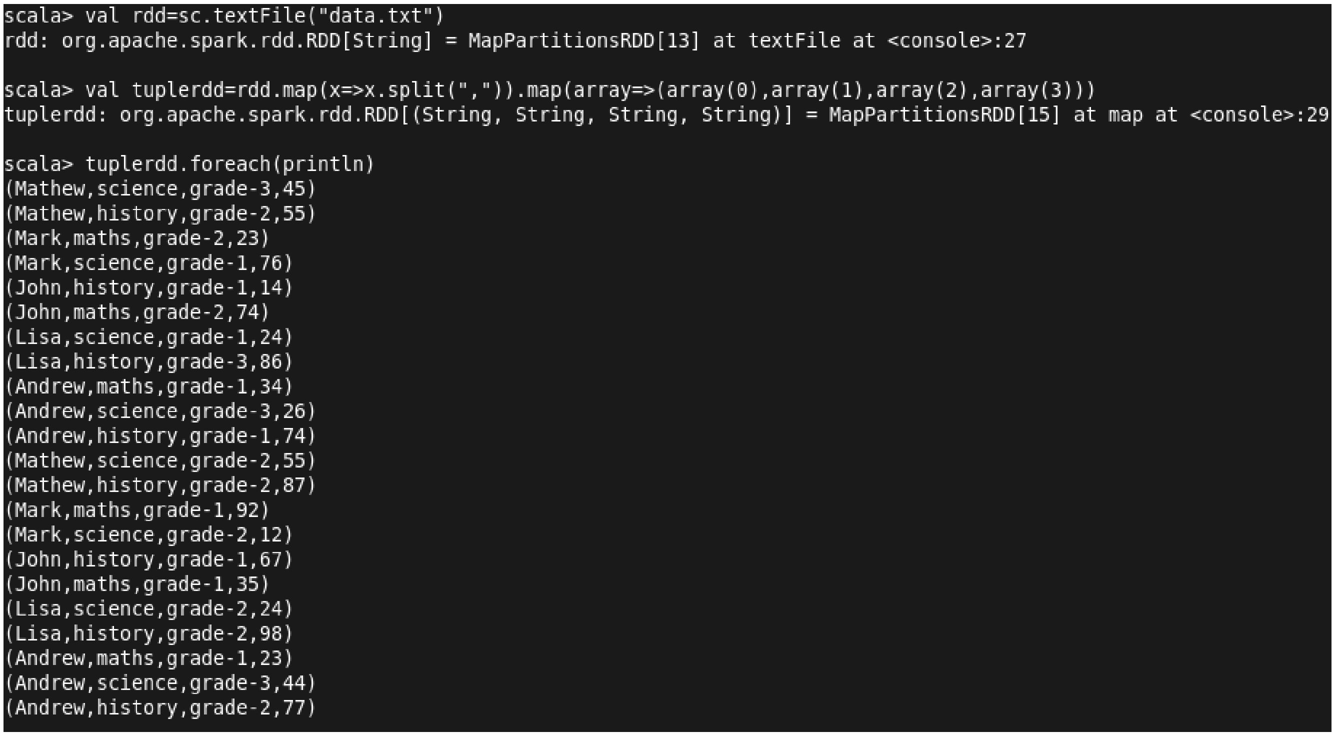
ASSIGNMENT17.2

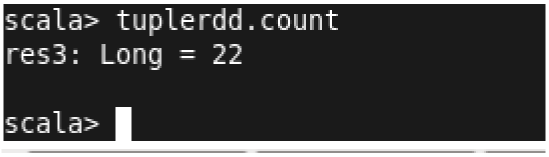
PROBLEM1

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



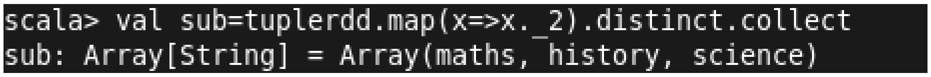
```
scala> val rdd=sc.textFile("data.txt")
rdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[13] at textFile at <console>:27

scala> val tuplerdd=rdd.map(x=>x.split(",")).map(array=>(array(0),array(1),array(2),array(3)))
tuplerdd: org.apache.spark.rdd.RDD[(String, String, String, String)] = MapPartitionsRDD[15] at map at <console>:29

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

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



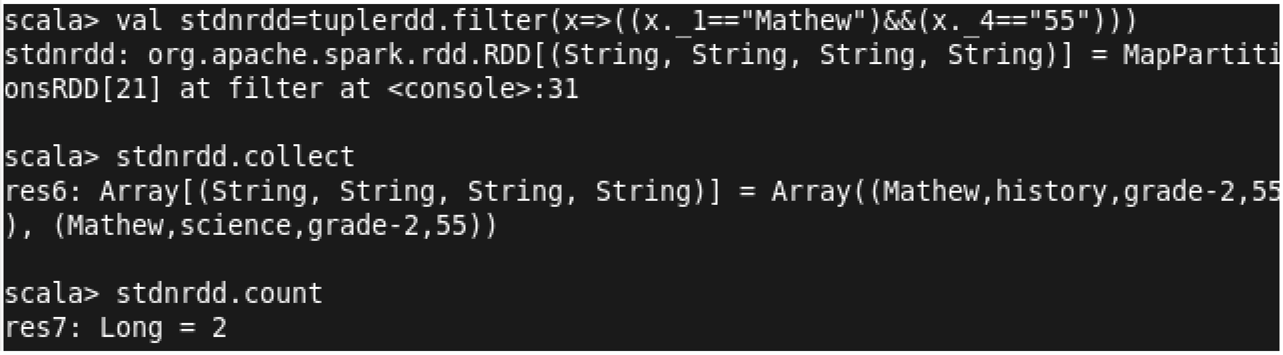
```
scala> tuplerdd.count
res3: Long = 22

scala> 
```

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



```
scala> val sub=tuplerdd.map(x=>x._2).distinct.collect
sub: Array[String] = Array(maths, history, science)
```

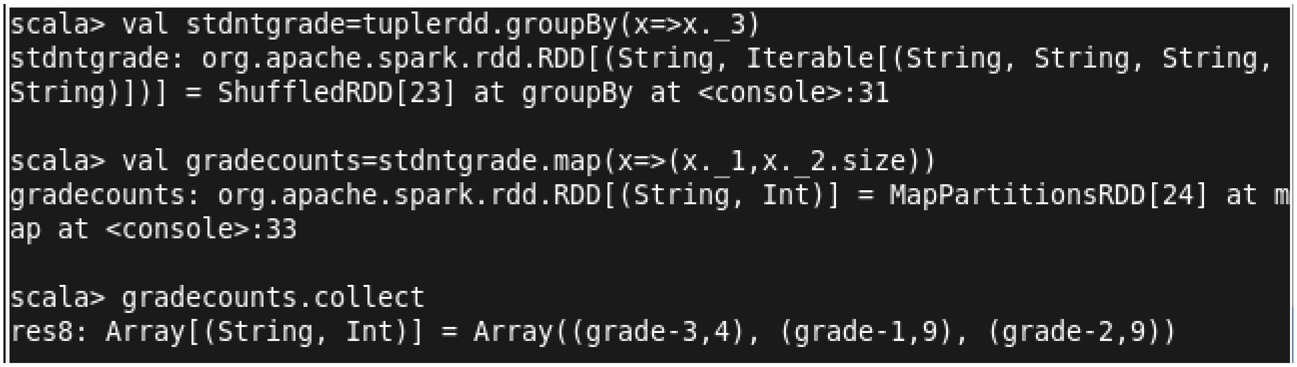
4. What is the count of the number of students in the school, whose name is Mathew and Marks is 55



```
scala> val stdnrdd=tuplerdd.filter(x=>((x._1=="Mathew")&&(x._4=="55")))
stdnrdd: org.apache.spark.rdd.RDD[(String, String, String, String)] = MapPartiti
onsRDD[21] at filter at <console>:31

scala> stdnrdd.collect
res6: Array[(String, String, String, String)] = Array((Mathew,history,grade-2,55
), (Mathew,science,grade-2,55))

scala> stdnrdd.count
res7: Long = 2
```

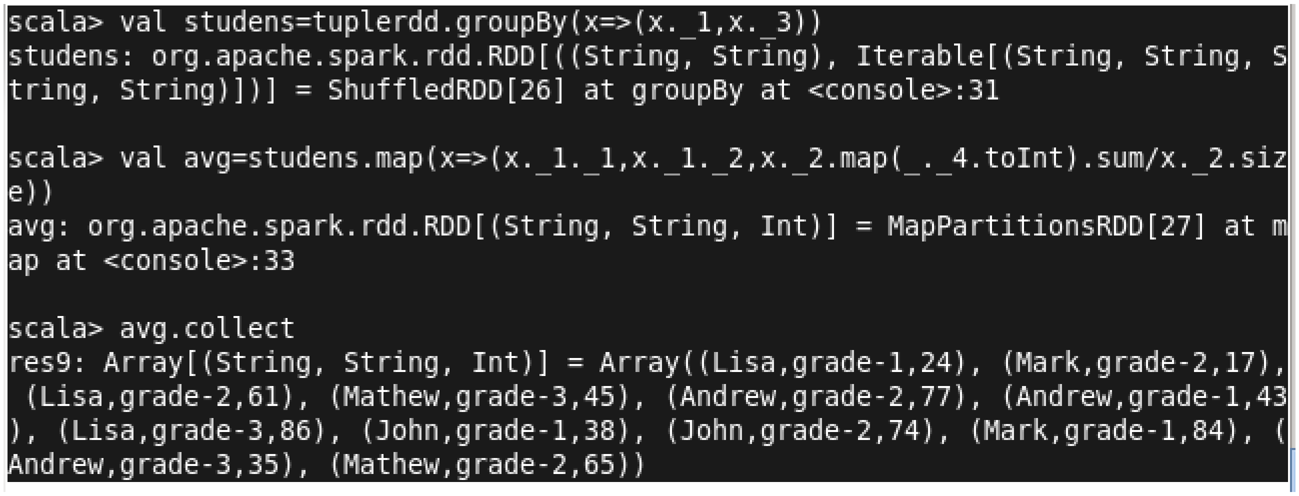
PROBLEM2

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



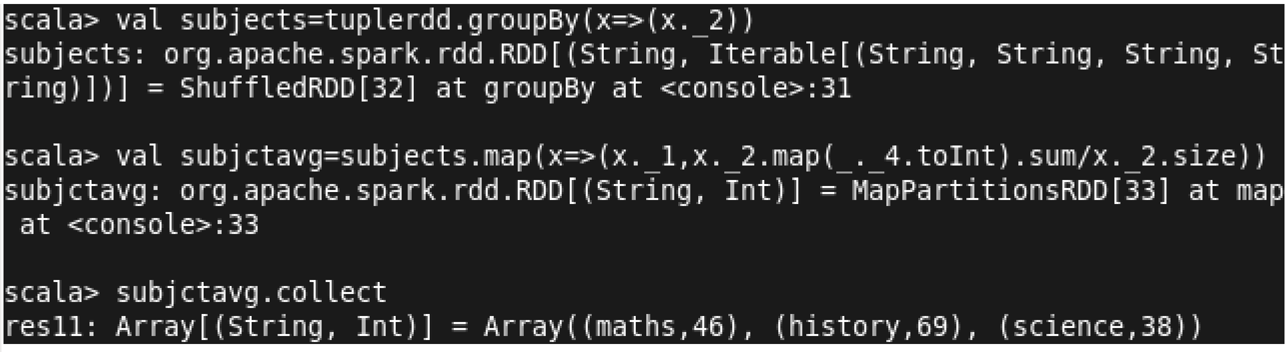
```
scala> val stdntgrade=tuplerdd.groupBy(x=>x._3)
stdntgrade: org.apache.spark.rdd.RDD[(String, Iterable[(String, String, String,
String)])] = ShuffledRDD[23] at groupBy at <console>:31

scala> val gradecounts=stdntgrade.map(x=>(x._1,x._2.size))
gradecounts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[24] at m
ap at <console>:33

scala> gradecounts.collect
res8: Array[(String, Int)] = Array((grade-3,4), (grade-1,9), (grade-2,9))
```

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



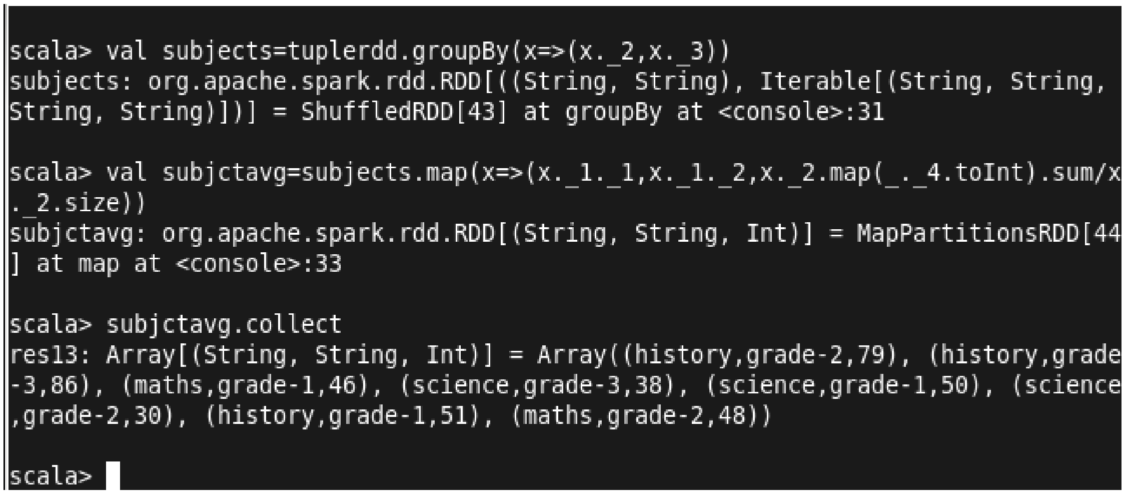
```
scala> val studens=tuplerdd.groupBy(x=>(x._1,x._3))
studens: org.apache.spark.rdd.RDD[((String, String), Iterable[(String, String, S
tring, String)])] = ShuffledRDD[26] at groupBy at <console>:31

scala> val avg=studens.map(x=>(x._1._1,x._1._2,x._2.map(_._4.toInt).sum/x._2.siz
e))
avg: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[27] at m
ap at <console>:33

scala> avg.collect
res9: Array[(String, String, Int)] = Array((Lisa,grade-1,24), (Mark,grade-2,17),
 (Lisa,grade-2,61), (Mathew,grade-3,45), (Andrew,grade-2,77), (Andrew,grade-1,43
), (Lisa,grade-3,86), (John,grade-1,38), (John,grade-2,74), (Mark,grade-1,84), (
Andrew,grade-3,35), (Mathew,grade-2,65))
```

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



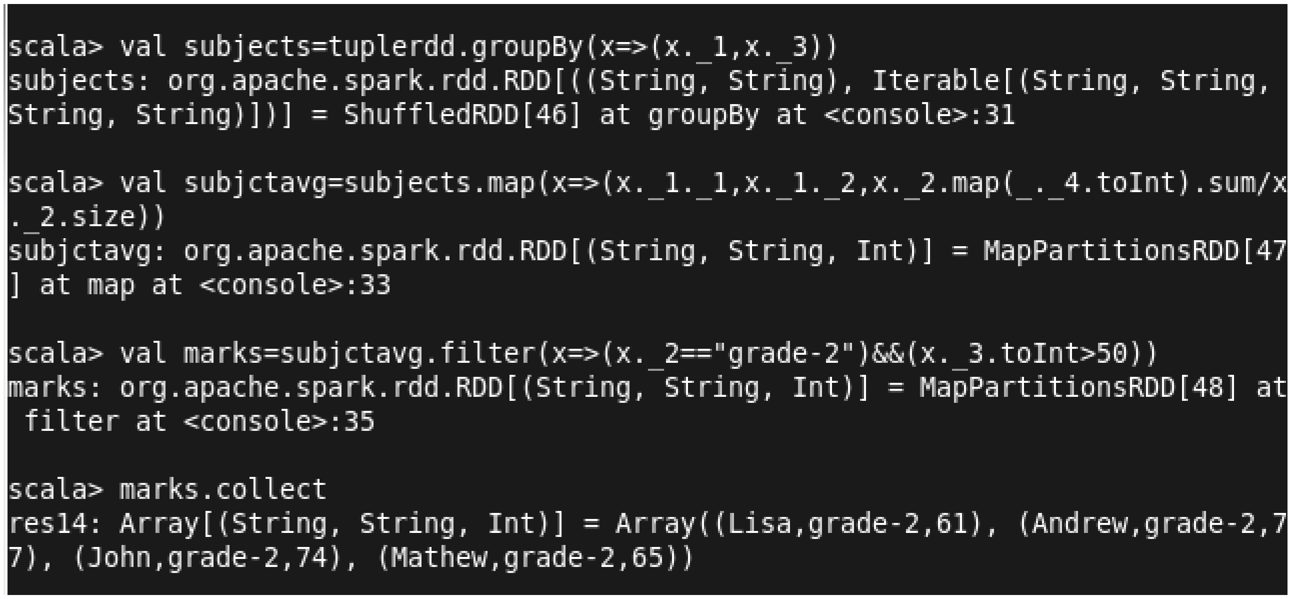
```
scala> val subjects=tuplerdd.groupBy(x=>(x._2))
subjects: org.apache.spark.rdd.RDD[(String, Iterable[(String, String, String, St
ring)])] = ShuffledRDD[32] at groupBy at <console>:31

scala> val subjctavg=subjects.map(x=>(x._1,x._2.map(_._4.toInt).sum/x._2.size))
subjctavg: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[33] at map
 at <console>:33

scala> subjctavg.collect
res11: Array[(String, Int)] = Array((maths,46), (history,69), (science,38))
```

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



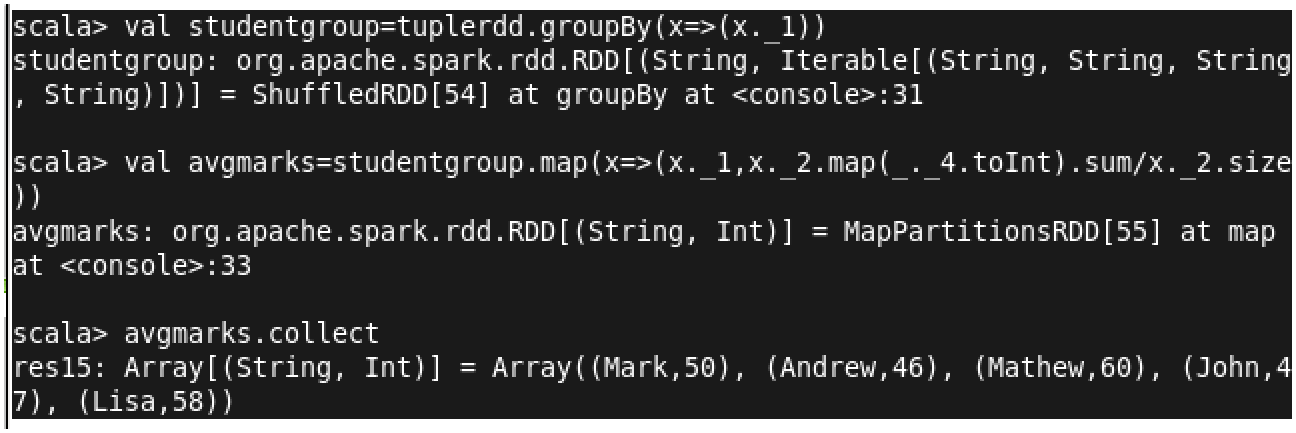
```
scala> val subjects=tuplerdd.groupBy(x=>(x._2,x._3))
subjects: org.apache.spark.rdd.RDD[((String, String), Iterable[(String, String,
String, String)])] = ShuffledRDD[43] at groupBy at <console>:31

scala> val subjctavg=subjects.map(x=>(x._1._1,x._1._2,x._2.map(_._4.toInt).sum/x
._2.size))
subjctavg: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[44
] at map at <console>:33

scala> subjctavg.collect
res13: Array[(String, String, Int)] = Array((history,grade-2,79), (history,grade
-3,86), (maths,grade-1,46), (science,grade-3,38), (science,grade-1,50), (science
,grade-2,30), (history,grade-1,51), (maths,grade-2,48))

scala> █
```

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

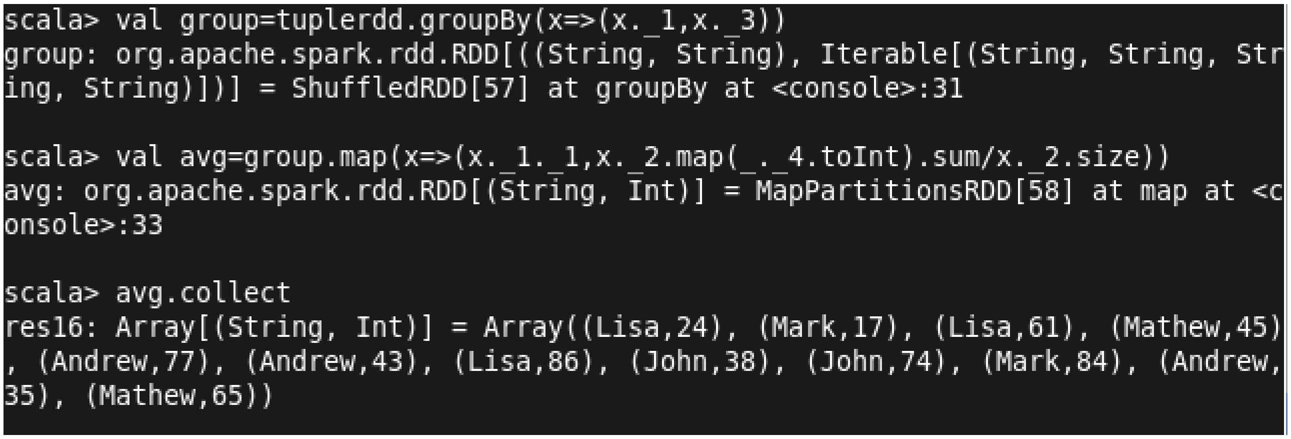


```
scala> val subjects=tuplerdd.groupBy(x=>(x._1,x._3))
subjects: org.apache.spark.rdd.RDD[((String, String), Iterable[(String, String,
String, String)])] = ShuffledRDD[46] at groupBy at <console>:31

scala> val subjctavg=subjects.map(x=>(x._1._1,x._1._2,x._2.map(_._4.toInt).sum/x
._2.size))
subjctavg: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[47
] at map at <console>:33

scala> val marks=subjctavg.filter(x=>(x._2=="grade-2")&&(x._3.toInt>50))
marks: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[48] at
 filter at <console>:35

scala> marks.collect
res14: Array[(String, String, Int)] = Array((Lisa,grade-2,61), (Andrew,grade-2,7
7), (John,grade-2,74), (Mathew,grade-2,65))
```

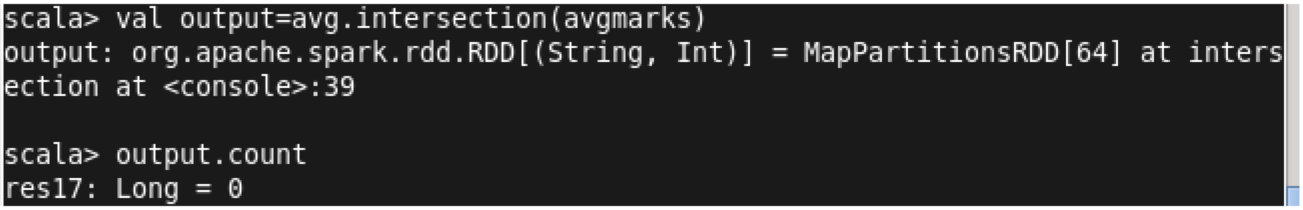
Problem Statement 3:
Are there any students in the college that satisfy the below criteria :
1. Average score per student_name across all grades is same as average score per
Student_name per grade
Hint - Use Intersection Property.



```
scala> val studentgroup=tuplerdd.groupBy(x=>(x._1))
studentgroup: org.apache.spark.rdd.RDD[(String, Iterable[(String, String, String
, String)])] = ShuffledRDD[54] at groupBy at <console>:31

scala> val avgmarks=studentgroup.map(x=>(x._1,x._2.map(_._4.toInt).sum/x._2.size
))
avgmarks: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[55] at map
at <console>:33

scala> avgmarks.collect
res15: Array[(String, Int)] = Array((Mark,50), (Andrew,46), (Mathew,60), (John,4
7), (Lisa,58))
```



```
scala> val group=tuplerdd.groupBy(x=>(x._1,x._3))
group: org.apache.spark.rdd.RDD[((String, String), Iterable[(String, String, Str
ing, String)])] = ShuffledRDD[57] at groupBy at <console>:31

scala> val avg=group.map(x=>(x._1._1,x._2.map(_._4.toInt).sum/x._2.size))
avg: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[58] at map at <c
onsole>:33

scala> avg.collect
res16: Array[(String, Int)] = Array((Lisa,24), (Mark,17), (Lisa,61), (Mathew,45)
, (Andrew,77), (Andrew,43), (Lisa,86), (John,38), (John,74), (Mark,84), (Andrew,
35), (Mathew,65))
```



```
scala> val output=avg.intersection(avgmarks)
output: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[64] at inters
ection at <console>:39

scala> output.count
res17: Long = 0
```