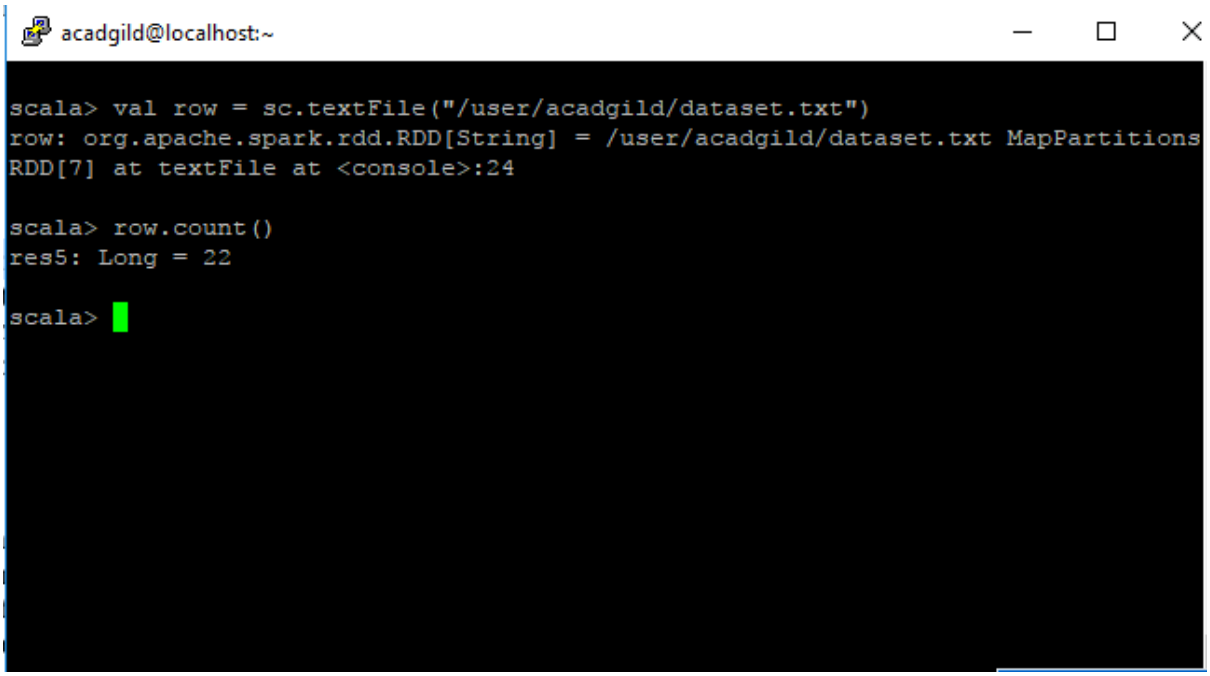
Assignment 19

Task 1

1.Write a program to read a text file and print the number of rows of data in the document.

val row = sc.textFile(“/user/acadgild/dataset.txt”)

row.count()

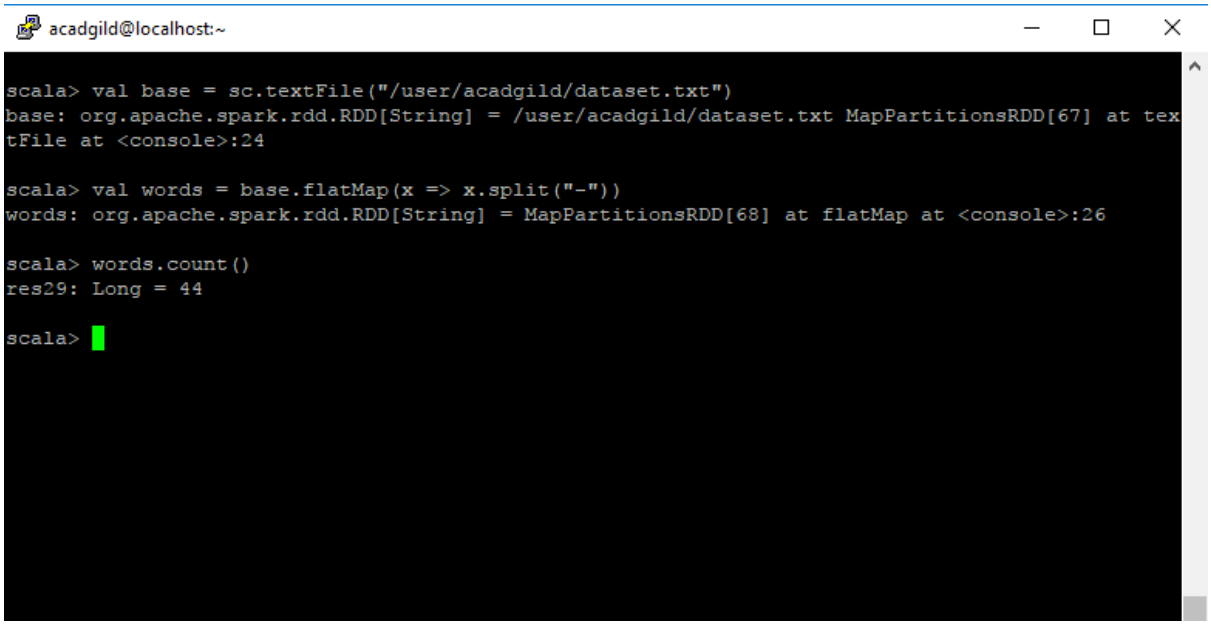


2. Write a program to read a text file and print the number of words in the document.

val base = sc.textFile(“/user/acadgild/dataset.txt”)

val words = base.flatMap(x => x.split(“,”))

words.count()

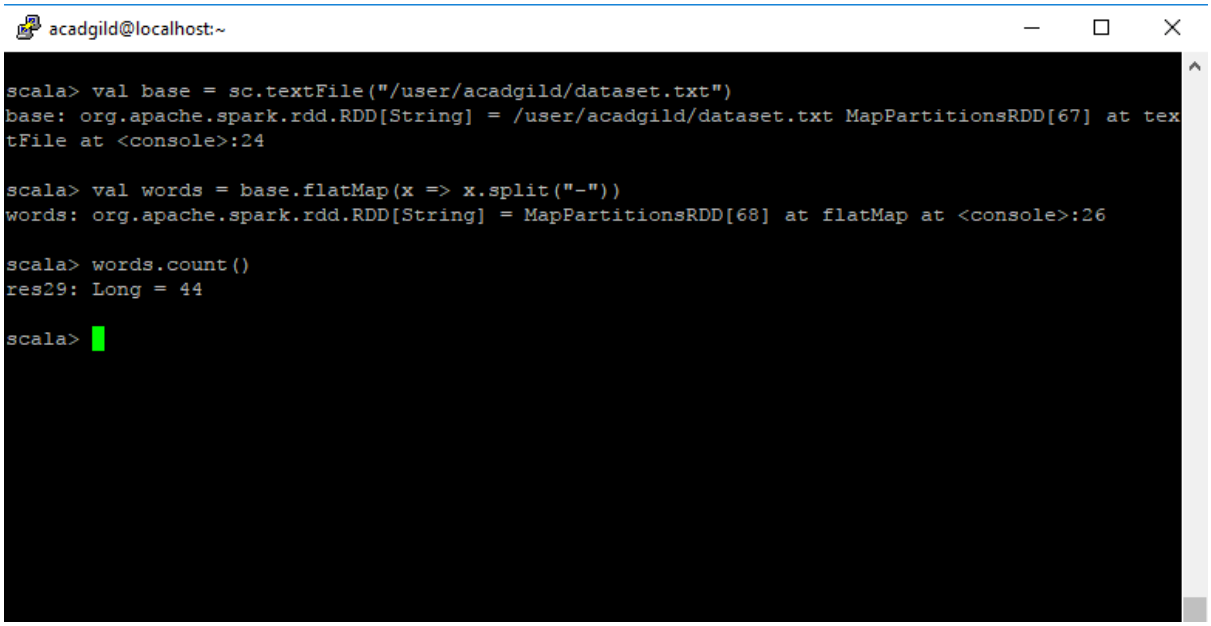


3. We have a document where the word separator is -, instead of space. Write a spark code, to obtain the count of the total number of words present in the document.

val base = sc.textFile("/user/acadgild/dataset.txt")

val words = base.flatMap(x => x.split("-"))

words.count()



Task 2

Problem Statement 1:

1. **Read the text file, and create a tupled rdd.**

val baseRDD = sc.textFile(“/user/acadgild/dataset.txt”).map(=>(x.split(“,”)(0), x.split(“,”)(1), x.split(“,”)(2), x.split(“,”)(3).toInt, x.split(“,”)(4).toInt))

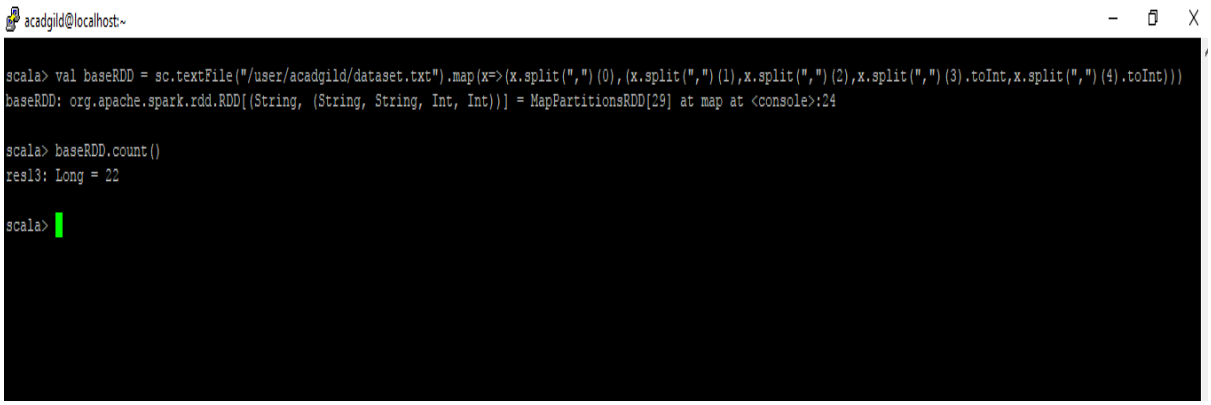
baseRDD.foreach(println)

```
scala> val baseRDD = sc.textFile("/user/acadgild/dataset.txt").map(x=>(x.split("
,")(0),(x.split(",")(1),x.split(",")(2),x.split(",")(3).toInt,x.split(",")(4).to
Int)))
baseRDD: org.apache.spark.rdd.RDD[(String, (String, String, Int, Int))] = MapPar
titionsRDD[20] at map at <console>:24

scala> baseRDD.foreach(println)
(Mathew,(science,grade-3,45,12))
(Mathew,(history,grade-2,55,13))
(Mark,(maths,grade-2,23,13))
(Mark,(science,grade-1,76,13))
(John,(history,grade-1,14,12))
(John,(maths,grade-2,74,13))
(Lisa,(science,grade-1,24,12))
(Lisa,(history,grade-3,86,13))
(Andrew,(maths,grade-1,34,13))
(Andrew,(science,grade-3,26,14))
(Andrew,(history,grade-1,74,12))
(Mathew,(science,grade-2,55,12))
(Mathew,(history,grade-2,87,12))
(Mark,(maths,grade-1,92,13))
(Mark,(science,grade-2,12,12))
(John,(history,grade-1,67,13))
(John,(maths,grade-1,35,11))
(Lisa,(science,grade-2,24,13))
(Lisa,(history,grade-2,98,15))
(Andrew,(maths,grade-1,23,16))
(Andrew,(science,grade-3,44,14))
(Andrew,(history,grade-2,77,11))

scala>
```

2.Find the count of total number of rows present.

val baseRDD =sc.textFile(“/user/acadgild/dataset.txt”).map(=>(x.split(“,”)(0), x.split(“,”)(1), x.split(“,”)(2), x.split(“,”)(3).toInt, x.split(“,”)(4).toInt)))

baseRDD.count()

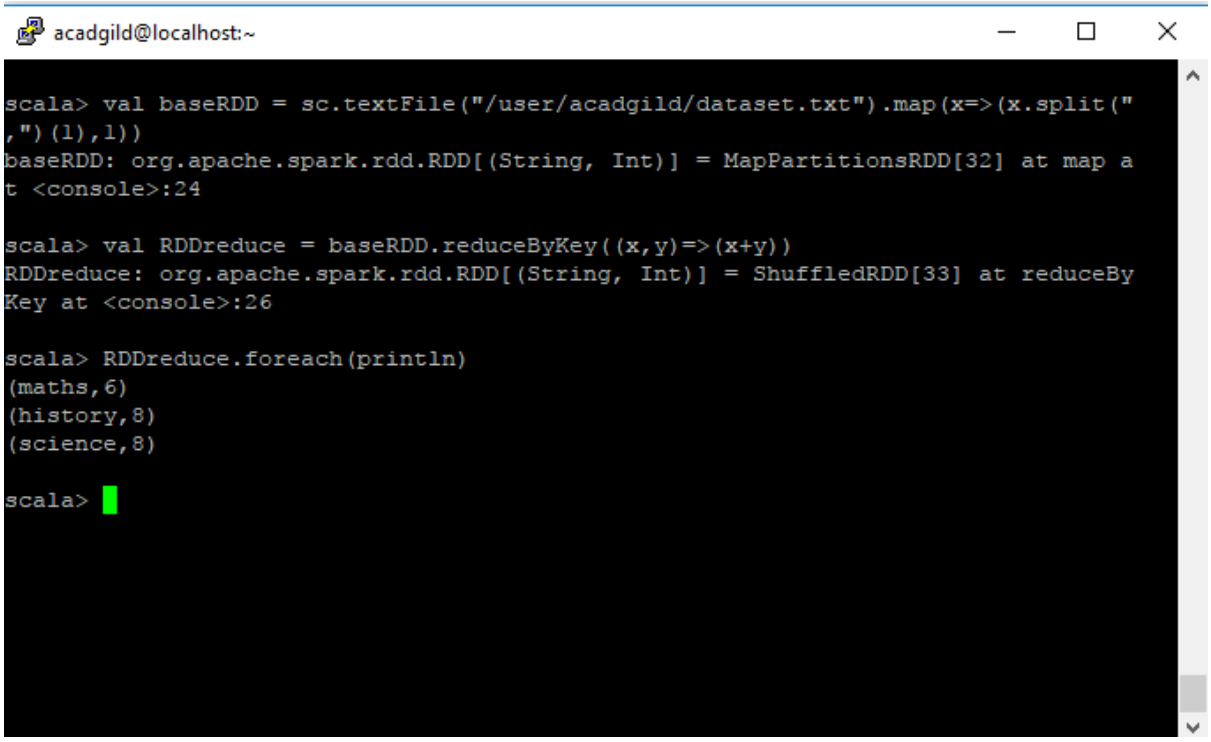


3.What is the distinct number of subjects present in the entire school

val baseRDD = sc.textFile("/user/acadgild/dataset.txt").map(x=>(x.split(",")(1),1))

val RDDreduce = baseRDD.reduceByKey((x,y)=>(x+y))

RDDreduce.foreach(println)

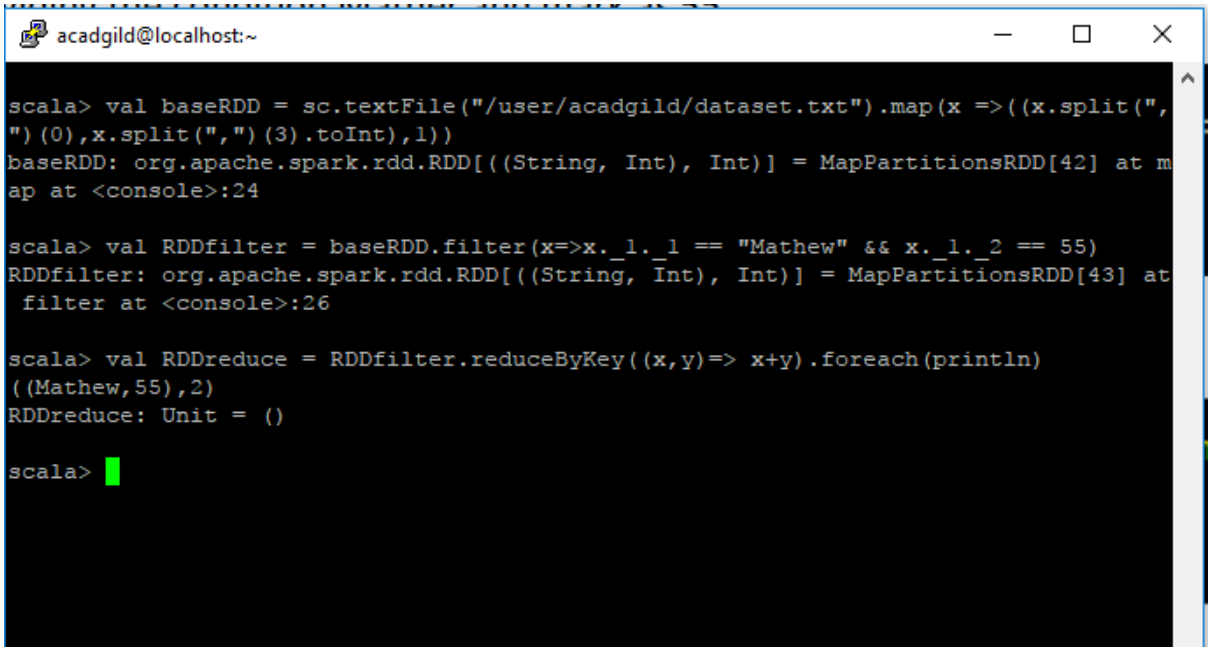


4. What is the count of the number of students in the school, whose name is Mathew and marks is 55

val baseRDD = sc.textFile(“/user/acadgild/dataset.txt”).map(x=>((x.split(“,”)(0),x.split(“,”)(3).toInt),1))

val RDDfilter = baseRDD.filter(x=>x._1._1 == “Mathew” && x._1._2 == 55)

val RDDreduce == RDDfilter.reduceByKey((x,y)=>x+y).foreach(println)

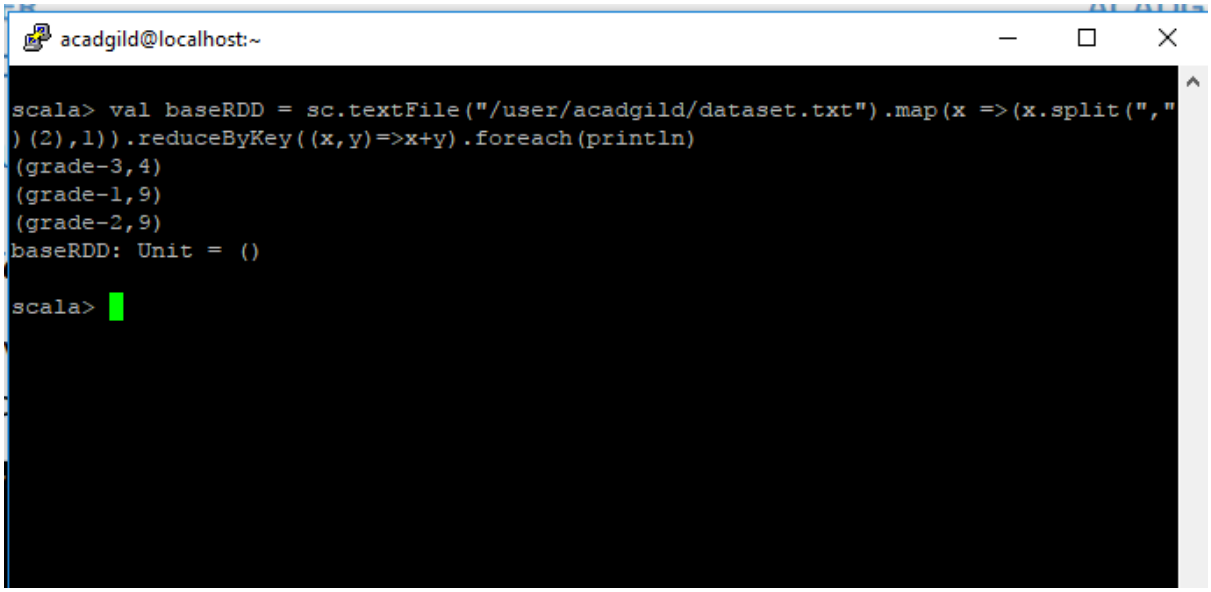


Problem Statement 2:

1. **What is the count of students per grade in the school?**

   val baseRDD = sc.textFile("/user/acadgild/dataset.txt").map(x =>

   (x.split(",")(2),1)).reduceByKey((x,y)=>x+y).foreach(println)



2. **Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)**

val baseRDD = sc.textFile("/user/acadgild/dataset.txt").map(x=>((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))

val RDDmap = baseRDD.mapValues(x=>(x,1))

val RDDreduce = RDDmap.reduceByKey((x,y) => (x._1 + y._1, x._2 + y._2))

val StudAvg = RDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

acadgild@localhost:~

```
scala> val baseRDD = sc.textFile("/user/acadgild/dataset.txt").map(x =>((x.split(",
")(0),x.split(",")(2)),x.split(",")(3).toInt))
baseRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[29] a
t map at <console>:24

scala> val RDDmap = baseRDD.mapValues(x=>(x,1))
RDDmap: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = MapPartitionsRDD
[30] at mapValues at <console>:26

scala> val RDDreduce = RDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
RDDreduce: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = ShuffledRDD[3
1] at reduceByKey at <console>:28

scala> val StudAvg = RDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}
StudAvg: org.apache.spark.rdd.RDD[((String, String), Double)] = MapPartitionsRDD[32
] at mapValues at <console>:30

scala> StudAvg.foreach(println)
((Lisa,grade-1),24.0)
((Mark,grade-2),17.5)
((Lisa,grade-2),61.0)
((Mathew,grade-3),45.0)
((Andrew,grade-2),77.0)
((Andrew,grade-1),43.666666666666664)
((Lisa,grade-3),86.0)
((John,grade-1),38.666666666666664)
((John,grade-2),74.0)
((Mark,grade-1),84.0)
((Andrew,grade-3),35.0)
((Mathew,grade-2),65.66666666666667)

scala>
```

3. **What is the average score of students in each subject across all grades?**

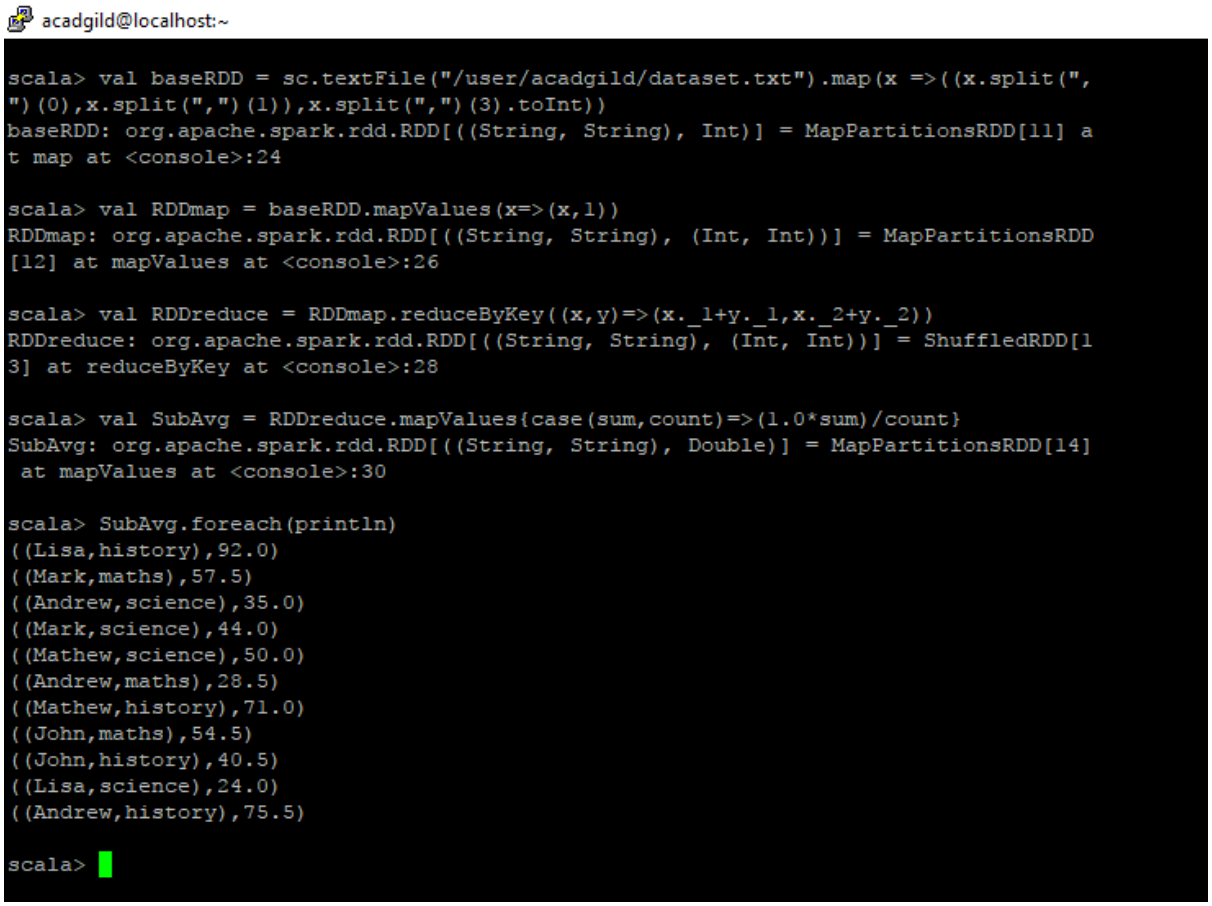
val baseRDD =

sc.textFile("/user/acadgild/dataset.txt").map(x=>((x.split(",")(0),x.split(",")(1))),x.split(",")(3).toInt))

val RDDmap = baseRDD.mapValues(x=>(x,1))

val RDDreduce = RDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))

val SubAvg = RDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}



4. What is the average score of students in each subject per grade?

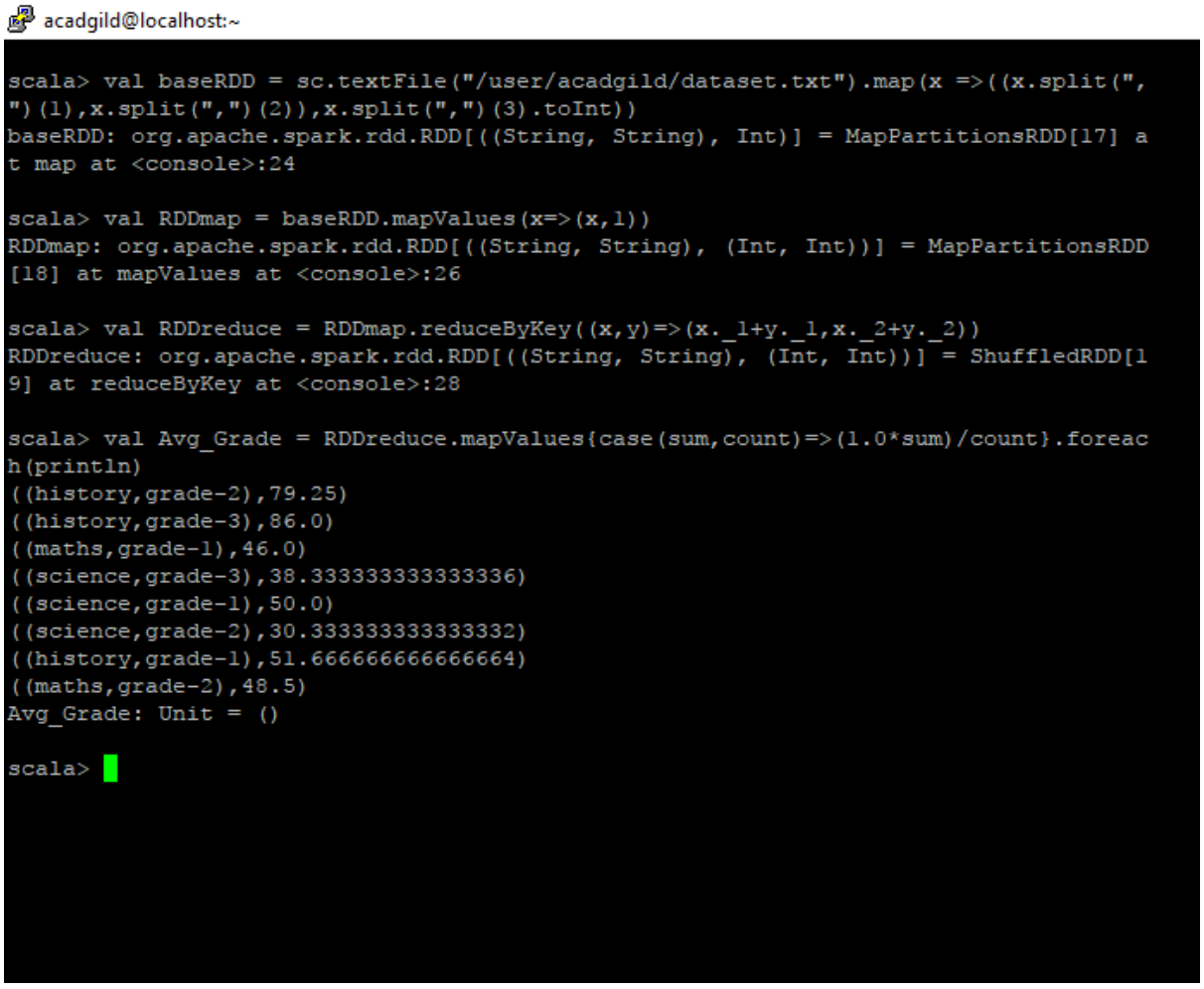
val baseRDD =

sc.textFile("/user/acadgild/dataset.txt").map(x=>((x.split(",")(1),x.split(",")(2)),x.split(",")(3).toInt))

val RDDmapvalue = baseRDD.mapValues(x=>(x,1))

val RDDreduce = RDDmapvalue.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))

val Avg_Grade = RDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}.foreach(println)



5. **For all students in grade-2, how many have average score greater than 50?**

val baseRDD =

sc.textFile("/user/acadgild/dataset.txt").map(x=>((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))

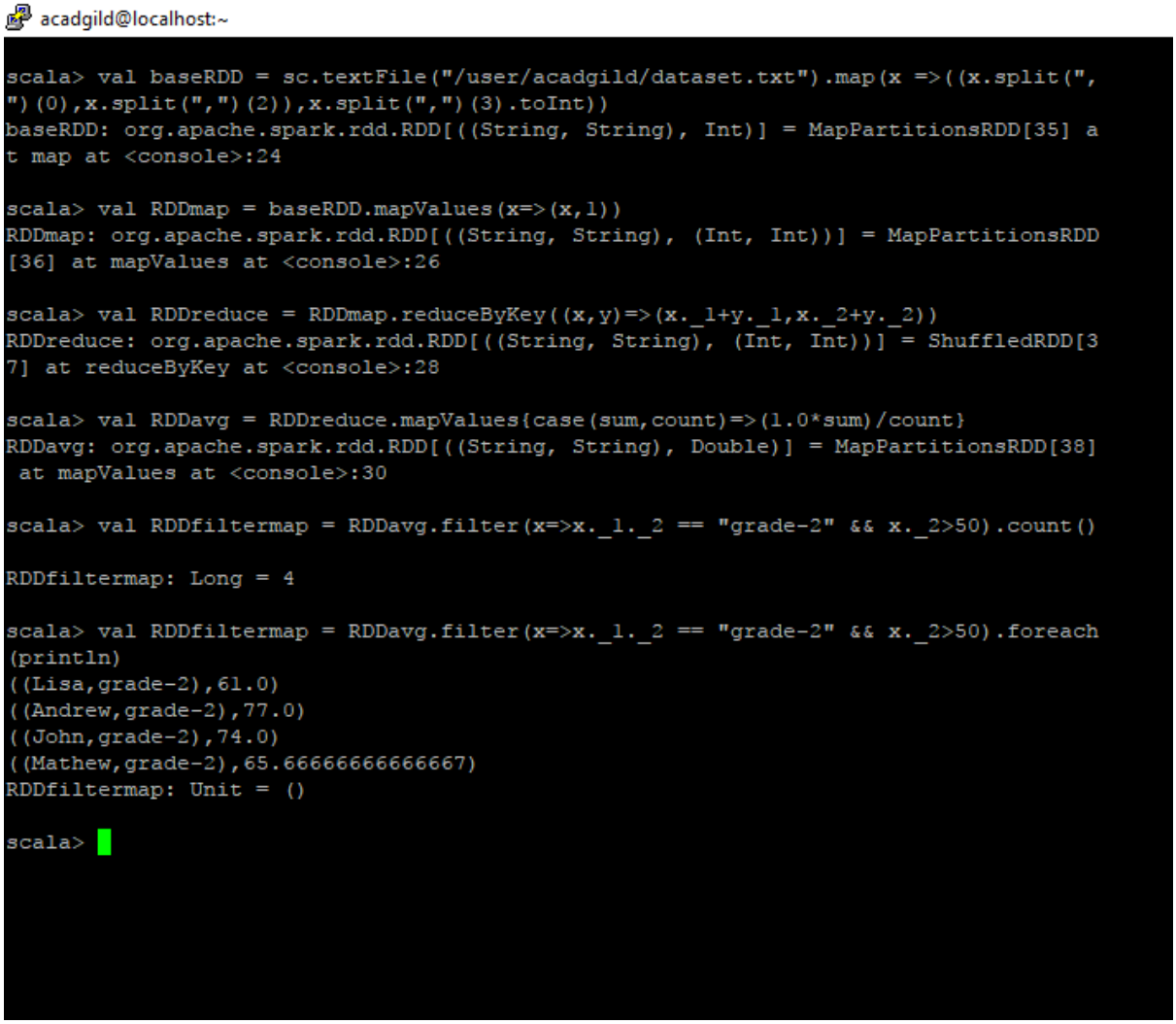
val RDDmap = baseRDD.mapValues(x=>(x,1))

val RDDreduce = RDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))

val RDDavg = RDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

val RDDfiltermap = RDDavg.filter(x=>x._1._2 == "grade-2" && x._2>50).count()

val RDDfiltermap = RDDavg.filter(x=>x._1._2 == "grade-2" && x._2>50).foreach(println)



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

We created a paired RDD named as baseRDD1 by extracting only name and marks

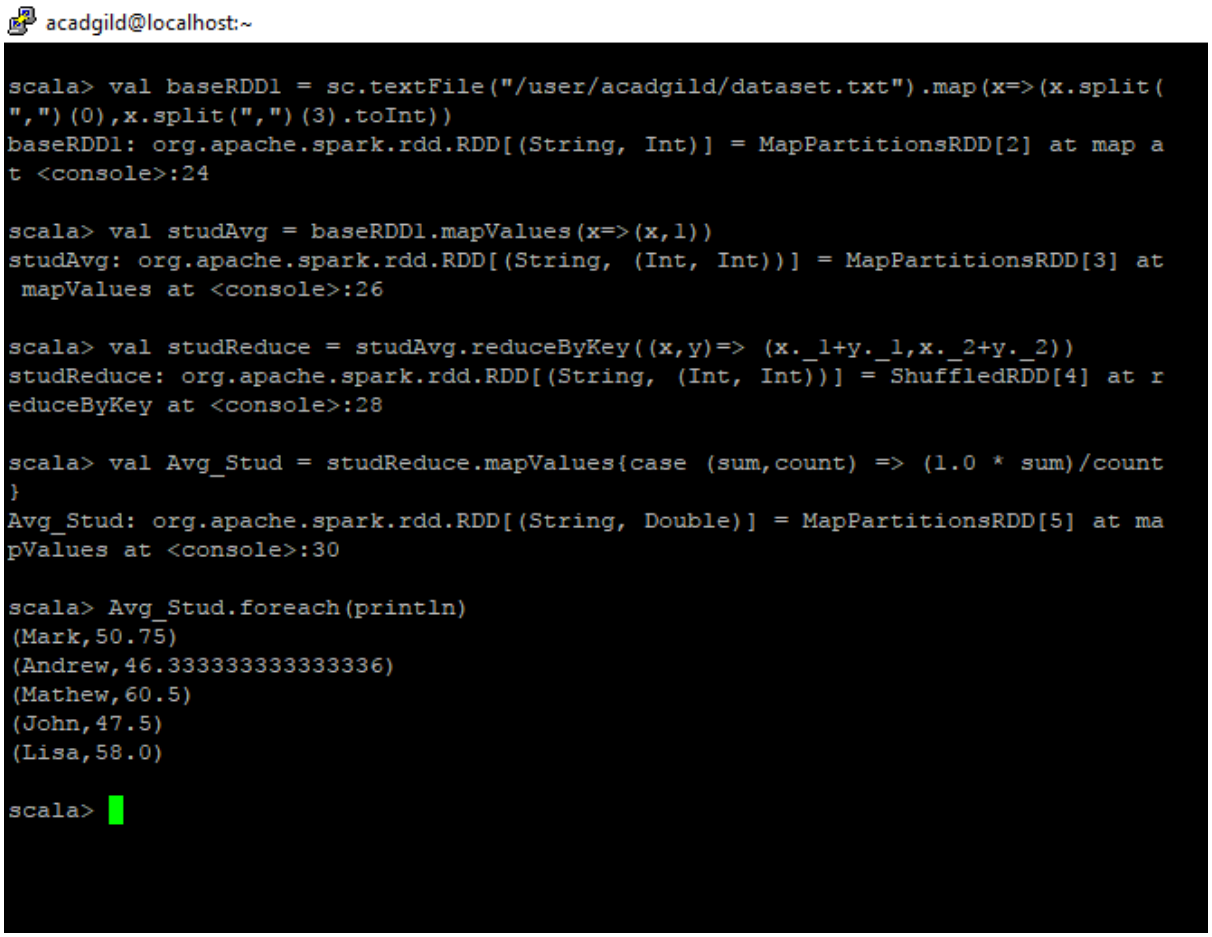
val baseRDD1 =

sc.textFile("/user/acadgild/dataset.txt").map(x=>(x.split(",")(0),x.split(",")(3).toInt))

val studAvg = baseRDD1.mapValues(x=>(x,1)).foreach(println)

val studReduce = studAvg.reduceByKey((x,y)=> (x._1+y._1,x._2+y._2))

val Avg_Stud = studReduce.mapValues{case (sum,count) => (1.0 * sum)/count}



we are creating another paired RDD named as baseRDD2 by extracting name and grade as key and marks as value from the input file,

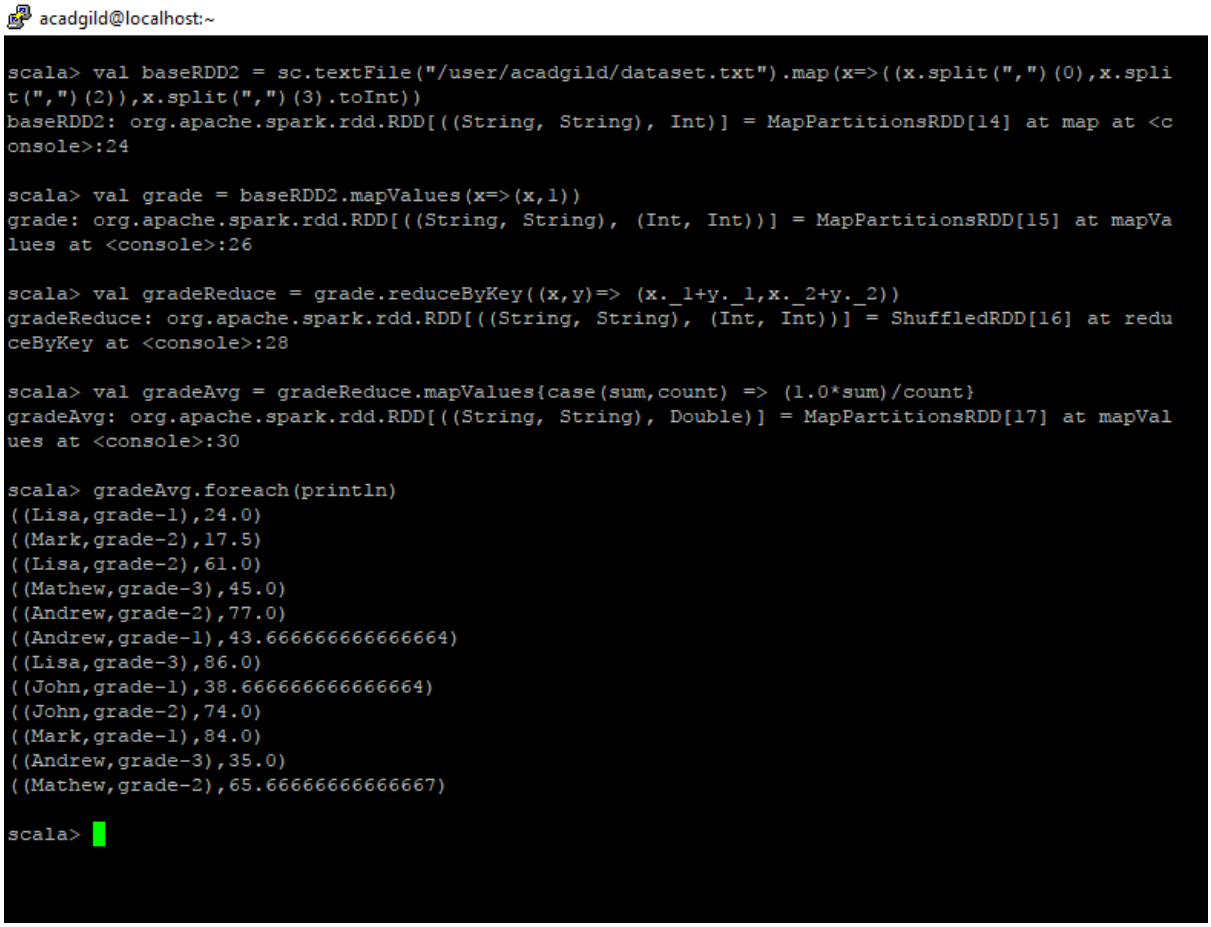
val baseRDD2 =

sc.textFile("/user/acadgild/dataset.txt").map(x=>((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt)).foreach(println)

val grade = baseRDD2.mapValues(x=>(x,1))

val gradeReduce = grade.reduceByKey((x,y)=> (x._1+y._1,x._2+y._2))

val gradeAvg = gradeReduce.mapValues{case(sum,count) => (1.0*sum)/count}



In below step we are using intersection function between flatgradeAvg and flatnameAvg rdd's to find whether any common student is there.

val flatgradeAvg = gradeAvg.map(x=> x._1._1 + "," + x._2.toDouble)

flatgradeAvg.foreach(println)

val flatAvg_Stud = Avg_Stud.map(x=>x._1+","+x._2)

flatAvg_Stud.foreach(println)

acadgild@localhost:~

```
scala> val flatgradeAvg = gradeAvg.map(x=> x._1._1 + "," + x._2.toDouble)
flatgradeAvg: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[21] at map at <console>:32

scala> flatgradeAvg.foreach(println)
Lisa,24.0
Mark,17.5
Lisa,61.0
Mathew,45.0
Andrew,77.0
Andrew,43.666666666666664
Lisa,86.0
John,38.666666666666664
John,74.0
Mark,84.0
Andrew,35.0
Mathew,65.66666666666667

scala> val flatAvg_Stud = Avg_Stud.map(x=>x._1+","+x._2)
flatAvg_Stud: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[22] at map at <console>:32

scala> flatAvg_Stud.foreach(println)
Mark,50.75
Andrew,46.333333333333336
Mathew,60.5
John,47.5
Lisa,58.0

scala>
```

val commanval = flatgradeAvg.intersection(flatAvg_Stud)

commanval.foreach(println)

acadgild@localhost:~

```
scala> val commanval = flatgradeAvg.intersection(flatAvg_Stud)
commanval: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[28] at intersection at <console>
:44

scala> commanval.foreach(println)

scala>
```