Python Task-1
Chopsticks Game
Deadline - 28th August 2021, Saturday, 11:59 PM

Rules of the Game (2 Player Version):

This official set of rules is called *rollover* where five fingers are subtracted should a hand's sum exceed 5 as described below.

1. Each player begins with one finger raised on each hand.
2. On a player's turn, they must either *attack* or *split*, but not both.
3. To ***attack***, a player uses one of their **live hands(Hands that have min of 1 Finger Raised)** to strike an opponent's live hand. The number of fingers on the opponent's struck hand will increase by the number of fingers on the hand used to strike.
4. To ***split***, a player strikes their own two hands together, and transfers raised fingers from one hand to the other as desired.
5. If any hand of any player reaches exactly **five fingers, then the hand is *killed***, and this is indicated by raising zero fingers (i.e. a closed fist).
6. A player may ***revive*** their own dead hand using a split, as long as they abide by the rules for splitting.
7. However, players may not revive opponents' hands using an attack. Therefore, a player with two dead hands can no longer play and is eliminated from the game.
8. If any hand of any player reaches five or more fingers, then it is considered a **"dead hand"**.
9. **A player wins once all opponents are eliminated i.e by each having two dead hands at once.**

Your Task - Design the above game in Python with the following specifications:

1. The Program starts with the Player1 & Player2 having **one finger** raised on both hands.
2. Player 1 goes first - The Program asks for input from the user in the form of :
    a. Move
    b. Combination of the moves

    (See player input section for more details)
3. After Player 1 has given the input, the program asks for input from Player 2 in the similar way.
4. After each turn, the program displays the current status of hands for both players.
5. The game ends when one of the players has dead hands. Display which player won the game.

Input Specification:

On any player's turn, the program asks for:

1. **Move**
   Either A (for Attack) or S(for Split)

2. **Combination**
   In case of Attack - **<Attack from> <Attack to>**
   Examples
   1. L R # Attack from Left to Right of Opposition
   2. R R # Attack from Right to Right of Opposition

   In case of Split - **<Hand to be Splitted> <Left hand after Split> <Right hand after Split>**
   Examples
   1. L 1 1 # Split Left hand into 1(Left Hand) & 1(Right Hand)
   2. R 3 1 # Split Right hand into 3(Left Hand) & 1(Right Hand)

Output Specification:
The first number displays the number on the left hand while the second number displays the number on the right hand
(It is so as to avoid confusion during input)

Notes:
- Program must be fully modular or class based. The Program must either be module based or classed based according to your preference.
- Program must be submitted before deadline
- Code should not be plagiarized

Example Run :
Current Status :
Player1 - 1 1
Player2 - 1 1

Enter move for Player 1 - A (Player Input)
Enter the move combination - L R (Player Input) #Which hand to attack
Current Status
Player1 - 1 1
Player2 - 1 2

Enter move for Player 2 - A (Player input)
Enter the move combination- L R (Player input) #Which hand to attack
Current Status
Player1 - 1 2
Player2 - 1 2

Enter move for Player 1 - A (Player input)
Enter the move combination - L L (Player input) #Which hand to attack
Current Status
Player1 - 1 2
Player2 - 2 2

Enter move for Player 2 - A (Player input)
Enter the move combination- R L (Player input) #Which hand to attack
Current Status
Player1 - 3 2
Player2 - 2 2

Enter move for Player 1 - A (Player input)
Enter the move combination- L R (Player input) #Which hand to attack
Current Status
Player1 - 3 2
Player2 - 2 0

Enter move for Player 1 - S (Player input)
Enter the move combination - L 1 1 (Player input) #How to split
Current Status
Player1 - 3 2
Player2 - 1 1

And the game goes on until one of the players wins …..