

**IMPORTANTE:**

- Crear un proyecto con su Nombre y Apellido.
- Realizar todas las funciones que se indican.
- Añadir comentarios a su código identificando cada inciso. **MUY IMPORTANTE!!!!**
- Se entrega librería con funciones GENERICAS, que podrá adaptar (modificar) según lo crea conveniente para luego usarlas en el desarrollo.

Dentro de un sistema de administración de un Torneo de Fútbol, se nos pide desarrollar una **LISTA DE ARBOLES**, codificando todas las funciones necesarias para su administración, utilizando las siguientes estructuras de datos:

Lista de Equipos	Tipo de Dato Equipo
COMPLETAR LAS ESTRUCTURAS DE TIPO NODO QUE CORRESPONDAN	<pre>typedef struct{     int idEquipo;     char nombreEquipo[30]; }stEquipo;</pre>
Árbol de Jugadores	Tipo de dato Jugador
COMPLETAR LAS ESTRUCTURAS DE TIPO NODO QUE CORRESPONDAN	<pre>typedef struct{     int idJugador;     int nroCamisetaJugador;     char nombreJugador[30];     int puntosAnotados; }stJugador;</pre>

Debemos tener en cuenta que la información se organiza de la siguiente manera:

- El sistema tiene que agregar jugadores e ir organizándolos en sus equipos correspondientes.
- Un jugador no puede estar en 2 equipos diferentes.
- Para cumplir con tal propósito iniciaremos el sistema cargando los datos a partir de un archivo binario que se le entrego.

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

**Se nos pide:**

1- Pasar todos los datos del **archivo** a la **lista de árboles**. Pensar y desarrollar las funciones que necesita para resolver este problema.

Tenga en cuenta que el archivo entregado responde a la siguiente estructura de datos:

```
typedef struct{
    int idEquipo;
    char nombreEquipo[30];
    int idJugador;
    int nroCamisetaJugador;
    char nombreJugador[30];
    int puntosAnotados;
}stJugadorEquipo;
```

2- Realizar las funciones necesarias para realizar el **ALTA** de jugadores al sistema(AltaJugador). Esta función debe interactuar con el usuario, solicitando la información para generar **un nuevo registro en la lista de árboles**.

Los pasos para seguir son los siguientes para agregar un jugador al equipo correspondiente:

- Cargar un jugador
- Preguntar por el equipo al que pertenece
- Buscar si el equipo existe
- Si el equipo no existe
  - Agregarla a la lista de equipos
- Si el equipo existe
  - La función de búsqueda devolverá el equipo
- Agregar al árbol de jugadores de ese equipo, a ese jugador

3- Realizar una función que muestre los equipos con sus correspondientes jugadores (in order). Tenga en cuenta la **modularización** y la **responsabilidad** de cada estructura.

4- Realizar una función **que pase a una lista de jugadores a los goleadores de cada equipo**.

- En el caso de haber más de 1 jugador con el mismo puntaje, enviarlos a todos ellos.
- Mostrar la lista de goleadores en el main()
  - Datos a mostrar: nombre del equipo, nombre del jugador y cantidad de puntos anotados.

5- Realizar una función (o varias) que determine e informa por pantalla; **por cada equipo**:

- Jugador que más puntos anoto.
- Promedio de puntos anotados por cada equipo.

6- Realizar una función (o varias) que **pase todos los jugadores del torneo** que tengan **un determinado número de camiseta** a un archivo binario y los guarde.

- Desarrollar la función principal con el siguiente prototipo:
  - void pasarDeLaListaAlArchivo (lista de equipos, nro de camiseta).
- El archivo saliente debe ser del tipo **stJugadorEquipo**

7- Realizar 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 los datos a mostrar en cada caso NO fueron especificados, decida según su criterio, cuales mostrar.
- 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 y USE NOMBRES DESCRIPTIVOS PARA FUNCIONES Y VARIABLES.

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

Apartado	1	2	3	4	5	6
Puntaje	35	10	10	15	15	15

**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				