

**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.

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 **FILA de Vendedores trabajada con LISTA DOBLE**. Para realizar esto, deberá **modularizar** y codificar las siguientes funciones:
  - a) Una función auxiliar que cree el nodo de la Lista Doble, la cual debe codificarse con **PUNTERO DOBLE. (2 puntos)**
  - b) Una función auxiliar que agregue el vendedor a la Fila. Pensar bien en qué lugar de la Fila deberá ser agregado. **(5 puntos)**
  - c) Un subprograma o función principal, que, por cada registro leído del archivo, lo agregue a la Fila. Recordar inicializar lo que corresponda. **(8 puntos)**
2. Hacer una función que muestre la **FILA** obtenida en el punto anterior, **modularizando** de la siguiente manera: **(8 puntos)**
  - a) La función mostrar de **Fila** invocará a la función mostrar de Lista Doble.
  - b) La función mostrar de **Lista Doble** deberá **mostrar la lista AL REVÉS de manera ITERATIVA**. Para ello, pensar qué parámetros deberá recibir.
  - c) La función mostrar de Lista Doble deberá invocar a la función **mostrar un Vendedor**.
3. La empresa quiere poder calcular cuánto deberá abonar por día a sus empleados en cada sector.  
Para eso, hay que realizar una **única función** que recorra **una sola vez** la **FILA**, y sumar el total a abonar en cada sector, teniendo en cuenta que a cada vendedor se le paga por día el 20% de sus ventas diarias. **Deberán retornarse los totales a abonar en c/u de los 3 sectores; pensar cómo hacerlo. (20 puntos).**

4. La empresa también quiere poder calcular un PLUS por ventas diarias que abona sólo a los vendedores del sector “electro”. Para eso debemos hacer lo siguiente:
- a) Pasar a una **LISTA SIMPLE** todos los vendedores del sector “**electro**”, **sin borrar los datos de la Fila**, y **ordenados de menor a mayor** de acuerdo a su cantidad de ventas diarias. **(17 puntos)**.
  - b) Realizar una función **RECURSIVA** que retorne la cantidad de Vendedores que tiene dicha **LISTA SIMPLE** que superen una cantidad de ventas diarias ingresada por parámetro **(15 puntos)**.
  - c) Realizar una función **RECURSIVA** que calcule el total a pagar por día en concepto de PLUS a **todos** los vendedores de electro de la **LISTA SIMPLE**, para lo cual deberá sumar \$500 por cada vendedor que NO supere el monto de ventas diarias establecido por la empresa, y \$2000 por cada vendedor que sí lo supere. **(17 puntos)**.
5. Hacer una función main() **(8 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