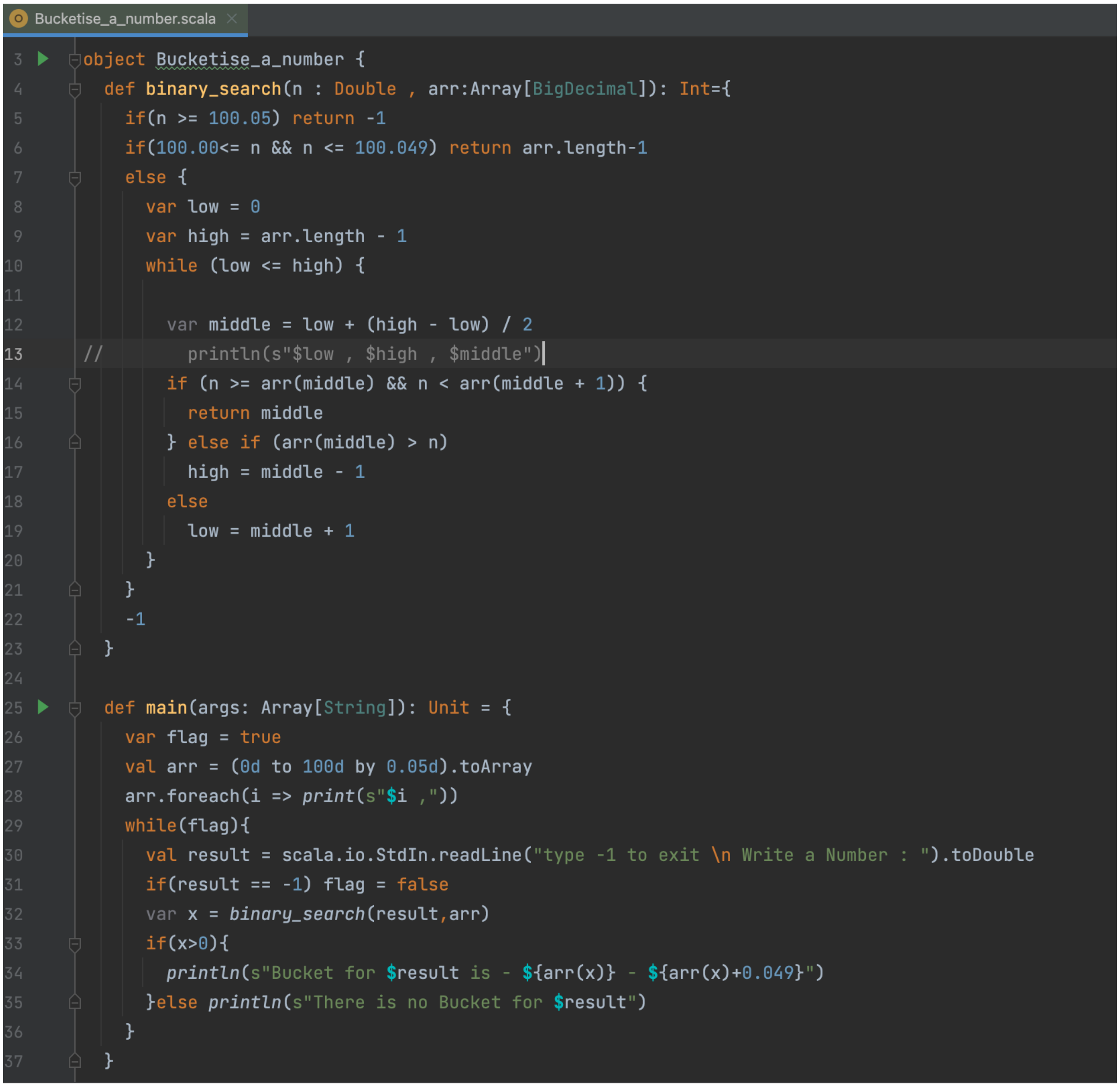
Scala Assignment

Name - Satyaprakash
Employee ID - TAS059 Date - 6 Jan 2022

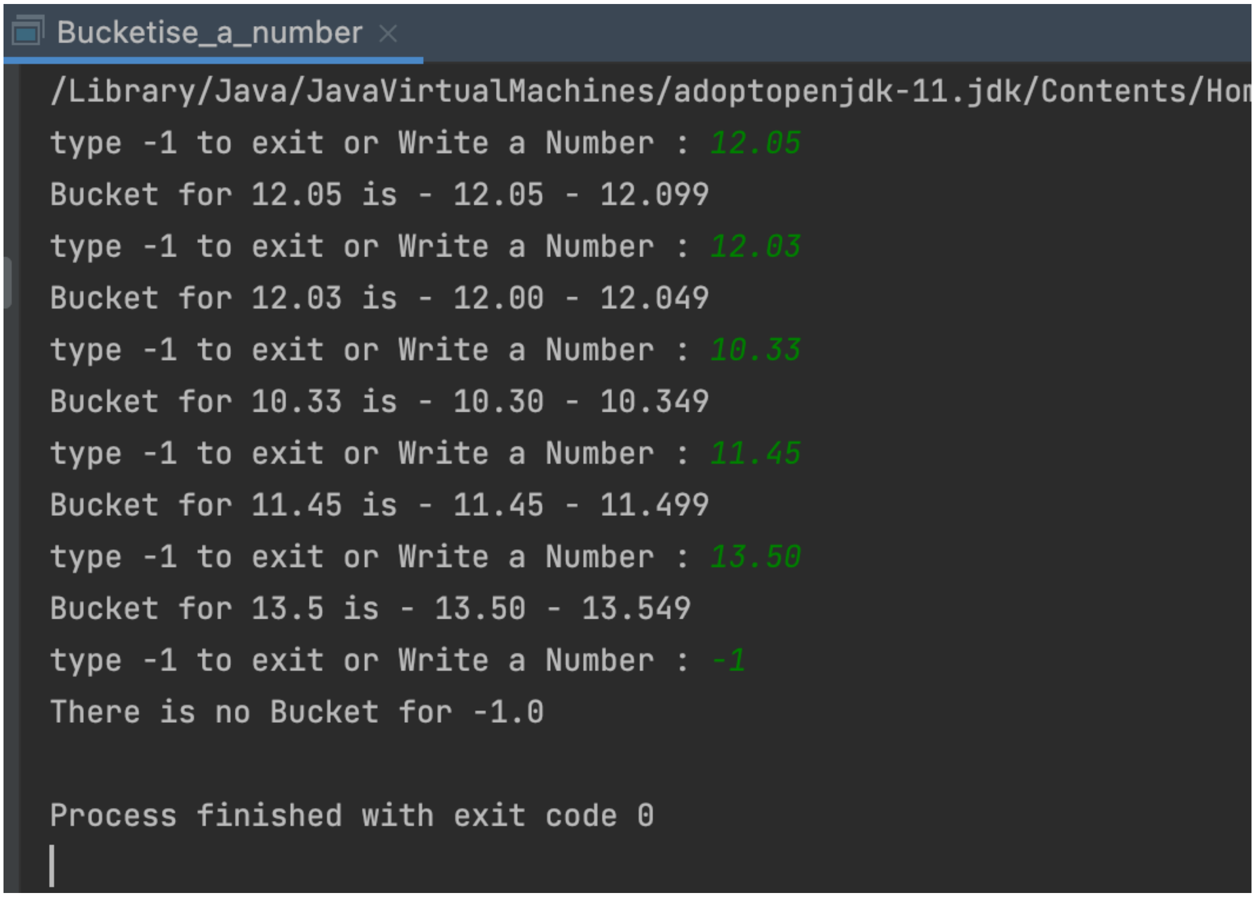
1. Bucketise the given array[Double] into buckets having range interval (x, x+0.049).
0.000 - 0.049
0.050 - 0.099
0.100 - 0.149
...
...
100.000 - 100.049

Sample -
12.05, 12.03, 10.33, 11.45, 13.50
Output- [12.050-12.099, 12.050-12.099, 10.300-10.349, 11.450-11.499, 13.500-13.549]



```scala
object Bucketise_a_number {
  def binary_search(n : Double , arr:Array[BigDecimal]): Int={
    if(n >= 100.05) return -1
    if(100.00<= n && n <= 100.049) return arr.length-1
    else {
      var low = 0
      var high = arr.length - 1
      while (low <= high) {

        var middle = low + (high - low) / 2
//          println(s"$low , $high , $middle")
        if (n >= arr(middle) && n < arr(middle + 1)) {
          return middle
        } else if (arr(middle) > n)
          high = middle - 1
        else
          low = middle + 1
      }
    }
    -1
  }

  def main(args: Array[String]): Unit = {
    var flag = true
    val arr = (0d to 100d by 0.05d).toArray
    arr.foreach(i => print(s"$i ,"))
    while(flag){
      val result = scala.io.StdIn.readLine("type -1 to exit \n Write a Number : ").toDouble
      if(result == -1) flag = false
      var x = binary_search(result,arr)
      if(x>0){
        println(s"Bucket for $result is - ${arr(x)} - ${arr(x)+0.049}")
      }else println(s"There is no Bucket for $result")
    }
  }
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/adoptopenjdk-11.jdk/Contents/Ho
type -1 to exit or Write a Number : 12.05
Bucket for 12.05 is - 12.05 - 12.099
type -1 to exit or Write a Number : 12.03
Bucket for 12.03 is - 12.00 - 12.049
type -1 to exit or Write a Number : 10.33
Bucket for 10.33 is - 10.30 - 10.349
type -1 to exit or Write a Number : 11.45
Bucket for 11.45 is - 11.45 - 11.499
type -1 to exit or Write a Number : 13.50
Bucket for 13.5 is - 13.50 - 13.549
type -1 to exit or Write a Number : -1
There is no Bucket for -1.0

Process finished with exit code 0
```

2. For given players statistics..
   Found the below -
   1. Player with the best highest run scored.
   2. Top 5 players by run scored.
   3. Top 5 players by wicket taken.
   4. Rank players with overall performance give weight 5x to wicket taken and (5/100)x to run scored.

   Sample -

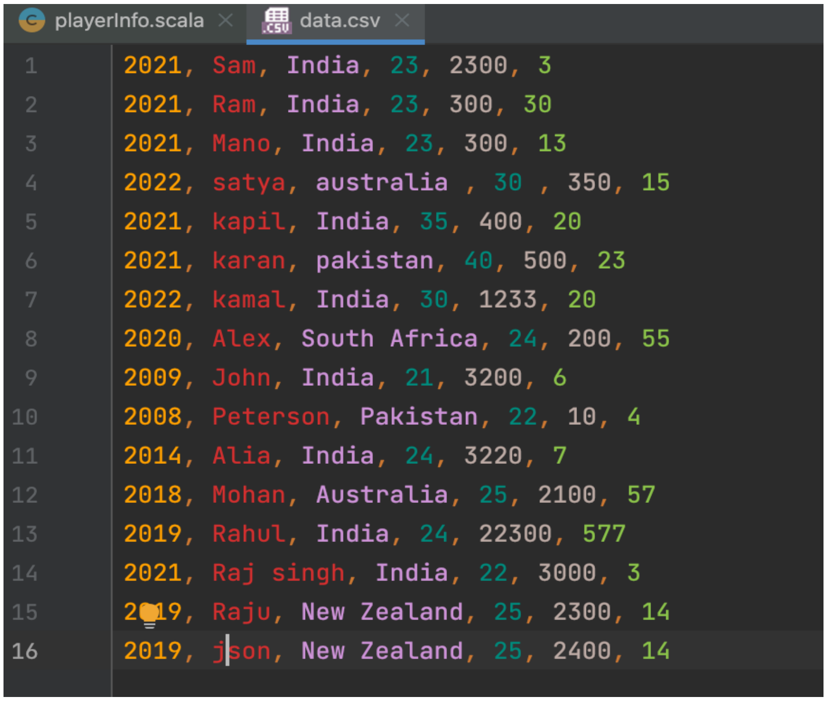
   Year, PlayerName, Country, Matches, Runs, Wickets
   2021, Sam, India, 23, 2300, 3
   2021, Ram, India, 23, 300, 30
   2021, Mano, India, 23, 300, 13

Scala Assignment

Name - Satyaprakash
Employee ID - TAS059

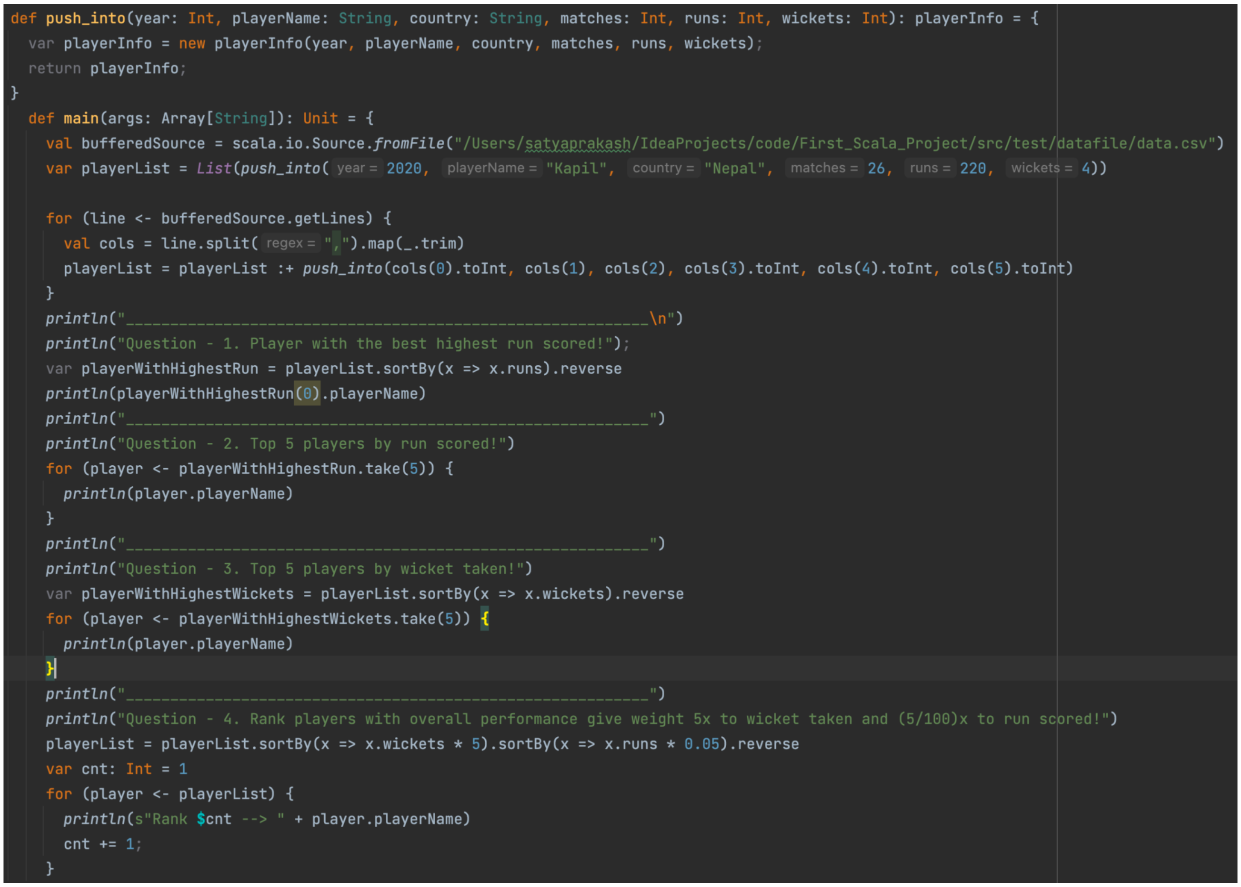
Date - 6 Jan 2022

DATA:



```
2021, Sam, India, 23, 2300, 3
2021, Ram, India, 23, 300, 30
2021, Mano, India, 23, 300, 13
2022, satya, australia , 30 , 350, 15
2021, kapil, India, 35, 400, 20
2021, karan, pakistan, 40, 500, 23
2022, kamal, India, 30, 1233, 20
2020, Alex, South Africa, 24, 200, 55
2009, John, India, 21, 3200, 6
2008, Peterson, Pakistan, 22, 10, 4
2014, Alia, India, 24, 3220, 7
2018, Mohan, Australia, 25, 2100, 57
2019, Rahul, India, 24, 22300, 577
2021, Raj singh, India, 22, 3000, 3
2019, Raju, New Zealand, 25, 2300, 14
2019, json, New Zealand, 25, 2400, 14
```

CODE :



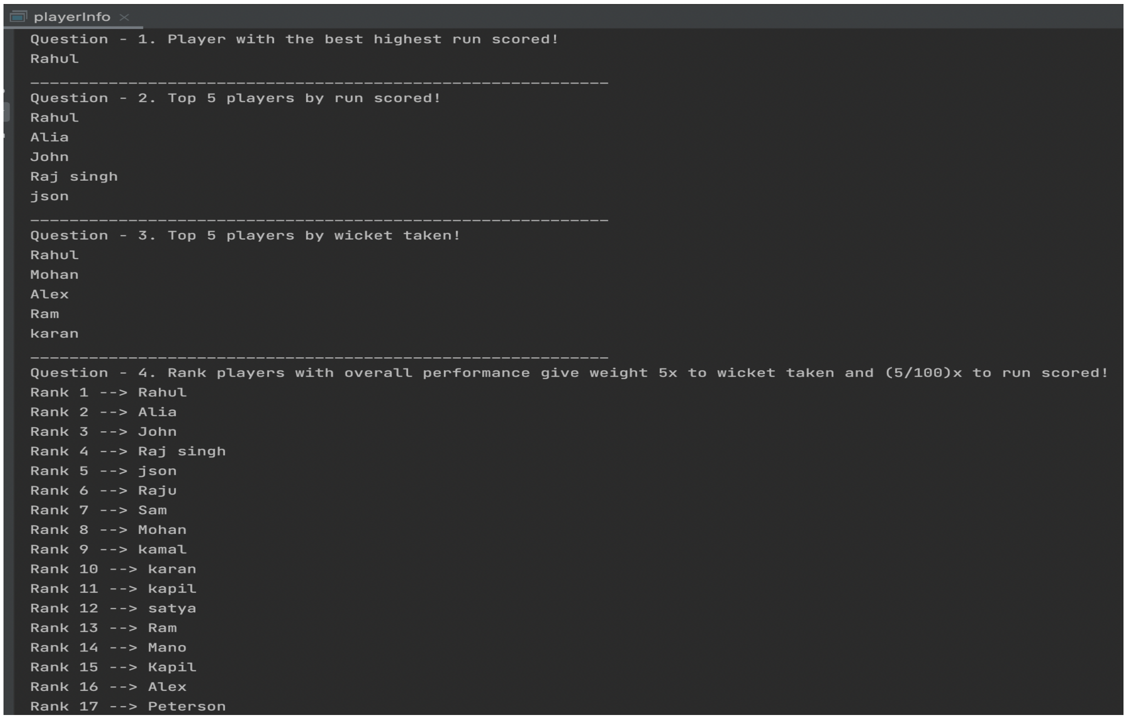
```
def push_into(year: Int, playerName: String, country: String, matches: Int, runs: Int, wickets: Int): playerInfo = {
  var playerInfo = new playerInfo(year, playerName, country, matches, runs, wickets);
  return playerInfo;
}
  def main(args: Array[String]): Unit = {
    val bufferedSource = scala.io.Source.fromFile("/Users/satyaprakash/IdeaProjects/code/First_Scala_Project/src/test/datafile/data.csv")
    var playerList = List(push_into( year = 2020, playerName = "Kapil", country = "Nepal", matches = 26, runs = 220, wickets = 4))

    for (line <- bufferedSource.getLines) {
      val cols = line.split( regex = ",").map(_.trim)
      playerList = playerList :+ push_into(cols(0).toInt, cols(1), cols(2), cols(3).toInt, cols(4).toInt, cols(5).toInt)
    }
    println("___________________________________________________________\n")
    println("Question - 1. Player with the best highest run scored!");
    var playerWithHighestRun = playerList.sortBy(x => x.runs).reverse
    println(playerWithHighestRun(0).playerName)
    println("___________________________________________________________")
    println("Question - 2. Top 5 players by run scored!")
    for (player <- playerWithHighestRun.take(5)) {
      println(player.playerName)
    }
    println("___________________________________________________________")
    println("Question - 3. Top 5 players by wicket taken!")
    var playerWithHighestWickets = playerList.sortBy(x => x.wickets).reverse
    for (player <- playerWithHighestWickets.take(5)) {
      println(player.playerName)
    }
    println("___________________________________________________________")
    println("Question - 4. Rank players with overall performance give weight 5x to wicket taken and (5/100)x to run scored!")
    playerList = playerList.sortBy(x => x.wickets * 5).sortBy(x => x.runs * 0.05).reverse
    var cnt: Int = 1
    for (player <- playerList) {
      println(s"Rank $cnt --> " + player.playerName)
      cnt += 1;
    }
```

Scala Assignment

Name - Satyaprakash
Employee ID - TAS059

Date - 6 Jan 2022

OUTPUT :



```
Question - 1. Player with the best highest run scored!
Rahul
-----------------------------------------------------------
Question - 2. Top 5 players by run scored!
Rahul
Alia
John
Raj singh
json
-----------------------------------------------------------
Question - 3. Top 5 players by wicket taken!
Rahul
Mohan
Alex
Ram
karan
-----------------------------------------------------------
Question - 4. Rank players with overall performance give weight 5x to wicket taken and (5/100)x to run scored!
Rank 1 --> Rahul
Rank 2 --> Alia
Rank 3 --> John
Rank 4 --> Raj singh
Rank 5 --> json
Rank 6 --> Raju
Rank 7 --> Sam
Rank 8 --> Mohan
Rank 9 --> kamal
Rank 10 --> karan
Rank 11 --> kapil
Rank 12 --> satya
Rank 13 --> Ram
Rank 14 --> Mano
Rank 15 --> Kapil
Rank 16 --> Alex
Rank 17 --> Peterson
```