Peer Review

Raj Keshav's Approach:

- **Ques1: Bucketise the given array[Double] into buckets having range interval (x, x+0.049).**
    - Used map to create the bucket boundaries.
    - Each bucket has a width of 0.049 and the sequence ranges from 0 to 2000. The bucket boundaries are stored in the buckets variable.
    - Next, mapped each element of the input array to a specific bucket using the map function to apply a function to each element of the input array.
    - The function takes a single argument num, which represents the value of the current element being processed.
    - The function computes the index of the bucket that num belongs to using integer division by 0.049. This index is then used to look up the corresponding bucket boundaries in the buckets sequence.

- **Ques2: Queries based on the given players statistics.**

    - Defined a case class Player with fields for year, name, country, matches, runs, and wickets. then created a sample sequence of Player instances for demonstration purposes.
    - The player with the highest run scored is found by calling the maxBy method on the players sequence with the runs field as the key.
    - The top 5 players by run scored are found by calling the sortBy method on the players sequence with the runs field as the key, in descending order and printing the first 5 elements of the resulting sequence .
    - The top 5 players by wickets are found by calling the sortBy method on the players sequence with the wickets field as the key, in descending order and then printing the first 5 elements of the resulting sequence.
    - Ranked the players by overall performance, defined as runs + (wickets * 5)by mapping each player to a tuple of the player instance and their overall performance score, then sorting the resulting sequence of tuples by the performance score in descending order.
    - Printing out the rank, name, number of runs scored, number of wickets taken, and overall performance score for each player using the zipWithIndex method to add a rank to each tuple.

Pankaj Kumar Singh's Approach:

- **Ques1: Bucketise the given array[Double] into buckets having range interval (x, x+0.049).**
    - Used the function get_bucket to find the bucket, which got the last two digits after the decimal
    - If it is smaller than the 50 , it falls in the bucket 00-49 bucket category
    - Else it falls in 50 - 99 bucket category
    - Concatenated the two values and returned
    - Function could have been created directly in the object instead of creating a class. Class instances take more memory here.
    - Code repetition could have been avoided in the if else part.

- **Ques2: Queries based on the given players statistics.**
    - Case class is a good option, doesnt require to create instance object.
    - Could have used sortby function for ranking different players, instead of using a map which increases the complexity.
    - Output formatting and code comments makes the solution easy to understand.