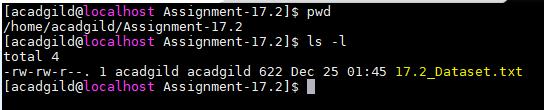
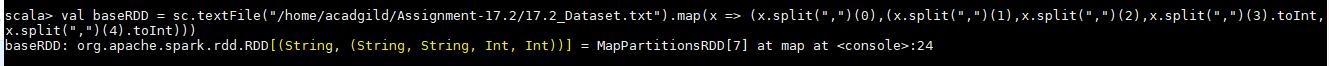
Dataset:

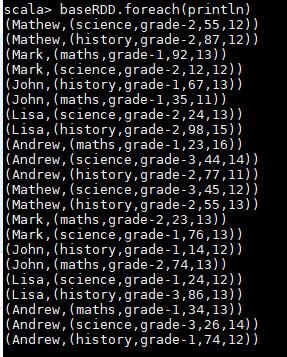


**Problem Statement 1:**

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

* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => (x.split(",")(0),(x.split(",")(1),x.split(",")(2),x.split(",")(3).toInt,x.split(",")(4).toInt)))
* val count1 = baseRDD.count()





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



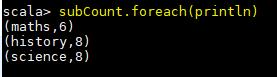
**3. What is the distinct number of subjects present in the entire school**

* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => (x.split(",")(1),1))
* val subCount = baseRDD.reduceByKey((x,y) => x + y)
* subCount.foreach(println)





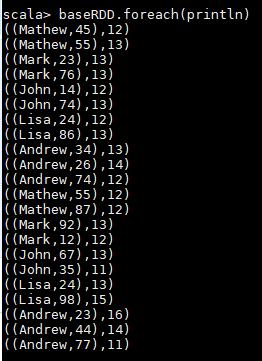


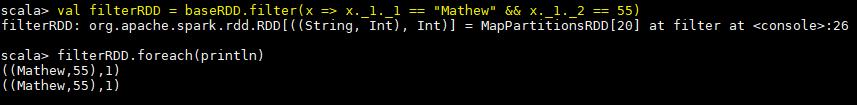


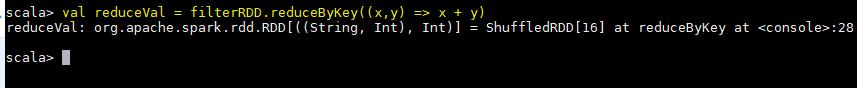
1. **What is the count of the number of students in the school, whose name is Mathew and marks is 55**

* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(3).toInt),1))
* val filterRDD = baseRDD.filter(x => x.\_1.\_1 == "Mathew" && x.\_1.\_2 == "55")
* val reduceVal = filterRDD.reduceByKey((x,y) => x + y)









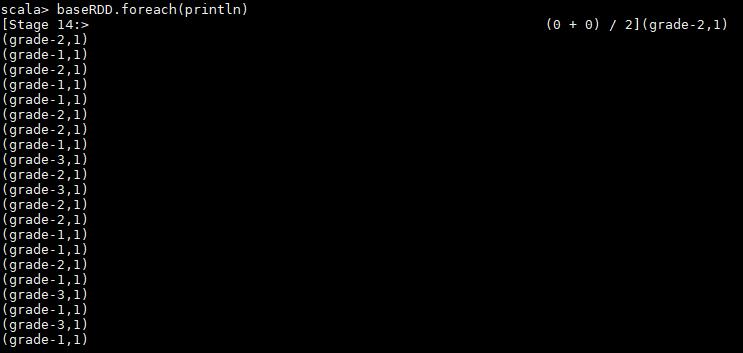


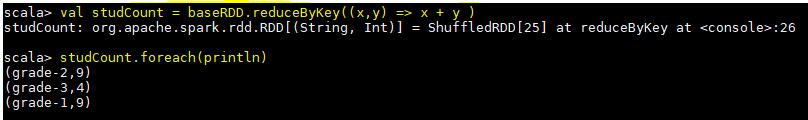
**Problem Statement 2:**

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

* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => (x.split(",")(2),1))
* val studCount = baseRDD.reduceByKey((x,y) => x + y )



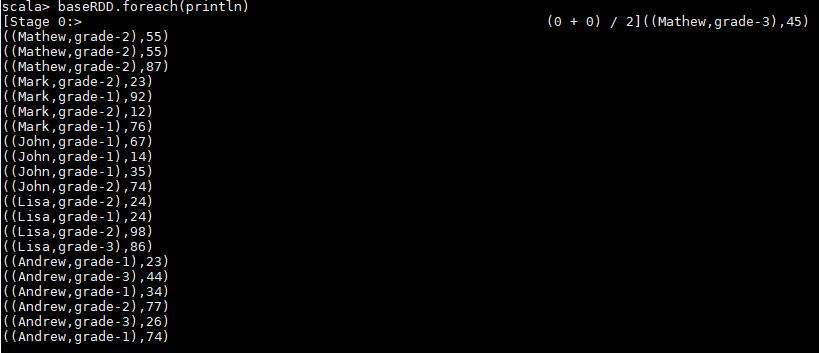


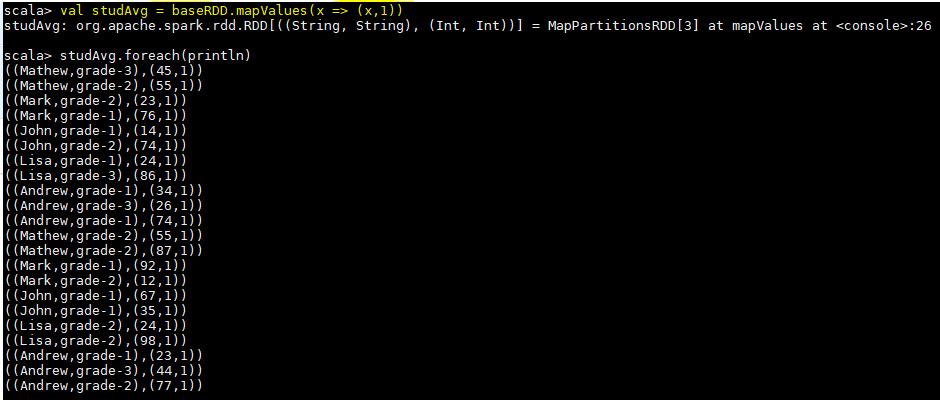


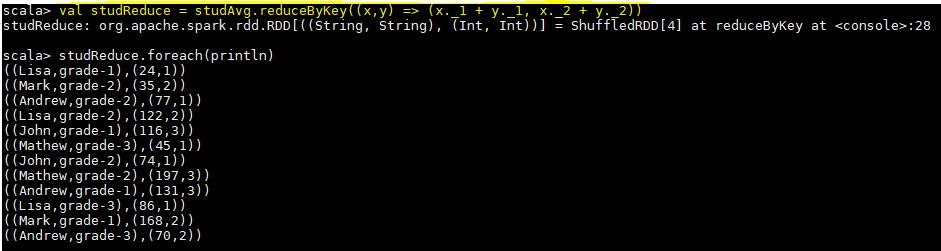
1. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

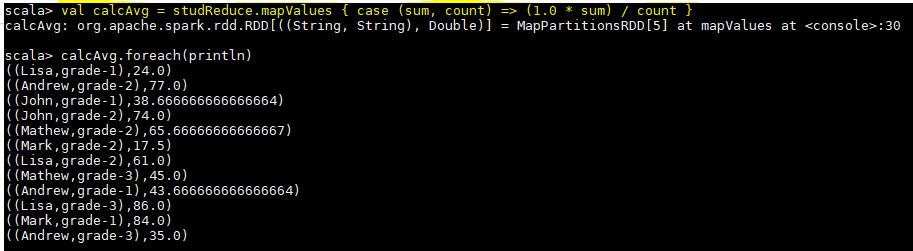
* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))
* val studAvg = baseRDD.mapValues(x => (x,1))
* val studReduce = studAvg.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val calcAvg = studReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }











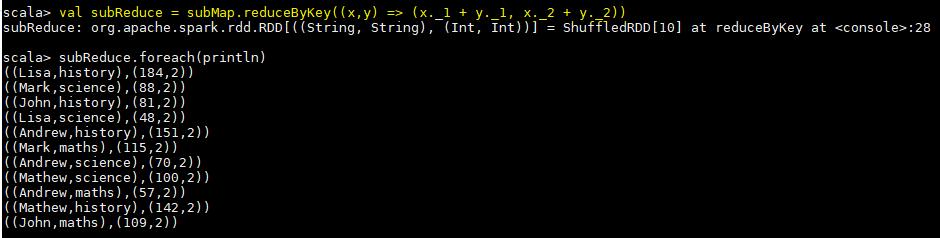
3. What is the average score of students in each subject across all grades?

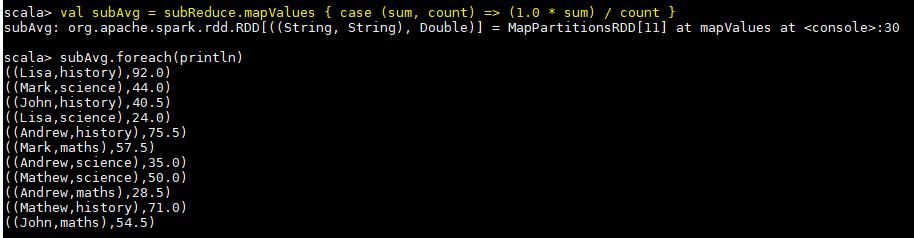
Below is the code used to find the result-

* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(1)),x.split(",")(3).toInt))
* val subMap = baseRDD.mapValues(x => (x,1))
* val subReduce = subMap.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val subAvg = subReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }



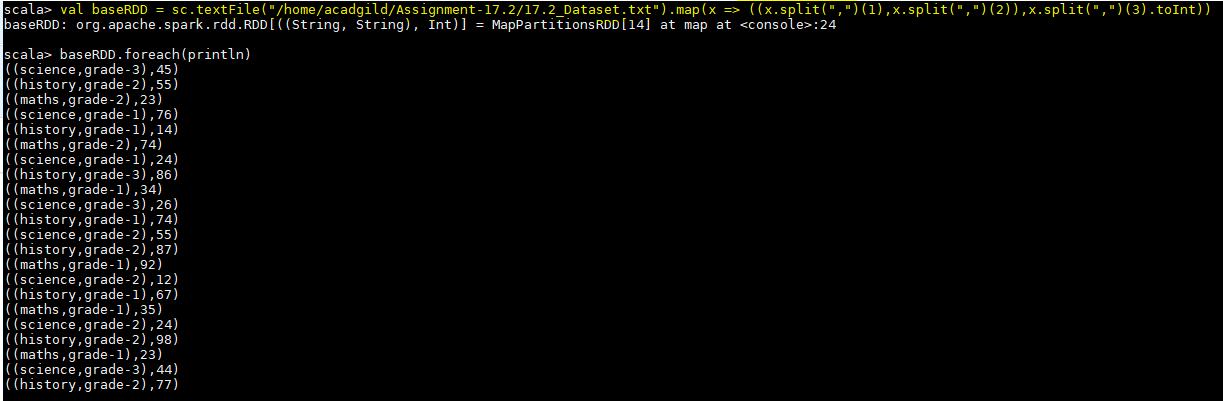




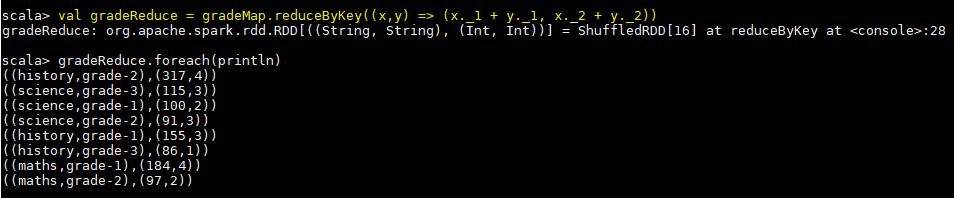


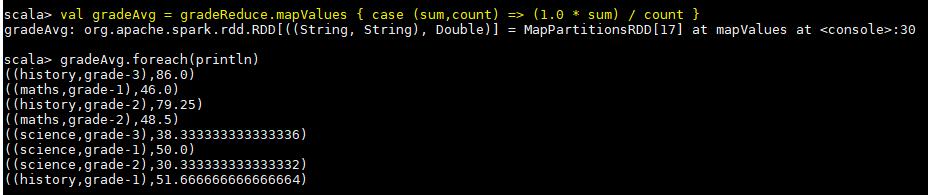
4. What is the average score of students in each subject per grade?

* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(1),x.split(",")(2)),x.split(",")(3).toInt))
* val gradeMap = baseRDD.mapValues(x => (x,1))
* val gradeReduce = gradeMap.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val gradeAvg = gradeReduce.mapValues { case (sum,count) => (1.0 \* sum) / count }





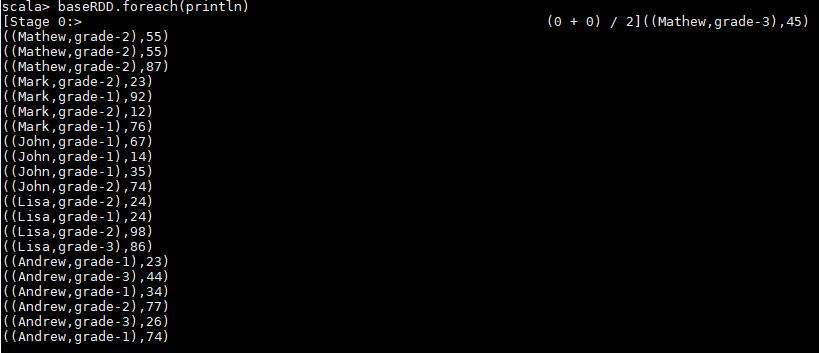


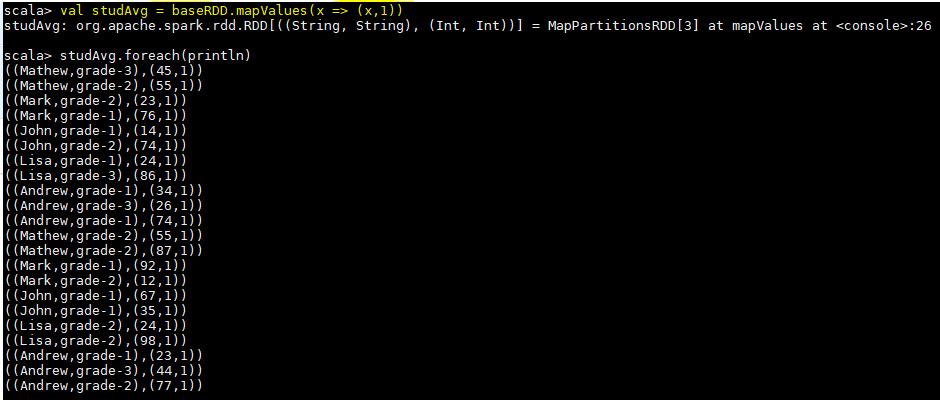


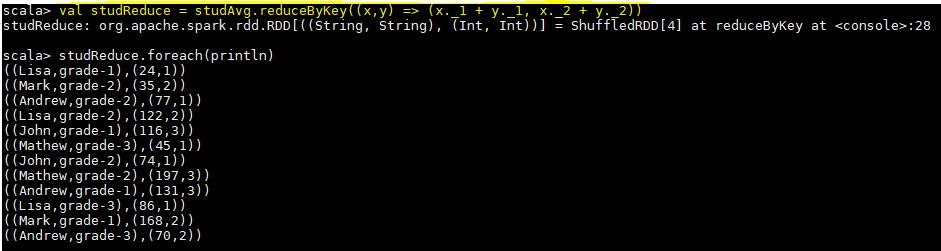
5. For all students in grade-2, how many have average score greater than 50?

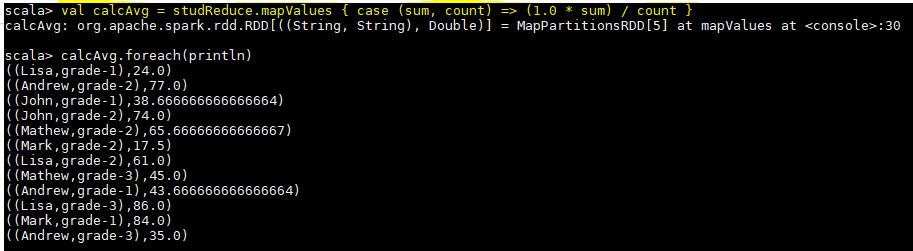
* val baseRDD = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))
* val studAvg = baseRDD.mapValues(x => (x,1))
* val studReduce = studAvg.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val calcAvg = studReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }
* val filterGrade = calcAvg.filter(x => x.\_1.\_2 == "grade-2" && x.\_2 > 50)
* val countStud = filterGrade2.count()

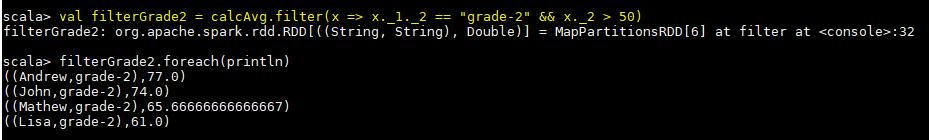


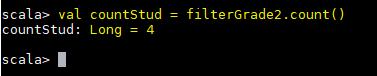










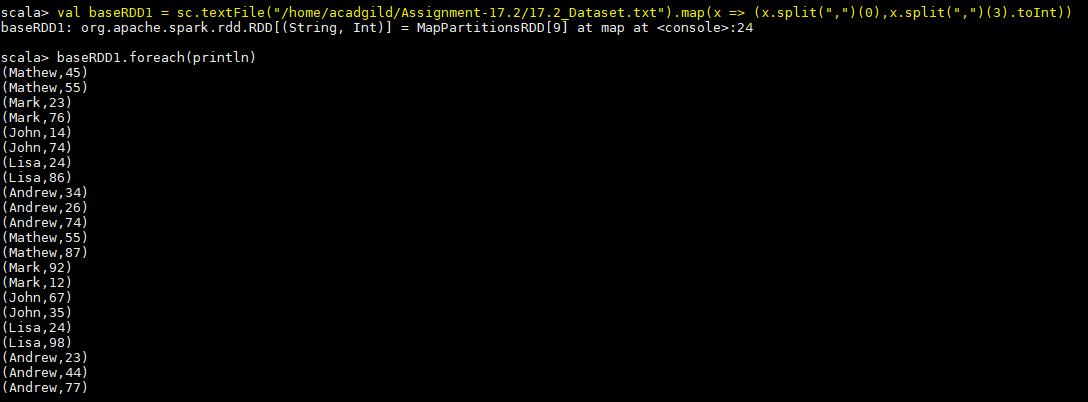


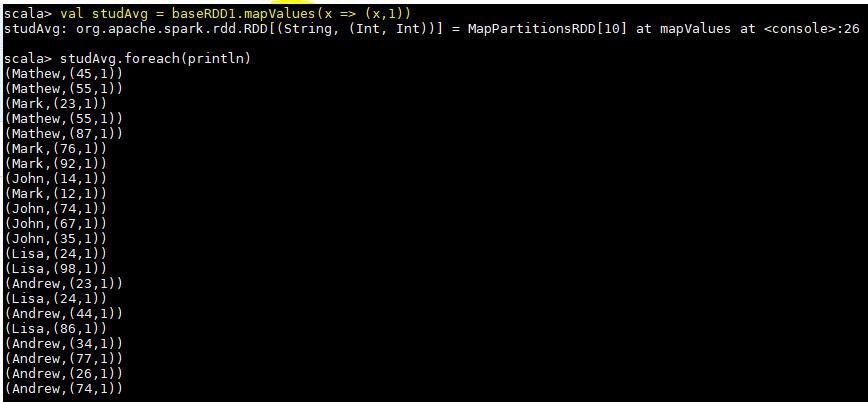
**Problem Statement 3:**

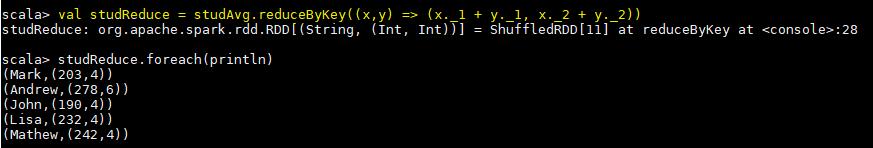
Are there any students in the college that satisfy the below criteria :

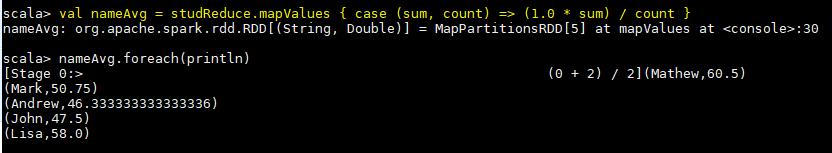
1. Average score per student\_name across all grades is same as average score per student\_name per grade

* val baseRDD1 = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => (x.split(",")(0),x.split(",")(3).toInt))
* val studAvg = baseRDD1.mapValues(x => (x,1))
* val studReduce = studAvg.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val nameAvg = studReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }









* val baseRDD2 = sc.textFile("/home/acadgild/Assignment-17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))
* val gradeMap = baseRDD2.mapValues(x => (x,1))
* val gradeReduce = gradeMap.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val gradeAvg = gradeReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }

