Universidad Tecnológica Nacional Facultad Regional Avellaneda										
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos										
Materia: Laboratorio de Computación I										
Apellido:					Fecha:		29/06/2020			
Nombre:					Docente (2):					
División:					Nota (2):					
Legajo:					Firma (2):					
Instancia (1):	**PP**		**RPP**		**SP**	**X**	**RSP**		**FIN**	

Se dispone de un archivo con datos acerca de los participantes de una carrera de bicicletas, que tiene el siguiente formato:
id_bike, **nombre** (del dueño), **tipo (**bmx, playera, mtb, paseo), **tiempo**

por ejemplo: 50,jorge,bmx,0
51,sofia,paseo,0
52,andrea,mtb,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 bicicletas.
3) Asignar tiempos: Se deberá hacer uso de la función map. la cual recibirá el linkedlist y una función que asignará a la bicicleta un valor de tiempo entre 50 y 120 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 bicicletas del tipo seleccionado.
5) Mostrar posiciones: Se deberá mostrar por pantalla un listado de las bicicletas ordenadas por tipo y dentro de las del mismo tipo que aparezcan ordenadas por tiempo ascendente.
6) Guardar posiciones: Se deberá guardar el listado del punto anterior en un archivo de texto.
7) Salir.

Requerimientos del desarrollo. • Se deberá crear la entidad “eBicicleta” con todos sus campos correspondientes. • se deberá utilizar la biblioteca linkedlist para almacenar las bicicletas 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.