Parcial 1 - Algoritmos I Taller: Tema A

Debés entregar el código completo en los campos correspondientes de cada ejercicio del formulario de entrega y subir el archivo .hs correspondiente. Este código debe poder ejecutarse en haskell sin errores. Te recomendamos para ello que pruebes con diferentes ejemplos antes de entregar.

Descripción del problema

Existen diferentes juegos relacionados con tenis que se pueden jugar. En particular, rey de la cancha y single-dobles. Rey de la cancha es un juego donde un jugador defiende su lugar ganando puntos, mientras los demás intentan quitarle la posición. Single-Doble es un juego de cuatro jugadores donde se juega los punto de individuales al mismo tiempo con dos pelotas, el equipo que logra ganar las dos pelotas en juego gana un punto para su equipo.

Ejercicio 1:

Definir el tipo `JuegoTenis` que consta de dos constructores `ReyDeLaCancha` y `SingleDoble` con los siguientes parámetros:

- El constructor `ReyDeLaCancha` debe tomar como parámetros el nombre del rey, cuantos puntos tiene, el nombre del primer contrincante, los puntos del primer contrincante, el nombre del segundo contrincante y sus puntos.
- El constructor `SingleDoble` debe tomar como parámetros el nombre del primer equipo, sus puntos, el nombre del segundo equipo y sus puntos

Ejercicio 2:

A partir del tipo definido en el punto anterior, definí los siguientes partidos:

```
rey1 :: JuegoTenis
rey1 = <COMPLETAR>
```

correspondiente al formato de juego Rey de la Cancha, donde el rey Guillermo lleva 5 puntos y sus contrincantes Mario y Cristina tienen 2 y 3 respectivamente.

```
singleDoble1 :: JuegoTenis
singleDoble1 = <COMPLETAR>
```

correspondiente al formato de juego Single-Doble, donde el equipo Fed tiene 10 puntos y su equipo contrincante Nad tiene 6 puntos.

Ejercicio 3:

Definir la función `estaReyGanando :: JuegoTenis -> Bool` que dado un `JuegoTenis`, devuelve `True` si el rey tiene mas puntos que sus contrincantes, `False` caso contrario.

Definir la función `juegoTerminado :: JuegoTenis -> Bool` que dado un `JuegoTenis`, devuelve `True` si el juego esta terminado. En el caso del rey de la cancha, esto ocurre cuando el rey llega a 15 puntos. En el caso de single-doble, alcanza con que alguno de los equipos llegue a 21 puntos.

Ejercicio 4:

Definir la función `cantidadJuegosTerminados :: [JuegoTenis] -> Int` que dada una lista de juegos de tenis, devuelve la cantidad de juegos que estan terminados.

Ejercicio 5:

Dado el tipo recursivo `ColaCanchas` definido de la siguiente manera

```
data ColaCanchas = Vacia | Encolada JuegoTenis ColaCanchas deriving Show
```

Definir la función `colaDeUnEquipo :: ColaCanchas -> String -> ColaCanchas` que dada una cola de juegos por jugar, y el nombre de un equipo, devuelve la cola de juegos que tiene solamente los juegos single-doble del correspondiente equipo.