

IMPORTANTE:

- ☐ Crear un proyecto con su Nombre y Apellido.
- ☐ Realizar todas las funciones que se indican.
- ☐ Añadir comentarios a su código identificando con el número de ejercicio e inciso a la/s función/es vinculadas a la resolución. Si no se identifica, la resolución no tendrá validez.

Una empresa de venta de indumentaria deportiva nos ha encomendado realizar un pequeño sistema para manejar el stock de productos en sus distintas sucursales.

La información se encuentra almacenada en el archivo subido al campus, el cual hay que descargar y copiar dentro de la carpeta del proyecto del Parcial. Se trata de un archivo de estructuras "stRegistro" que responde a la siguiente estructura de datos:

```
typedef struct{
    int idSucursal;
    char nombreSucursal[25];
    char nombreProducto[25];
    char deporte[25];
    int stockProducto;
}stRegistro;
```

Se nos pide desarrollar la siguiente funcionalidad:

1. **Hacer una función que recorra el archivo adjuntado y cargue los datos en una LISTA (simple o doble) de ARBOLES (LDA)**, clasificando los datos de la siguiente manera:

<p>En cada nodo de la lista se ubica c/u de las sucursales de la empresa, de acuerdo a la siguiente struct:</p> <pre>typedef struct{ int idSucursal; char nombreSucursal[25]; }stSucursal;</pre>	<p>En cada nodo de la lista también se encuentra un árbol con todos los productos que tiene en stock cada sucursal, de acuerdo a la siguiente struct:</p> <pre>typedef struct{ char nombreProducto[25]; char deporte[25]; int stockProducto; }stProducto;</pre> <p>En dicho árbol, los productos deberán ordenarse <u>por orden alfabético de acuerdo al campo "nombreProducto"</u></p>
--	--

Para realizar esto, primero deberán codificarse las estructuras necesarias para manejar la lista y para manejar los árboles, y luego deberán codificarse (**modularizando**) las funciones que sean necesarias para:

- Copiar los datos correspondientes a la estructura stSucursal y a la estructura stProducto.
- Buscar la sucursal a la cual corresponde el producto a cargar en la LDA
- Agregar una nueva sucursal si ésta no existe.
- Agregar un nuevo producto a la sucursal que corresponda (recordar que **en el árbol los productos deberán insertarse por orden alfabético de acuerdo al campo "nombreProducto"**) (**ojo: puede haber productos repetidos**; ej: calza puede estar repetido porque hay calzas de running, calzas de ciclismo, etc).

(35 puntos)

2. Realizar las funciones necesarias para **mostrar todos los elementos de la estructura compuesta**, permitiendo visualizar cada sucursal con sus productos en stock. **Los PRODUCTOS de cada sucursal (el árbol) deberán mostrarse EN ORDEN ALFABETICO (pensar qué modo de mostrado usar). (10 puntos)**
3. Nuestro cliente desea poder buscar determinado producto y ver en cuáles de sus sucursales hay stock del mismo. Para eso, deberá **hacerse una función que reciba por parámetro el nombre del producto y el deporte al cual corresponde** (puede usarse como ejemplo “remera manga larga” “running”, que es un producto existente en el archivo), **y que busque si existe o no en el árbol de productos de c/u de las sucursales**
 - Ojo! recordar que el árbol está ordenado de acuerdo al campo nombreProducto, y que ahora hay que buscar por 2 campos: “nombreProducto” y “deporte”.
 - Pensar bien cómo buscar para no perder la eficiencia que tienen las búsquedas binarias en los Arboles Binarios de Búsqueda.
 - **Para poder mostrar las sucursales correspondientes, deberán pasarse a un arreglo las sucursales en las cuales hay stock del producto buscado y mostrar dicho arreglo en el main.**
 - **Modularizar. (17 puntos)**
4. Nuestro cliente también desea poder averiguar en cuál de sus sucursales hay menor stock de productos de un deporte determinado. Para eso, deberá **hacerse una función que reciba por parámetro el deporte, sume el stock de todos los productos de ese deporte en c/u de las sucursales, y devuelva la sucursal que tiene menor cantidad de los mismos.** (Ojo! recordar que el árbol está ordenado de acuerdo al campo nombreProducto). **Modularizar. (14 puntos).**
5. **Calcular qué porcentaje representa el stock total de productos de una de las sucursales con respecto al stock total de todas las sucursales.** Para ello, deberá modularizarse:
 - sumar stock de todos los productos de la sucursal elegida (a elección del usuario del sistema, deberá ingresar por parámetro)
 - sumar stock de todos los productos de todas las sucursales de la empresa
 - calcular porcentaje del stock de la sucursal con respecto al stock total **(14 puntos)**
6. Hacer una función main() **(10 puntos)**
 - Para hacer esto, cree las variables que considere necesarias e invoque las funciones (de forma directa o indirecta) como corresponde en cada caso.
 - Muestre los resultados cada vez que sea necesario.
 - A fin de identificar cada inciso, comente su código indicando a qué apartado corresponde, por ejemplo: // Ejercicio 3.a