

UTN – TUP		
Laboratorio 1 - Segundo Parcial Junio 2024	Apellido y Nombre	Nota

- Deberán generar un proyecto nuevo donde dejarán en la primera línea de código su nombre, apellido y número de comisión.
- Añadir comentarios a su código identificando cada ejercicio. Toda función no identificada, no será corregida.
- Si una función está completamente comentada, no se corregirá.
- Recordar que todo proyecto debe tener 1 solo main en donde se invoquen a las funciones correspondientes. Todo examen que tenga más de 1 main, será desaprobado automáticamente.

Dadas las siguientes estructuras:

```
#define CAPACIDAD 7 ///capacidad de pasajeros por viaje
```

```
typedef struct
{
    int idViaje;
    char origen[50];
    char destino[50];
    int cantidadPasajeros;
} Viaje;
```

```
typedef struct
{
    int idPasajero;
    int idViaje;
    char nombre[50];
    char dni[10];
    int despacheEquipaje; ///1 o 0
} Pasajero;
```

Obtenido	Valor	Inciso
	10	1. Crear la/las funciones necesarias para cargar un archivo de Viajes y otro de Pasajeros.
	10	2. Realizar una función que retorne la cantidad de registros cargados en el archivo de viajes. <b>No</b> se debe recorrer todo el archivo ni utilizar contadores.
	20	3. Realizar una función que reciba por parámetro un número de <b>id</b> y pase a todos los <b>pasajeros</b> que correspondan a dicho viaje a un arreglo <b>dinámico</b> . Pensar muy bien las funciones auxiliares que serán necesarias para trabajar la dimensión del arreglo.
	20	4. Realizar una función que muestre todos los <b>viajes</b> disponibles en el archivo de forma <b>recursiva</b> . <b>Disponibles</b> : aquellos cuya cantidad de pasajeros esté por debajo de la constante CAPACIDAD.
	20	5. Realizar una función que permita modificar <b>uno o más datos</b> de un registro <b>específico</b> dentro del archivo de pasajeros. La función recibe por parámetro el <b>número</b> de registro.
	10	6. Hacer una función que pase los pasajeros cargados en el arreglo <b>dinámico</b> del punto 3 a <b>dos</b> archivos: uno para los que <b>despachan equipaje</b> y otro para los que <b>no</b> .
	10	7. Hacer una función <b>main</b> que ejecute las funciones anteriores y demuestre su correcto funcionamiento. Cree las <b>variables</b> que considere necesarias e <b>invoque las</b> funciones como corresponde en cada caso. <u>Muestre por pantalla los resultados.</u>

**IMPORTANTE:** Se debe modularizar en **TODAS** las funciones generadas. Se deben aplicar buenas prácticas (identificadores descriptivos, uso de retornos, etc). La prolijidad del código será considerada al momento de corregir el examen.

¡Éxitos!