Algoritmos y Estructuras de Datos
Guía de ejercicios de preparación para la 2da eval. parcial - parte 2 (versión 2.6)
Temas: Archivos. Estructura cola, pila, lista, lista doblemente enlazada. Combinación de estructuras.

3) Sensores IoT[1]. Se debe crear un módulo que recupere la secuencia de mediciones obtenidas desde un conjunto de sensores de temperatura ubicados en estaciones remotas y que fueron guardados en un archivo de texto.

Se pide:
Crear la función ***leer()*** que lea el archivo de texto donde se guardaron las mediciones (a medida que pudieron ser transmitidas) y agregarlas a una estructura de datos en memoria. Para una misma hora puede haber ninguna o muchas mediciones.
Cada nodo de la lista deberá tener el dato de la hora (entero) y un puntero a una sublista cuyos nodos guardarán: identificador del sensor (entero) y medición (decimal) ordenados por <id-sensor>.

Los datos a leer del archivo tienen el formato:
`<id-sensor>, <hora:minutos-lectura>, <medición>` (una medición por línea).

Mostrar por pantalla con el siguiente formato:

```
<hora0>: <id-sensor-1, medición1>, <id-sensor-2, medición2>, ..., <id-sensor-N, mediciónN>
<hora1>: <id-sensor-1, medición1>, <id-sensor-2, medición2>, ..., <id-sensor-N, mediciónN>
...
<hora23>: ...
```

Nota: El archivo de mediciones es suministrado. Para dividir por el caracter ‘,’ las palabras de la línea con las mediciones, se recomienda usar la función ***strtok()*** (ver https://www.campusvirtual.frba.utn.edu.ar/especialidad/mod/resource/view.php?id=82907).

4) Antivirus. El objetivo es crear el módulo actualizador de firmas desde un archivo. Para ello se debe leer el archivo binario firmas.dat (proporcionado) que contiene los siguientes datos:

a. id (entero)
b. nombre (cadena de 25 caracteres)
c. firma (cadena de 25 caracteres)

El algoritmo que detecta virus analiza los archivos del sistema de archivos (file system del sistema operativo) y los compara con la firma de todos los nodos una lista almacenada en memoria.

Se pide:
Crear la estructura para leer el archivo de registros.
Crear una lista enlazada con los datos leídos del archivo. La función ***importar()*** deberá agregar a una lista en memoria nodos ordenados alfabéticamente por el campo Nombre.

Crear la función ***analizar()*** que recibe como parámetro un array de cadenas de caracteres (que representa los archivos a analizar) y la lista de firmas. Retorna un booleano indicando fue detectado un virus.

Crear la función ***exportar()*** que guardará el archivo firmas.dat con el contenido de la lista actualizada. Recibirá como parámetros un puntero a la lista de firmas y el nombre del archivo a guardar.

Lotes para probar la función ***analizar()***:

[1] http://www.interempresas.net/Medicion/FeriaVirtual/Producto-Sensores-inteligentes-para-IoT-176331.html

Lote 1 (hay virus)
{“fasdfafafaeeww”, “2jhg3jf2hjf4h23hgj234j”, “32kljh34kl2h3lk4g23”, “213kjh4k23g4g23io”, “2o34y2332nsnjhsaghkj”}

(el elemento en la posición 1 coincide con la firma de uno de los virus)

Lote 2 (sin virus)
{“afasklfalksfuiyiy”, ”opteqljhweqljkhrefadslfd”, ”up32ljh324kjhl234kj”, “ln32kljh432klj4klhljk”, “32uoi4kjhl234khg234jk”}

Lote 3 (hay virus)
{“fasjkhfkljhfjkhah”, ”khjl32jkhl234jkgh43jhg”, ”kh324khj342khj243k”, ”kjh32kjhl43kjhl24jkh”, ”kljhfakjsjkhlsdahjksdl”}

(el último archivo coincide con un virus)

Para comparar string en c++

```
string cat = "felino";
string human = "humano";

cout << cat.compare(human) << endl;
```

Más info: http://www.cplusplus.com/reference/string/string/compare/

Para convertir un array de char en string:

```
char array [10];
string s = string(array);
```