

IMPORTANTE:

- **Crear un proyecto con su Nombre y Apellido.**
- **Realizar todas las funciones que se indican.**
- **Añadir comentarios a su código identificando cada inciso.**

Como parte del proceso previo al TP Final de la cátedra Laboratorio 2, se nos ha encomendado realizar un pequeño módulo para organizar información existente en un archivo de Compras de Clientes. El archivo consta de la siguiente estructura de datos:

```
typedef struct{
    int NroCliente;
    char NyA[30];
    int NroPedido;
    float MontoCompra;
} RegistroCompraCliente;
```

Contaremos con una lista simplemente vinculada con la siguiente estructura:

```
typedef struct{
    int NroCliente;
    char NyA[30];
} Cliente;

typedef struct{
    int NroPedido;
    Cliente cliente;
    float MontoCompra;
    struct NodoPedido * siguiente;
} NodoPedido;
```

Se nos pide desarrollar la siguiente funcionalidad:

1. Dada la siguiente estructura DE TIPO Pedidos:

```
typedef struct{
    struct NodoPedido * inicio;
    struct NodoPedido * fin;
} Pedidos;
```

Hacer una función que recorra el archivo dado ("registroCompraCliente.dat") y cree una **TDA Fila de NodoPedido implementada con listas simples**.

- Para pasar los datos **del archivo al TDA Fila**, que debe inicializarse previamente, deberá: Abrir el archivo, recorrerlo, crear los nodos y agregarlos a la Fila. La función de agregar a la Fila tiene como parámetro: **La fila**, y los **datos** para crear **el nodo pedido**.
 - Para trabajar con la lista, tiene que desarrollar las funciones que administren la misma, a saber: Inicializar la lista, Crear Nodo, Agregar al final.
2. Hacer una función que pida al usuario del sistema el ingreso de PEDIDOS, todos los que el mismo quiera, directamente a la FILA.
 - Para cumplir con tal propósito debe hacerlo directamente utilizando el TDA Fila.
 - La función le debe ir solicitando al usuario la carga de los datos del Pedido y debe ir agregandolos al TDA Fila.
 - Invoque las funciones que considere necesarias.

UTN FRMDP - TUP - Laboratorio 2
Recup Primer Parcial - 28 de octubre de 2020

3. Hacer una función que sume los **MontoCompra** de cada cliente y retorne el Total de Montos de compra de toda la lista. La función
 - tiene que ser RECURSIVA.
 - y recibir como parámetro la lista.
4. Hacer una función que **elimine un cliente** del **TDA Fila**.
5. Hacer una función que recorra la lista de **Pedidos** (NodoPedido) y los copie a un nuevo archivo de **RegistroCompraCliente**, aquellos cuyo monto de compra sea mayor a \$ 5.000,00.
 - La función debe trabajar directamente con la lista simplemente vinculada.
6. Hacer una función que muestre **la Fila de Pedidos**.
7. Hacer una función main() que invoque a los modulos anteriores y demuestre el funcionamiento del programa.
 - 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.
 - Si lo considera, cree un menú de opciones para ejecutar cada función o subprograma.
 - A fin de identificar cada inciso, comente su código indicando a qué apartado corresponde, por ejemplo: // Apartado 3

El desarrollo de la función main es de carácter obligatorio y sumará puntos extras en caso de necesitarlos.

En todo momento **recuerde trabajar de forma tal que cada estructura administre sus datos de acuerdo a su responsabilidad.**

Apartado	1	2	3	4	5	6	7	
							main	compila
Puntaje	20	20	15	20	15	10	Si/No	Si/No

Tabla de puntuación:

Obtenido	10	20	30	40	50	60	70	80	90	100
Nota	1	2	3	4	5	6	7	8	9	10
Condición	Desaprobado					Aprobado				