Assignment Submission- Session 19

Task 1
1.Write a program to read a text file and print the number of rows of data in the document.
2. Write a program to read a text file and print the number of words in the document.
3. We have a document where the word separator is -, instead of space. Write a spark code, to obtain the count of the total number of words present in the document.

Solution: Content of text file:

This is a wonderful Spark Scala session
This is a wonderful Spark Scala session
This is a wonderful Spark Scala session
This is a wonderful Spark Scala session

```
import org.apache.spark.{SparkConf, SparkContext}
object Task_1 {

  def main(args: Array[String]): Unit = {
    var conf = new SparkConf().setMaster("local[*]").setAppName("Example")
    val sc = new SparkContext(conf)

    val readFromFile = sc.textFile("E:\\ACADGILD\\SCALA_SPARK\\test.txt")
    val numOfRows = readFromFile.count()
    println("Num of Rows of data = "+numOfRows)

    var numOfWords = readFromFile.flatMap(line => line.split(" ")).count()

    //if file has words separated by "-"
   // var numOfWords = readFromFile.flatMap(line => line.split("-")).count()

    println("Num of Rows of words = "+numOfWords)

  }

}
```

Task 2
Problem Statement 1:
1. Read the text file, and create a tupled rdd.
2. Find the count of total number of rows present.
3. What is the distinct number of subjects present in the entire school
4. What is the count of the number of students in the school, whose name is Mathew and marks is 55

```
import org.apache.spark.{SparkConf, SparkContext}
object Task_2_Prob_1 {

  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setMaster("local").setAppName("Task_2_Prob_1")
    val sc = new SparkContext(conf)
    val readFile = sc.textFile("E:\\ACADGILD\\SCALA_SPARK\\19_Data.txt")

    // tupled RDD
    val tupleRDD = readFile.map(line => line.split(",")).map(array => (array(0),
array(1), array(2), array(3).toInt, array(4).toInt))
    println("Tupled RDD below")
    tupleRDD.foreach(println)

    //number of rows
    val numOfRows = readFile.count()
    println("Num of Rows of data = "+numOfRows)

    //distinct subject in the school
    val mapSub = readFile.map(row => row.split(",")(1))
    val distSub = mapSub.distinct()

    println("Distinct Subjects")
    distSub.foreach(println)

    //count of distinct subject
    println("Count of distinct subject in the school =" + distSub.count)

    // name mathew and marks 55
    val mathew55 = readFile.filter(row => row.contains("Mathew") &&
row.contains("55"))
    mathew55.foreach(println)

    println("Number of student with name Mathew and marks 55 ="+ mathew55.count)
  }
}
```

Problem Statement 2:

1. What is the count of students per grade in the school?
2. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)
3. What is the average score of students in each subject across all grades?
4. What is the average score of students in each subject per grade?
5. For all students in grade-2, how many have average score greater than 50?

```
import org.apache.spark.{SparkConf, SparkContext}

object Task_2_Prob_2 {
  def main(args: Array[String]): Unit = {
    val conf =  new SparkConf().setAppName("Task_2_Prob_2").setMaster("local")
    val sc = new SparkContext(conf)
    val readFile = sc.textFile("E:\\ACADGILD\\SCALA_SPARK\\19_Data.txt")
    //count of students per grade in the school
    val mapByGrade = readFile.map(row => (row.split(",")(2),(1)))

    val stuPerGrade = mapByGrade.reduceByKey((grade,value) => grade +
value).sortByKey()

    stuPerGrade.foreach(println)

    //find the average of each student (student with same name and different grade
are different)
    val mapNameAndGrade = readFile.map(row =>
((row.split(",")(0),row.split(",")(2)),(row.split(",")(3).toInt,1)))

    val totalMarksAndCount = mapNameAndGrade.reduceByKey((marks,subCount) =>
(marks._1 + subCount._1, marks._2 + subCount._2))

    totalMarksAndCount.foreach(println)

    val avgOfEachStu = totalMarksAndCount.mapValues{
      case (marksTot, numOfSub) => (marksTot.toFloat)/numOfSub
    }.sortByKey()

    avgOfEachStu.foreach(println)

    //average score of students in each subject across all grades

    val mapNameAndSub = readFile.map(row =>
((row.split(",")(0),row.split(",")(1)),(row.split(",")(3).toInt,1)))

    val marksSumPerSub = mapNameAndSub.reduceByKey((marks, subCount) => (marks._1 +
subCount._1, marks._2 + subCount._2) )

    val avgOfEachSub = marksSumPerSub.mapValues{
      case (marksTot, numOfSub) => (marksTot.toFloat)/numOfSub
    }.sortByKey()

    avgOfEachSub.foreach(println)

    //for all students in grade 2, how many have average >50
    val avgOver50_Grade_2 = avgOfEachStu.filter(entry => entry._1._2=="grade-2" &&
entry._2>50)

    avgOver50_Grade_2.foreach(println)
   println(avgOver50_Grade_2.count)

  }
}
```

Problem Statement 3:
Are there any students in the college that satisfy the below criteria:

1. Average score per student_name across all grades is same as average score per student_name per grade
   Hint - Use Intersection Property