

Question #2: Boxing Match Simulation

Objective: Function

A boxing match is a fight between two boxers, Red and Blue corners, over multiple rounds. Boxers are scored on a 10-point system.

Rules for Modified 10-point system:

1. **Initial Score:** Each boxer starts with 10 points in each round. Their points will be deducted according to the following conditions.
2. **For “knockdown”:** each knockdown will be 1 point deducted. The minimum point is 0.
3. **Round winner:** The boxer with higher point in a round wins that round. If both boxers have the same point, the round is considered as tie. The round winner is recorded as 'R', 'B', or 'T'.
4. **Match winner:** The total point of each boxer across all rounds is calculated. The boxer with the higher score wins the match. if the total scores are equal, the match is considered as **draw**. The match winner is recorded as 'Red', 'Blue', or 'Draw'.

Task:

You need to implement three functions:

1. **parse_boxing_match(match_result):**

match_result is a string begins with 'RB' or 'BR' follow by even numbers of integer number, i.e.

'RB 0 1 2 0 1 0 0 0'

This function return two list for red and blue corner. (see example)

2. **calculate_round_points(r_knockdown, b_knockdown):**

r_knockdown, b_knockdown are the number of knockdowns for red corner boxer and blue corner boxer respectively.

This function returns a list [red_point, blue_point] calculated from 10-point system rules for knockdown.

3. **show_match_result(red_point, blue_point):**

This function takes two parameters: red_point and blue_point, lists, then, shows three lines:

First line is round stat, 'R' for red win, 'B' for blue win, and 'T' for a tie.

Second line is the total points of each boxer

Third line is the match result. Boxer with more points win the match. Otherwise, it is a 'Draw'

INPUT

Python function call

OUTPUT

As shown in examples

EXAMPLES

Input (from keyboard)	Output (on-screen)
<code>print(parse_boxing_match('RB 0 1 2 0 1 0 0 0'))</code>	<code>[0, 1, 2, 0], [1, 0, 0, 0]</code>
<code>print(parse_boxing_match('BR 0 1 2 0 1 0 0 0'))</code>	<code>[1, 0, 0, 0], [0, 1, 2, 0]</code>
<code>print(calculate_round_points([0, 0, 0, 0], [0, 0, 0, 0]))</code>	<code>[10, 10, 10, 10]</code> <code>[10, 10, 10, 10]</code>
<code>print(calculate_round_points([0, 1, 2, 1], [5, 0, 0, 0]))</code>	<code>[10, 9, 8, 9]</code> <code>[5, 10, 10, 10]</code>
<code>show_match_result([10, 9, 8, 9], [5, 10, 10, 10])</code>	<code>RBBB</code> <code>36 35</code> <code>Red</code>
<code>show_match_result([5, 10, 10, 10], [10, 9, 8, 9])</code>	<code>BRRR</code> <code>35 36</code> <code>Blue</code>
<code>show_match_result([9, 7, 10, 8], [7, 9, 10, 8])</code>	<code>RBTT</code> <code>34 34</code> <code>Draw</code>

Test Cases in Grader

Testcases will be grouped. Each group has the following criteria:

Testcases quantity	Testcase characteristics
50%	A mix of wins, losses, and ties for both corner.
25%	Every round is a tie and match draw
25%	One corner wins every round

Source Code Template

```
def parse_boxing_match(match_result):
    # parse red and blue knockdowns from match result

    return [red_knockdown, blue_knockdown]

def calculate_round_points(red_knockdowns, blue_knockdowns):
    # calculate points for each boxer

    return red_points, blue_points

def show_match_result(red_points, blue_points):
    # display the result

exec(input().strip()) # Don't modify this line
```