

**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 cadena de tiendas nos ha encomendado realizar un pequeño sistema para analizar los rendimientos de sus vendedores.

Los datos a analizar se encuentran almacenados en el archivo de vendedores subido al campus, el cual hay que descargar y copiar dentro de la carpeta del proyecto del Parcial, y responde a la siguiente estructura de datos:

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

En el sistema a desarrollar se utilizarán las siguientes estructuras:

```
typedef struct _nodoS {
    stVendedor dato;
    struct _nodoS * siguiente;
} nodoS;

typedef struct _nodoD {
    stVendedor dato;
    struct _nodoD * siguiente;
    struct _nodoD * anterior;
} nodoD;

typedef struct {
    nodoD * inicio;
    nodoD * fin;
}Fila;
```

Se nos pide desarrollar la siguiente funcionalidad:

1. Utilizando el archivo de Vendedores, hacer una función que recorra el mismo y construya una **LISTA SIMPLE de Vendedores**. Para realizar esto, deberá **modularizar** y codificar las siguientes funciones:
  - a) Una función auxiliar que cree un *nodoS*, cargue sus datos y retorne un puntero al mismo. **(2 puntos)**
  - b) Una función auxiliar que agregue un nuevo nodo **al final** de una lista **de forma recursiva**. **(5 puntos)**
  - c) Un subprograma o función principal, que, por cada registro leído del archivo, cree un nodo y lo agregue a la lista. La función recibe la lista vacía y retorna la lista cargada. **(10 puntos)**
2. Realizar una función que muestre la **LISTA SIMPLE** de Vendedores cargada previamente. La misma deberá estar **modularizada** de la siguiente manera: **(8 puntos)**
  - a) Una función principal que recorre y muestra la lista.
  - b) Una función auxiliar que muestra un vendedor.
3. Realizar una función **recursiva** que retorne la cantidad de Vendedores pertenecientes al rubro "ropa" que tiene la **LISTA SIMPLE** de Vendedores. **(12 puntos)**
4. La empresa que nos encargó el proyecto desea premiar a los vendedores que realicen mayor cantidad de ventas diarias. Para ello debemos realizar un top ten ordenado del mayor al menor **modularizando** de la siguiente manera:
  - a. Hacer una función que busque en la **LISTA SIMPLE** al vendedor con mayor cantidad de ventas diarias y retorne el id de éste. **(13 puntos)**

- b. Hacer una función de **borrado de nodo** que busque en la **LISTA SIMPLE** el nodo que coincida con el Nro de id pasado por parámetro, guarde y **retorne** la estructura de tipo **stVendedor** correspondiente a ese nodo, y borre el nodo de la lista. Para ello, deberá recibirse por parámetro un **puntero doble a la lista**. **(17 puntos)**
  - c. Finalmente, deberá realizarse un subprograma o función principal que guarde el top ten de vendedores en una **FILA IMPLEMENTADA CON LISTA DOBLE**. Para ello, por cada vendedor encontrado con la función del inciso a), deberá crearse un nodo de lista doble y luego deberá agregarse en la Fila, de acuerdo a las estructuras del comienzo, y hasta llegar a la cantidad de 10 vendedores. **(12 puntos)**
5. Hacer una función que muestre la **FILA** obtenida en el punto anterior, **modularizando** de la siguiente manera: **(10 puntos)**
  - a) La función mostrar de **Fila** invocará a la función mostrar de Lista Doble y le pasará los parámetros que correspondan.
  - b) La función mostrar de **Lista Doble** deberá ser **recursiva y mostrar la lista al revés**.
  - c) La función mostrar de Lista Doble deberá invocar a la función mostrar un vendedor del inciso 2) b).
6. Hacer una función main() **(11 puntos)**
  - Para hacer esto, cree las variables que considere necesarias e invoque las funciones (de forma directa o indirecta) como corresponde en cada caso.
  - **RECORDAR INICIALIZAR LAS LISTAS Y FILAS (para ello deberán realizarse las respectivas funciones)**
  - 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