

Apellido:		Fecha:	
Nombre:		Docente:	
División:		Nota:	
Legajo:		Firma:	
Instancia:	PP		RPP
			SP
			RSP
			X
			FIN
			X

PARTE 1

Desarrollar en ANSI C:

Los logs son archivos en los cuales se registra información inherente a los procesos ejecutados en un servidor. Se requiere desarrollar un programa el cual permita separar y filtrar los mensajes de fallos reportados por un servidor web. Todos los mensajes generados por el servidor, se escribirán en un archivo **log.txt**. Cada línea del archivo tendrá información separada por “;” y el siguiente formato:

Fecha	Hora	Nombre del servicio que falló (64 caracteres máximo)	Gravedad del error (0 a 9)	Mensaje de error (64 caracteres máximo)
-------	------	--	----------------------------	---

Ejemplo:

09/07/2020;09:33;ServicioClientes.exe;5;Disk faliure
 09/07/2020;09:38;ScriptCache.exe;9;Out of memory
 10/07/2020;10:39;ServiceCleanDisk.exe;6;Segmentation fault

El programa tendrá un menú con las opciones:

1. Leer Log
2. Procesar información
3. Mostrar estadísticas
4. Imprimir lista logs

1. Leer Log

Deberá parsear el archivo log.txt cargando una lista dinámica con las entidades “LogEntry” descripta a continuación.

```
struct S_LogEntry
{
    char date[11];
    char time[6];
    char serviceName[65];
    int gravedad;
    char msg[65];
};
typedef struct S_LogEntry LogEntry;
```

2. Procesar información

Deberá evaluar el campo “Gravedad” con el siguiente criterio:

- Si la gravedad es menor a 3, se descartará el error.
- Si la gravedad es 3, se deberán copiar los mensajes en el archivo warnings.txt.
- Si la gravedad tiene un valor entre 4 y 7 (inclusive) se imprimirán por pantalla con el siguiente formato :

Fecha	Hora	Nombre servicio	Mensaje de error	Gravedad
-------	------	-----------------	------------------	----------

- Si la gravedad es mayor a 7, se copiarán los mensajes en el archivo errors.txt.

Cada línea de los archivos **warnings.txt** y **errors.txt** deberán tener el siguiente formato:

Fecha	Hora	Nombre servicio	Mensaje de error	Gravedad
-------	------	-----------------	------------------	----------

Los campos estarán separados por “;”

3. Mostrar estadísticas

Deberá indicar por pantalla la cantidad de fallos para los siguientes valores de gravedad:

- Menores a 3
- 3
- 4 y 7 (inclusive)
- Mayores a 7

Nota 0: El código deberá tener comentarios con la documentación de cada una de las funciones y respetar las reglas de estilo de la cátedra.

Nota 1: Se deberá utilizar la biblioteca LinkedList y la función ll_filter() para generar las listas filtradas que luego se imprimirán en pantalla o en los archivos.

Nota 2: Cada una de las entidades identificadas deberá contar con su archivo *.h y *.c

CONDICIONES DE APROBACIÓN

Para la aprobación directa (nota ≥ 6), se deberá tener el programa funcionando en su totalidad como se pide en la parte 1 del examen y haber contestado todas las preguntas de la parte 2, que se darán el día del examen.

Para la aprobación con final (nota = 4 o 5), se deberá realizar el parseo del archivo, la función ll_filter, y haber contestado la pregunta oral de la estructura de la LinkedList.