

2do Parcial**IMPORTANTE:**

- Realizar todas las funciones que se indican.
- Añadir su nombre y apellido al “main.c” y comentarios a su código identificando cada inciso.
- Si no se encuentra en Zoom durante el desarrollo del examen, la evaluación no será validada y por lo tanto su nota final será **Ausente**.

Enunciado

Se tiene un archivo con registros de los pasajeros de distintos aviones en el día de hoy, la idea es desglosarlo en las estructuras que se encuentran en el esquema debajo.

Recordad que el archivo una vez abierto y leído todos sus datos, no os volveréis a abrir, deberán trabajar con los incisos sobre las estructuras compuestas aclaradas.

- El archivo está ordenado por “idAvion” creciente.
- Un pasajero es único, es decir, lo encontrarán en un avión solo y en ninguno más.

“pasajerosAviones.bin”	nodoAvion	nodoPasajero
<pre>typedef struct {     char dni[9];     char nombre[30];     float pesoEquipaje;     int edad;     int idAvion;     int idPasaje; }registroPasajero;</pre>	<pre>typedef struct nodoAvion {     int idAvion;     nodoPasajero* pasajeros;     struct nodoAvion* sig; }nodoAvion;</pre>	<pre>typedef struct nodoPasajero {     Persona p;     int idPasaje;     float pesoEquipaje;     struct nodoPasajero* sig; }nodoPasajero;</pre>
		<p><b>Persona</b></p> <pre>typedef struct {     char nombre[30];     char dni[9];     int edad; }Persona;</pre>

Obtenido	Valor	Inciso
	5	<ul style="list-style-type: none"> <li>• Hacer una función main () que invoque a sus incisos y demuestre el correcto funcionamiento del programa.</li> </ul>
	25	<p><b>1. Cargar los datos desde el archivo</b>  Hacer una función o funciones <i>que inserten en la lista de aviones el avión correspondiente, así como en la lista de pasajeros, cada pasajero que coincida con su avión correspondiente.</i></p> <ul style="list-style-type: none"> <li>• La inserción de un avión en la lista de aviones se da al final.</li> <li>• La inserción en la lista de pasajeros se da en orden por “DNI” creciente.</li> </ul> <p><b>Nota:</b> Se debe insertar ordenado, y no ordenar la lista luego de haberse cargado.</p>

	15	<b>2.</b> Realizar una función que retorne el idAvion con la máxima cantidad de pasajeros.
	15	<b>3.</b> Realizar una función que informe el Peso total en Equipaje en un idAvion X recibido por parámetro.
	25	<b>4.</b> Realizar una función que, recibiendo por parámetro un idAvion X, elimine aquellos pasajeros que su edad sea mayor a 50. Si el avión se queda sin pasajeros (lista de pasajeros nula), elimine también ese nodo de avión.
	15	<b>5.</b> Realizar una función que pase a archivos las personas de cada uno de los aviones. Ej: idAvion1.bin -> todos los pasajeros del idAvion =1 idAvion2.bin -> todos los pasajeros del idAvion = 2  Puede ser cualquier nombre, no necesariamente debe ser como el ejemplo.
		NOTA FINAL

**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				