

Se dispone de un archivo con datos sanitarios de países afectados por el virus Covid-19, que tiene el siguiente formato:

por ejemplo

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:

1) **Cargar archivo:** Se pedirá el nombre del archivo y se cargará en un linkedlist los elementos del mismo.

3) Asignar estadísticas: Se deberá hacer uso de la función map, la cual recibirá el linkedlist y una función que asignará a cada país un valor de vacunados con una dosis (entre 1% y 60%), vacunados con dos dosis (entre 1% y 40%), ambos asignados de manera aleatoria y porcentaje sin vacunar que será lo que falte para el 100%.

5) **Filtrar por países en el horno:** Se deberá generar un archivo igual al original, pero donde solo aparezcan países donde la cantidad de no vacunados sea mayor a la de vacunados.

6) Ordenar por nivel de vacunación: Se deberá mostrar por pantalla un listado de los países ordenados por cantidad de vacunados con 1 dosis.

7) **Mostrar más castigado:** Informar el nombre del país o países con el mayor porcentaje de no vacunados. Y cuál es ese número.

8) Salir.

Requerimientos del desarrollo. • Se deberá crear la entidad “ePaís” con todos sus campos correspondientes. • se deberá utilizar la biblioteca linkedlist para almacenar los países del archivo. • La función “ll_filter ()” 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.

Se deberá agregar al LinkedList 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”. ll_map itera todos los elementos de la lista y se los pasa a la función “pFunc” que recibirá el elemento y le asignará a los campos vac1Dosis y vac2Dosis un número aleatorio y calculará el campo sinVacunar. el retorno de “pFunc” que es el país con todos sus campos cargados 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 para la aprobación directa (nota ≥ 6), se deberá tener el programa funcionando en su totalidad.

Para la aprobación con final (nota = 4 ó 5), se deberá realizar el parseo del archivo, la función ll_map, y haber resuelto algún filtrado.