

UTN – FRMDP – TUP	Nombre y Apellido	Nota
Laboratorio 2 - Primer Parcial – Comisión 10 Septiembre 2022		

IMPORTANTE:

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

Sean las siguientes estructuras:

/// estructura jugador

typedef struct

```
{
    int dni;
    char nombre_apellido[30];
    int edad;
    char equipo[30];
```

} player;

/// estructura nodo

```
typedef struct _nodoPlayer
{
    player p;
    struct _nodoPlayer * sig;
} nodoPlayer;
```

Y la función:

```
nodoPlayer * inicLista ()
{
    return NULL;
}
```

El código básico que se le entregó, carga automáticamente un arreglo de jugadores. Utilice esa información como origen de datos para realizar las funciones que se detallan a continuación:

Obtenido	Valor	Inciso
	10	1) Hacer una función que muestre un arreglo de “players” en forma recursiva, desde el inicio hasta el final del arreglo. Modularizar.
	5	2) Hacer una función que cree un nodoPlayer, cargue sus datos y retorne un puntero al mismo. Piense que parámetro/s convendría que reciba esta función.
	20	3) Hacer una función que inserte un nuevo nodo a la lista, manteniendo el orden creciente por “dni” . Esta función debe recibir un puntero doble a la lista y el nuevo nodo a insertar , utilizar el siguiente prototipado: void insertInOrder (nodoPlayer ** lista, nodoPlayer * nuevo)
	15	4) Hacer una función que pase los “players” del arreglo que ya está cargado a una lista , invocando las funciones de los incisos 2 y 3. O sea, debe recorrer el arreglo, crear los nodos y agregarlos en orden a la lista. La función recibe el arreglo cargado, la lista vacía y retorna la lista cargada.
	10	5) Hacer una función que muestre los “players” mayores a 25 años de una lista en forma recursiva, pero de forma invertida (del último al primero). Modularizar.
	15	6) Hacer una función que busque al player de mayor edad en la lista y retorna el dni del mismo. Puede ser iterativa o recursiva.
	20	7) Hacer una función que pase los “players” de la lista que se encuentra cargada a una nueva lista , pasando únicamente los de un determinado equipo (dato que se recibe por parámetro) . Al encontrar un player que cumpla con la condición, crear un nuevo nodo a partir de sus datos e insertarlo en la nueva lista (utilice las funciones 2 y 3). nodoPlayer * pasarDeListaToLista (nodoPlayer * listaVieja, nodoPlayer * listaNueva, char equipo [])

	5	8) Hacer una función main () que invoque a las funciones anteriores y demuestre el correcto funcionamiento del programa. Para hacer esto, cree las variables que considere necesarias e invoque las funciones como corresponde en cada caso.
--	---	--

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				