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 gran tienda nos ha encomendado realizar un pequeño sistema para manejar los vendedores de los distintos sectores de su tienda.
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 “stRegistroVendedor” que responde a la siguiente estructura de datos:

```
typedef
    struct{ int
    idSector;
    char sector[30];///los sectores son (y están escritos tal cual): "ropa","muebles","electro"
    int dni;
    char
    nombreYapellido[40]; int
    ventasDiarias;
}stRegistroVendedor;
```

Se nos pide desarrollar la siguiente funcionalidad:

1. Hacer una función que recorra el archivo binario adjuntado y copie los datos en un ARREGLO de ARBOLES (ADA). La información almacenada deberá distribuirse según los siguientes tipos de datos:

typedef struct{ int idSector; char sector[30]; }stSector;	typedef struct{ int dni; char nombreYapellido [40]; int ventasDiarias; }stVendedor;

   Para realizar esto, primero deberá definir la estructura del árbol, del arreglo y codificarlas. Luego deberán codificarse (**modularizando**) las funciones que sean necesarias para gestionar el arreglo y el árbol. **(22 puntos)**

2. Realizar las funciones necesarias para mostrar todos los elementos de la estructura compuesta, **permitiendo visualizar cada sector con sus vendedores**.
   - **Los vendedores de cada sector (el árbol) deberán mostrarse EN ORDEN CRECIENTE DE ACUERDO AL DNI (pensar qué modo de mostrado usar). (8 puntos)**

3. Nuestro cliente desea poder buscar un empleado y ver sus ventas diarias.

- Para eso, deberá hacer una función que reciba por **parámetro el dni del empleado**, lo busque en c/u de los sectores, y una vez encontrado, **retorne sus ventas diarias** (Pensar qué retornar si el empleado no existe en el ADA). **Modularizar. (15 puntos).**

4. Nuestro cliente también desea poder buscar en un sector de su elección cuántos de sus vendedores superan cierta cantidad de ventas diarias a elección del usuario.

- Para eso, deberá hacerse una función que reciba por parámetro la cantidad de ventas diarias a superar y el id del sector que le interesa analizar, busque el sector elegido (con la función ya hecha para el ejercicio 1), cuente cuántos de los vendedores de ese sector superan ese monto de ventas, y lo retorne. **Modularizar. (15 puntos).**

5. Nuestro cliente también desea poder calcular cuál fue el sector que tuvo más ventas diarias. Para ello, deberá modularizarse:
   - Hacer una función que sume y retorne las ventas diarias totales de UN sector.
   - Hacer una función que analice TODOS los sectores, y encuentre y retorne el sector que tuvo mayor cantidad ventas diarias **(15 puntos)**

6. Calcular qué porcentaje representan las ventas diarias totales de un sector a elección del usuario con respecto a las ventas diarias totales de todos los sectores.
   - Para ello, deberá modularizarse:
     - Buscar el sector elegido, el cual deberá ingresar por parámetro, utilizando la función hecha en el ejercicio 1.
     - Utilizar la función hecha en el ejercicio 5 para sumar las ventas diarias del sector elegido.
     - Utilizar la función hecha en el ejercicio 5 para sumar las ventas diarias de todos los sectores de la tienda
     - calcular el porcentaje de ventas del sector elegido en relación al total de ventas diarias de toda la tienda **(15 puntos)**

7. 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