IMPORTANTE:

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

Como parte de una pasantía en la UTN FR Mar del Plata, se nos ha encomendado realizar un pequeño módulo para organizar información existente en un archivo de Estudiantes de la Tecnicatura Universitaria en Programación. El archivo consta de la siguiente estructura de datos:

```
typedef struct {
    int legajo;
    char nombre[20];
    char apellido [20];
    char dni[10];
    int edad;
    char genero;
    int cantMateriasAprobadas; /// total de materias aprobadas del estudiante en la carrera (máximo 20)
    float promedioSinAplazos; /// indica el promedio de estudiante, sin tener en cuenta aplazos float
    float promedioConAplazos; /// indica el promedio de estudiante, teniendo en cuenta aplazos
} estudiante;
```

A partir de dicha información, tendrá que crear el nodo para gestionar una lista simplemente enlazada. El nombre debe ser nodoEstudiante y los tienen que ser los campos del tipo de dato estudiante.

typedef struct

} nodoEstudiante;

Contamos con el siguiente código desarrollado por un pasante que ya no se encuentra en el proyecto:

```
nodoEstudiante * inicListaEstudiante ()
{
return NULL;
}
```

UTN FRMDP - TUP - Laboratorio 2 - Primer Parcial

Se nos pide desarrollar la siguiente funcionalidad:

- 1. Utilizando el archivo que se le entregó ("EstudiantesTUP.dat"), hacer una función que recorra el mismo de forma secuencial y construya una lista de estudiantes. Para realizar esto, deberá **modularizar** y codificar las funciones:
 - a. Crear nodo, inicializar la lista.
 - **b.** Una función que agregue un nuevo nodo <u>al final</u> de una la lista.
- 2. Realizar una función que muestre la lista cargada previamente. La misma deberá estar **modularizada** de acuerdo con la responsabilidad de cada estructura de datos.
- 3. Se desea **averiguar qué porcentaje de estudiantes** poseen un promedio superior a un valor determinado. Para lo cual deberá realizar:
 - **a.** Una función que retorne la cantidad de estudiantes que tiene la lista.
 - **b.** Una función que averigüe, que cantidad de estudiantes poseen un promedio con aplazos superior a un valor determinado.
 - **c.** Un subprograma, que deberá invocar a las funciones anteriores (3.a y 3.b), calcule porcentaje e informe por pantalla los resultados obtenidos.
- 4. <u>Se quiere obtener a los 10 mejores promedios</u> de la lista de estudiantes, con el fin de seleccionar a los mismos para un programa de ayudantías. Dichos estudiantes deberán alojarse en una lista doble. Para lo cual deberá realizar:
 - **a.** Hacer una función que <u>busque</u> al estudiante de mayor promedio sin aplazos en la lista y retorna el legajo de este.
 - **b.** Hacer una función que <u>elimine</u> el nodo de la lista que coincida con el legajo pasado por parámetro.
 - La función recibe un puntero doble a la lista
 - y retorna una estructura de tipo estudiante.
 - c. Una función que <u>pase los estudiantes encontrado en la lista a una lista doblemente vinculada de estudiantes</u>, ordenándolos según el promedio sin aplazos (de mayor a menor).
- 5. Realizar una función que muestre la lista doblemente vinculada obtenida del punto 4.
 - La función recibe la lista doble COMO PUNTERO DOBLE.
 - Esta función deberá mostrar el arreglo al revés.
- 6. Hacer una función que <u>recorre la lista creada</u> y cargada en el punto 1 <u>y pase a una</u>

 <u>Pila los legajos de los estudiantes que hayan aprobado más de 10</u>

 <u>materias</u> de la carrera.
- La función debe retornar la cantidad total de estudiantes que se encuentran en esta situación.

UTN FRMDP - TUP - Laboratorio 2 - Primer Parcial

- 7. Hacer una función main() que invoque a los módulos 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.a

Apartado	1		0	3			4			_	6	7	
	а	b	2	а	b	O	а	b	С	5	а	Todas las llamadas	Compila
Puntaje	5	15	5	5	8	7	5	10	10	5	15	5	5

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		Des	aprob	ado		Aprobado					