**Consigna:**

Vamos a programar un juego de adivinanzas, guardar los puntajes y armar una tabla de puntajes con todos los jugadores

| **typedef struct nodoArbol{**  **int dato ;**  **struct nodoArbol \* izq;**  **struct nodoArbol \* der;**  **}nodoArbol;** |
| --- |

Los números para adivinar los cargaremos al árbol.

* El usuario debe elegir un nivel de dificultad que va determinar la cantidad de números que se cargarán.
* Según la cantidad de números, los mismos se deben generar de manera aleatoria (de 1 a 100) pero sin repetir.

| **typedef struct listaPuntajes{**    **struct listaPuntajes\* sig;**  **int puntos;**  **}listaPuntajes ;** | **typedef struct nodoJugador{**  **char nombreJugador[20];**  **struct nodoJugador\* siguiente;**  **struct listaPuntajes\* puntajes;**  **}nodoJugador;** |
| --- | --- |

* Le preguntamos al usuario su nombre. Si el nombre NO está, creamos un nodo del tipo **nodoJugador** y su lista de **puntajes** vacía. Si el nombre SI está, recuperamos el nodo completo porque tenemos su lista de puntajes anteriores.
* Buscamos (de algún modo) un número dentro del árbol y ese será el número a adivinar.
* Contamos la cantidad de veces que tarda el usuario en llegar a nuestro número. Ese será su puntaje. En cada intento, le decimos si el número es mayor o menor.
* Cuando tenemos el puntaje, creamos un nodo del tipo **listaPuntajes** y lo insertamos en el atributo **puntajes** del nodo que creamos -o recuperamos- en la primera consigna.
* El juego termina cuando se adivina el número y ahi reiniciamos el proceso con otro o el mismo jugador.
* En algún momento tenemos que finalizar el juego y mostrar la tabla de puntuación.
  + Sumamos los **puntos** de cada **listaPuntajes** de cada **nodoJugador.**
  + Buscamos el menor de los puntos de todos los jugadores.
  + El menor de todos, es el jugador ganador!
* A divertirse!