**Name- Rajkeshav Kumar Jha**

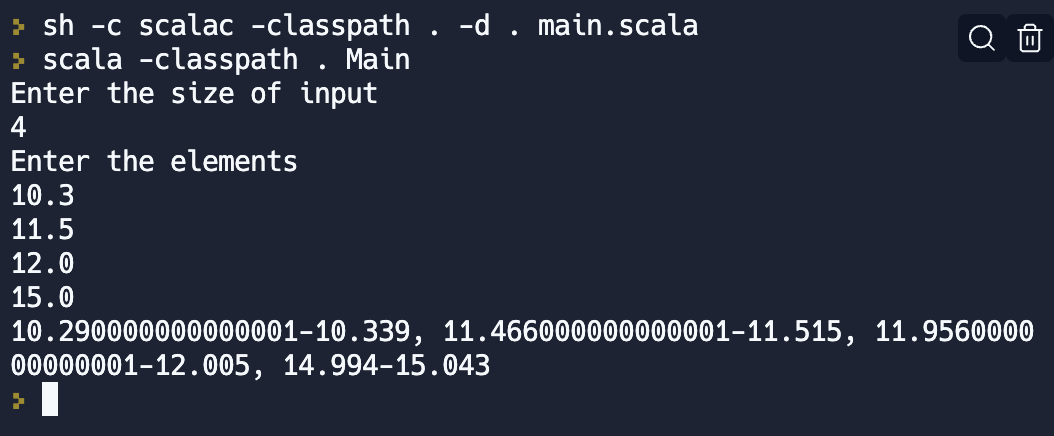
**TAS129**

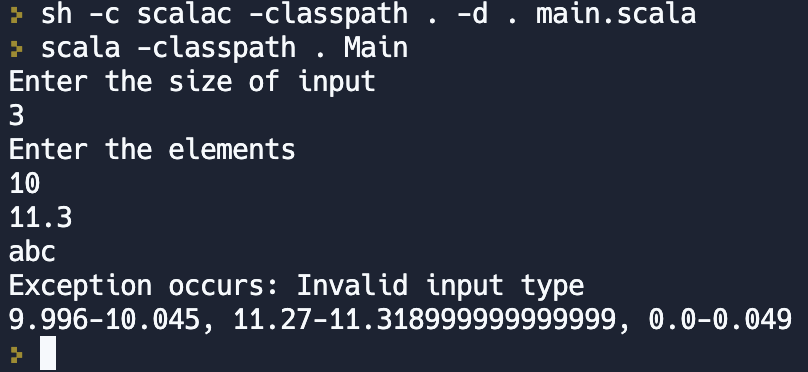
Scala Assignment

**Q1)**

| object Main {   def main(args: Array[String]): Unit = {    // taking user input    println("Enter the size of input")   var size = scala.io.StdIn.readInt()    val arr = new Array[Double](size)    println("Enter the elements")  try{  for (i <- 0 to size-1) {  arr(i) = scala.io.StdIn.readDouble()  }  }  catch{   // handling the exception to deal with input of other types  case e: Exception => println("Exception occurs: Invalid input type")  }    val buckets = (0 to 2000).map(\_ \* 0.049) // create a sequence of bucket boundaries   //mapping the array input according to required bucket    val output = arr.map { num =>  val bucketIndex = (num / 0.049).toInt  s"${buckets(bucketIndex)}-${buckets(bucketIndex) + 0.049}"  }    //printing the final output   println(output.mkString(", "))   } } |
| --- |

**Output:**





**Q2)**

| object PlayerStats {  case class Player(year: Int, name: String, country: String, matches: Int, runs: Int, wickets: Int)    def main(args: Array[String]): Unit = {  // Sample data  val players = Seq(  Player(2021, "Sam", "India", 23, 2300, 3),  Player(2021, "Ram", "India", 23, 300, 30),  Player(2021, "Mano", "India", 23, 300, 13)  )    // 1. Player with the highest run scored  val playerWithHighestRun = players.maxBy(\_.runs)  println(s"Player with the highest run scored: ${playerWithHighestRun.name} (${playerWithHighestRun.runs} runs)")  // 2. Top 5 players by run scored  val top5PlayersByRun = players.sortBy(\_.runs)(Ordering[Int].reverse).take(5)  println("Top 5 players by run scored:")  top5PlayersByRun.foreach(player => println(s"${player.name} (${player.runs} runs)"))    // 3. Top 5 players by wicket taken  val top5PlayersByWickets = players.sortBy(\_.wickets)(Ordering[Int].reverse).take(5)  println("Top 5 players by wickets taken:")  top5PlayersByWickets.foreach(player => println(s"${player.name} (${player.wickets} wickets)"))    // 4. Rank players with overall performance  val rankedPlayers = players.map { player =>  val overallPerformance = player.runs + (player.wickets \* 5)  (player, overallPerformance)  }.sortBy(\_.\_2)(Ordering[Int].reverse)    println("Ranking players with overall performance (runs + 5\*wickets):")  rankedPlayers.zipWithIndex.foreach { case ((player, score), index) =>  println(s"${index + 1}. ${player.name} (${player.runs} runs, ${player.wickets} wickets) - $score")  }  } } |
| --- |

**Output:**

