

<p align="center">Universidad Tecnológica Nacional Facultad Regional Avellaneda</p>					
<p align="center">Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos</p>					
<p>Materia: Laboratorio de Computación I</p>					
Apellido:			Fecha:	30/11/2022	
Nombre:			Docente ⁽²⁾ :	Baus / Maggiori	
División:	1°A		Nota ⁽²⁾ :		
Legajo:			Firma ⁽²⁾ :		
Instancia ⁽¹⁾ :	PP		RPP		SP
				X	RSP
					FIN

Se dispone de un archivo con datos acerca de películas, que tiene el siguiente formato:

id_peli, titulo, genero, duracion

por ejemplo: 1, Adventures of Rocky, Adventure, 0

2, My Brother the Devil, Drama, 0

3, Criminal, Drama, 0

se deberá realizar un programa que permita el análisis de dicho archivo y sea capaz de generar nuevos archivos de salida de formato similar filtrados por varios criterios:

el programa contará con el siguiente menú:

- 1) Cargar archivo:** Se pedirá el nombre del archivo y se cargará en un linkedlist los elementos del mismo.
- 2) Imprimir lista:** Se imprimirá por pantalla la tabla con los datos de las películas.
- 3) Asignar tiempos:** Se deberá hacer uso de la función map. la cual recibirá el linkedlist y una función que asignará a la película un valor de duración entre 100 y 240 minutos calculado de manera aleatoria se mostrará por pantalla el mismo.
- 4) Filtrar por tipo:** Se deberá generar un archivo igual al original, pero donde solo aparezcan películas del tipo seleccionado.
- 5) Mostrar duraciones:** Se deberá mostrar por pantalla un listado de las películas ordenadas por género y dentro de las del mismo género que aparezcan ordenadas por duración descendente.
- 6) Guardar películas:** Se deberá guardar el listado del punto anterior en un archivo de texto.
- 7) Salir.**

Requerimientos del desarrollo. • Se deberá crear la entidad “eMovie” con todos sus campos correspondientes. • se deberá utilizar la biblioteca linkedlist para almacenar las películas leídas del archivo. • se deberá agregar a la biblioteca la función “ll_filter ()” la cual devolverá una nueva linkedlist que contenga alguno de los elementos de la lista original, según algún criterio • se deberá utilizar la función

Detalle de la función “ll_filter ()” prototipo de la función:

linkedlist* ll_filter (linkedlist* this, int (*pFunc) (void* element));

la función “ll_filter” recibirá una lista y una función “pFunc”. se deberá iterar todos los elementos de la lista y pasárselos a la función “pFunc”. la función “pFunc” devolverá 1 si ese ítem se debe agregar a la lista resultado o 0 si no debe agregarse. la función “ll_filter” generará la nueva lista resultado, agregará a la misma los ítems correspondientes y la devolverá.

Detalle de la función “ll_map ()” prototipo de la función:

linkedlist* ll_map (linkedlist* this, void* (*pFunc) (void* element));

La función “ll_map” recibirá una lista y una función “pFunc”. se deberán iterar todos los elementos de la lista y pasárselos a la función “pFunc” que recibirá el elemento y le calculará el campo tiempo. el retorno de “pFunc” se agregará a la lista resultado. esta nueva lista será devuelta por ll_map.

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: separar en archivos las entidades, parser y generador de informes.

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

Para la aprobación con final (nota = 4 o 5), se deberá realizar el parseo del archivo, la función ll_map. y guardado en archivo de texto.o.