizado para mover las líneas existentes hacia abajo y hacer espacio para la nueva línea.** |
| **Valor Devuelto** |  |  | **void** | **La función no devuelve un valor (void), pero modifica directamente el array list[], insertando la nueva línea en la posición correcta.** |
| **Descripción de la Función** |  |  |  | **La función insertLine inserta una nueva línea en una lista de líneas list[]. La línea se inserta en el lugar adecuado según su longitud, desplazando las líneas existentes para hacer espacio. Primero, la función encuentra la posición en la que la nueva línea debe ser insertada, moviendo las líneas posteriores hacia abajo. Luego, se inserta la nueva línea en la posición determinada y se actualiza la longitud de la nueva línea en la lista.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **main** | **Nombre** | **Tipo** | **Descripción** |
| **Argumentos** | Argumento 1 | **argc** | **int** | **Contador de argumentos del programa** |
|  | Argumento 2 | **argv** | **Puntero a array de punteros de tipo char** | **Argumentos del programa** |
| **Variables Locales** | Variable 1 | **i** | **int** | **Asignador de número de líneas** |
| **Valor Devuelto** |  |  | **int** | **1 en el caso de haber más de tres argumentos. 2 en el caso de que el formato del segundo argumento no sea correcto. 3 en el caso de que la extensión introducida no sea correcta. Otros casos: se devuelve el valor correspondiente a la función ejecutada** |
| **Descripción de la Función** |  |  |  | **Función principal del programa de pruebas de la librería** |

## Casos de Prueba

|  |  |  |
| --- | --- | --- |
| head | | |
| Archivo de prueba | Línea de comando ejecutada (entrada) | Salida |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -head 2 < tests/prueba.txt | shdjflksdhflsdhfsdf  dfdf |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -head 1200 < tests/prueba.txt | shdjflksdhflsdhfsdf  dfdf  dfdfdfffffffffffffffffffffffff  fffdsfgrwertwert  wporuweruwepiur  dvcxvjdfhv  dgdfgjh  dsfdkf  mmmfmfm.kshfkshf  skbfdljdgfureth  ¿¿¿¿¿  11!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!l |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -head 0 < tests/prueba.txt |  |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -head < tests/prueba.txt | shdjflksdhflsdhfsdf  dfdf  dfdfdfffffffffffffffffffffffff  fffdsfgrwertwert  wporuweruwepiur  dvcxvjdfhv  dgdfgjh  dsfdkf  mmmfmfm.kshfkshf  skbfdljdgfureth |
|  | | |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -head 2 < tests/prueba2.txt | "Hola, ¿cómo estás?" me preguntó.  Ruta del archivo: C:\Archivos\MiArchivo.txt |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -head 90 < tests/prueba2.txt | "Hola, ¿cómo estás?" me preguntó.  Ruta del archivo: C:\Archivos\MiArchivo.txt  Primera línea.  Segunda línea.  Nombre Edad Ciudad  Juan 25 Madrid  Adiós!mundo!  Ella dijo: '¡Hola, mundo!'  Y él respondió: "¿Qué tal?"  Primer línea  Segunda línea  Tercera línea |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -head 0 < tests/prueba2.txt | - |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -head < tests/prueba2.txt | "Hola, ¿cómo estás?" me preguntó.  Ruta del archivo: C:\Archivos\MiArchivo.txt  Primera línea.  Segunda línea.  Nombre Edad Ciudad  Juan 25 Madrid  Adiós!mundo!  Ella dijo: '¡Hola, mundo!'  Y él respondió: "¿Qué tal?"  Primer línea |
|  | | |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -head 2 < tests/100000\_random\_sentences.txt | Para de cuida los oportunidad.  Proyectos sanar con dice poderosa la canción el. |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -head 0 < tests/100000\_random\_sentences.txt | - |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -head < tests/100000\_random\_sentences.txt | Para de cuida los oportunidad.  Proyectos sanar con dice poderosa la canción el.  De corre susurran ciudad oportunidades sol.  Cambiar motor una una de flores olas de.  Corazón comienza el son la las.  Pájaros una acompañan adelante el.  Cada se puertas oportunidad nueva damos.  Crecer flores a palabras amor una brillantes de herir noche adelante.  Un olas una libros progreso los y en.  Llena las café brillantes que estrellas historia noche risa en un que la poderosa el. |

|  |  |  |
| --- | --- | --- |
| tail | | |
| Archivo de prueba | Línea de comando ejecutada (entrada) | Salida |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -tail 2 < tests/prueba.txt | ¿¿¿¿¿  11!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -tail 1200 < tests/prueba.txt | shdjflksdhflsdhfsdf  dfdf  dfdfdfffffffffffffffffffffffff  fffdsfgrwertwert  wporuweruwepiur  dvcxvjdfhv  dgdfgjh  dsfdkf  mmmfmfm.kshfkshf  skbfdljdgfureth  ¿¿¿¿¿  11!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -tail 0 < tests/prueba.txt |  |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -tail < tests/prueba.txt | dfdfdfffffffffffffffffffffffff  fffdsfgrwertwert  wporuweruwepiur  dvcxvjdfhv  dgdfgjh  dsfdkf  mmmfmfm.kshfkshf  skbfdljdgfureth  ¿¿¿¿¿  11!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! |
|  | | |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -tail 2 < tests/prueba2.txt | Segunda línea  Tercera línea |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -tail 90 < tests/prueba2.txt | "Hola, ¿cómo estás?" me preguntó.  Ruta del archivo: C:\Archivos\MiArchivo.txt  Primera línea.  Segunda línea.  Nombre Edad Ciudad  Juan 25 Madrid  Adiós!mundo!  Ella dijo: '¡Hola, mundo!'  Y él respondió: "¿Qué tal?"  Primer línea  Segunda línea  Tercera línea |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -tail 0 < tests/prueba2.txt | - |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -tail < tests/prueba2.txt | Primera línea.  Segunda línea.  Nombre Edad Ciudad  Juan 25 Madrid  Adiós!mundo!  Ella dijo: '¡Hola, mundo!'  Y él respondió: "¿Qué tal?"  Primer línea  Segunda línea  Tercera línea |
|  | | |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -tail 2 < tests/100000\_random\_sentences.txt | Aprender mundo camino las cada esperanza playa damos camino los amistad cada la.  A proyectos nueva por juegan amor amistad. |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -tail 0 < tests/100000\_random\_sentences.txt | - |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -tail < tests/100000\_random\_sentences.txt | De de libros vida el despierta noche su.  Los las las brillantes damos camino el que brillantes mirada el en progreso por.  Estrellas historia en fuerza dependiendo son.  Rápidamente que una parque camino.  En niños el nos oscuridad brilla por un corre quién oscuridad son el mundos.  Motor paso adelante el pájaros poderosa una oportunidad.  El del con parque en el café son el del y.  Los a llena la en de que en que los son nueva damos.  Aprender mundo camino las cada esperanza playa damos camino los amistad cada la.  A proyectos nueva por juegan amor amistad. |

|  |  |  |
| --- | --- | --- |
| longlines | | |
| Archivo de prueba | Línea de comando ejecutada (entrada) | Salida |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -long 4 < tests/prueba.txt | 1 - 11!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! (Longitud en caracteres: 39)  2 - dfdfdfffffffffffffffffffffffff (Longitud en caracteres: 30)  3 - shdjflksdhflsdhfsdf (Longitud en caracteres: 19)  4 - fffdsfgrwertwert (Longitud en caracteres: 16) |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -long 1200 < tests/prueba.txt | 1 - 11!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! (Longitud en caracteres: 39)  2 - dfdfdfffffffffffffffffffffffff (Longitud en caracteres: 30)  3 - shdjflksdhflsdhfsdf (Longitud en caracteres: 19)  4 - fffdsfgrwertwert (Longitud en caracteres: 16)  5 - mmmfmfm.kshfkshf (Longitud en caracteres: 16)  6 - wporuweruwepiur (Longitud en caracteres: 15)  7 - skbfdljdgfureth (Longitud en caracteres: 15)  8 - dvcxvjdfhv (Longitud en caracteres: 10)  9 - dgdfgjh (Longitud en caracteres: 7)  10 - dsfdkf (Longitud en caracteres: 6)  11 - ¿¿¿¿¿ (Longitud en caracteres: 5)  12 - dfdf (Longitud en caracteres: 4) |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -long 0 < tests/prueba.txt | Error en el valor introducido (tiene que ser mayor que 0) |
| [prueba.txt](https://drive.google.com/file/d/1kxi9BTNUNDenXArrH8ELEUWfzUYgR3U_/view?usp=sharing) | ./test.exe -long < tests/prueba.txt | 1 - 11!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! (Longitud en caracteres: 39)  2 - dfdfdfffffffffffffffffffffffff (Longitud en caracteres: 30)  3 - shdjflksdhflsdhfsdf (Longitud en caracteres: 19)  4 - fffdsfgrwertwert (Longitud en caracteres: 16)  5 - mmmfmfm.kshfkshf (Longitud en caracteres: 16)  6 - wporuweruwepiur (Longitud en caracteres: 15)  7 - skbfdljdgfureth (Longitud en caracteres: 15)  8 - dvcxvjdfhv (Longitud en caracteres: 10)  9 - dgdfgjh (Longitud en caracteres: 7)  10 - dsfdkf (Longitud en caracteres: 6) |
|  | | |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -long 12 < tests/prueba2.txt | 1 - Ruta del archivo: C:\Archivos\MiArchivo.txt (Longitud en caracteres: 43) |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -long 90 < tests/prueba2.txt | 1 - Ruta del archivo: C:\Archivos\MiArchivo.txt (Longitud en caracteres: 43)  2 - "Hola, ¿cómo estás?" me preguntó. (Longitud en caracteres: 33)  3 - Y él respondió: "¿Qué tal?" (Longitud en caracteres: 27)  4 - Ella dijo: '¡Hola, mundo!' (Longitud en caracteres: 26)  5 - Nombre Edad Ciudad (Longitud en caracteres: 22)  6 - Juan 25 Madrid (Longitud en caracteres: 22)  7 - Tercera línea (Longitud en caracteres: 21)  8 - Segunda línea (Longitud en caracteres: 17)  9 - Primera línea. (Longitud en caracteres: 14)  10 - Segunda línea. (Longitud en caracteres: 14)  11 - Adiós!mundo! (Longitud en caracteres: 12)  12 - Primer línea (Longitud en caracteres: 12) |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -long 0 < tests/prueba2.txt | Error en el valor introducido (tiene que ser mayor que 0) |
| [prueba2.txt](https://drive.google.com/file/d/1CWnDEH8bO11_JLvfCnKJGSwiAiPL5tAj/view?usp=sharing) | ./test.exe -long < tests/prueba2.txt | 1 - Ruta del archivo: C:\Archivos\MiArchivo.txt (Longitud en caracteres: 43)  2 - "Hola, ¿cómo estás?" me preguntó. (Longitud en caracteres: 33)  3 - Y él respondió: "¿Qué tal?" (Longitud en caracteres: 27)  4 - Ella dijo: '¡Hola, mundo!' (Longitud en caracteres: 26)  5 - Nombre Edad Ciudad (Longitud en caracteres: 22)  6 - Juan 25 Madrid (Longitud en caracteres: 22)  7 - Tercera línea (Longitud en caracteres: 21)  8 - Segunda línea (Longitud en caracteres: 17)  9 - Primera línea. (Longitud en caracteres: 14)  10 - Segunda línea. (Longitud en caracteres: 14) |
|  | | |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -long 2 < tests/100000\_random\_sentences.txt | 1 - Historia desconocidos contar libros hablan gato brillantes corazón susurran universo palabras parque oportunidad cantan paso. (Longitud en caracteres: 125)  2 - Oportunidades adultos camino acompañan oportunidades pájaros para oportunidades cada ciudad de mirada la hojas recuerdos. (Longitud en caracteres: 121) |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -long 0 < tests/100000\_random\_sentences.txt | Error en el valor introducido (tiene que ser mayor que 0) |
| [100000\_random\_sentences.txt](https://drive.google.com/file/d/1y9wkYHwzy2HNipYELUX3JhG20aurEoN5/view?usp=drive_link) | ./test.exe -long < tests/100000\_random\_sentences.txt | 1 - Historia desconocidos contar libros hablan gato brillantes corazón susurran universo palabras parque oportunidad cantan paso. (Longitud en caracteres: 125)  2 - Oportunidades adultos camino acompañan oportunidades pájaros para oportunidades cada ciudad de mirada la hojas recuerdos. (Longitud en caracteres: 121)  3 - Brilla día susurran sueños dependiendo aprender mientras son sueños despierta corazón más proyectos desconocidos cantan. (Longitud en caracteres: 120)  4 - Poderosa café que llena futuro una canción sueños rápidamente dependiendo despierta disfrutan proyectos rayito progreso. (Longitud en caracteres: 120)  5 - Flores dependiendo hablan ciudad con mientras estrellas rápidamente comienza vida caliente dependiendo amistad las por. (Longitud en caracteres: 119)  6 - Rompen con a rápidamente esperanza aprender brillantes cada mundo disfrutan mundo nueva oportunidades orilla disfrutan. (Longitud en caracteres: 119)  7 - Olas adultos susurran mirada mientras con brillantes aventura brillantes oportunidad corre ciudad del de desconocidos. (Longitud en caracteres: 118)  8 - Nueva playa desconocidos llena comienza cuida mientras oportunidades tesoro ideas de es rápidamente mientras comienza. (Longitud en caracteres: 118)  9 - Estrellas juegan oportunidad hermosas niños comienza cuida oportunidad el corazón cada en oportunidad brillantes risa. (Longitud en caracteres: 118)  10 - Oportunidades orilla oportunidades los gato tesoro dependiendo de corre quién comienza el el brillantes desconocidos. (Longitud en caracteres: 117) |

# Comentarios Personales

## PROBLEMAS ENCONTRADOS

El problema principal que hemos encontrado ha sido simular el funcionamiento de una lista ordenada en un lenguaje de programación que no cuenta con orientación a objetos. Prácticamente cada vez que se hacía uso de memoria, daba algún tipo de error, como el indicado arriba referente a acceso a posiciones de memoria no permitidas. Finalmente, se ha podido implementar una lista ordenada especializada en cadenas de caracteres que inserta (si es posible) la cadena en la primera posición disponible teniendo en cuenta la longitud de las cadenas de caracteres ya presentes en la misma.

Otro problema relacionado con el proceso de elaboración del TAD ha sido la falta retroalimentación de errores de memoria por parte del compilador, haciendo que a veces las funciones no funcionaran correctamente, pero tampoco lanzasen ningún error. Esto hizo bastante costoso el proceso de refinamiento y comprobación de las funciones.

Una vez programada los algoritmos para las tres funciones, sin ningún tipo de error común (como los acceso a memoria), detectamos que, en el caso de longlines, no detectaba bien el tamaño de las líneas con caracteres especiales (como las tildes, las interrogaciones, etc.). Por ello, tuvimos que investigar en distintas páginas web y documentación, lo que supuso más tiempo del previsto.

## CRÍTICAS CONSTRUCTIVAS

No tenemos ninguna crítica. Se nos ha dado bastante tiempo, y el temario necesario para poder hacer la mayoría de la práctica sin problemas. La única función un poco más complicada era la última, pero era de una dificultad asumible respecto a la asignatura.

## PROPUESTA DE MEJORAS

Por nuestra parte, no hemos encontrado ningún aspecto de la práctica que pueda o deba ser mejorado.

## EVALUACIÓN DEL TIEMPO DEDICADO

En total, teniendo en cuenta el tiempo empleado en la lectura y comprensión del tema y en la realización de los ejercicios correspondientes al tema 3 y las funciones pedidas en esta práctica llegamos a la conclusión de que, por cada uno de nosotros, se ha requerido un tiempo medio de unas 30 horas. Indudablemente, la función de la librería pedida en la que hemos empleado más tiempo ha sido la función longlines, por el hecho de tener que implementar una estructura de datos auxiliar no disponible en C para el proceso de ordenación de cada una de las líneas introducidas.