**Task 1**

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

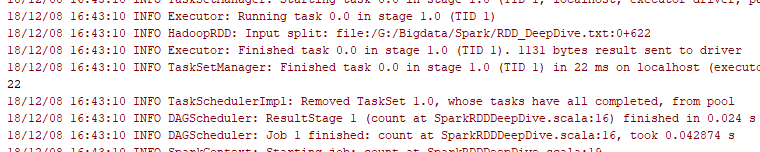
**2. Write a program to read a text file and print the number of words in the document.**

**3. We have a document where the word separator is -, instead of space. Write a spark**

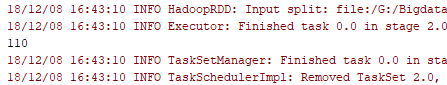
**code, to obtain the count of the total number of words present in the document.**

***Code logic***

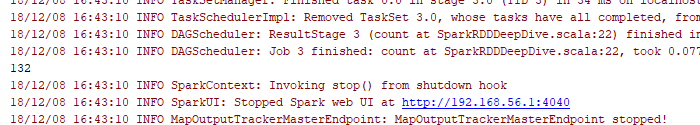
**package** core  
  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** SparkRDDDeepDive {  
 **def** main(args: Array[String]): Unit = {  
  
 *println*(**"hey Scala"**)  
  
 **val** conf = **new** SparkConf().setMaster(**"local"**).setAppName(**"SimpleApp"**)  
 **val** sc = **new** SparkContext(conf)  
 **val** x1 = sc.textFile(**"G:\\Bigdata\\Spark\\RDD\_DeepDive.txt"**);  
  
 *//1.Write a program to read a text file and print the number of rows of data in the document.  
  
 println*(x1.count())



*//2. Write a program to read a text file and print the number of words in the document.* **val** x2 = x1.flatMap( y => y.split(**","**))  
 *println*(x2.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** x3 = x2.flatMap( y => y.split(**"-"**))  
 *println*(x3.count())  
  
 }  
}



**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**

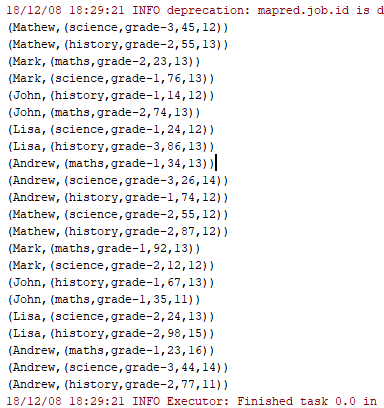
**4. What is the count of the number of students in the school, whose name is Mathew and**

**marks is 55**

*Code logic*

**package** core  
  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** SparkRDDDeepDiveTask2 {  
  
 **def** main(args: Array[String]): Unit = {  
  
 *println*(**"hey Scala"**)  
  
 **val** conf = **new** SparkConf().setMaster(**"local"**).setAppName(**"SimpleApp"**)  
 **val** sc = **new** SparkContext(conf)  
 **val** x1 = sc.textFile(**"G:\\Bigdata\\Spark\\RDD\_DeepDive.txt"**);  
  
 *//1. Read the text file, and create a tupled rdd.* **val** x2 = x1.map(x => (x.split(**","**)(0),(x.split(**","**)(1),x.split(**","**)(2),x.split(**","**)(3).toInt,x.split(**","**)(4).toInt)))  
 x2.foreach(*println*)  
  
 *//2. Find the count of total number of rows present.  
 println*(x1.count())  
  
 *//3. What is the distinct number of subjects present in the entire school* **val** splitvalue = x1.map(x => (x.split(**","**)(0),x.split(**","**)(1),x.split(**","**)(2),x.split(**","**)(3).toInt,x.split(**","**)(4).toInt))  
 **val** x3 = splitvalue.map(x => (x.\_2,1)).  
 reduceByKey((a,b) => (a+b))  
 x3.foreach(*println*)  
  
 *//4. What is the count of the number of students in the school, whose name is Mathew and  
 //marks is 55* **val** x4 = splitvalue.filter(x => (x.\_1 == **"Mathew"** && x.\_4 == 55))  
 *println*(x4.count())  
  
}  
}

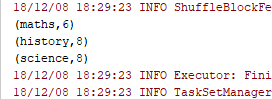
Output for **Read the text file, and create a tupled rdd.**



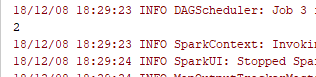
**Output for Find the count of total number of rows present.**

****

**Output for What is the distinct number of subjects present in the entire school**

****

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

****

**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?**

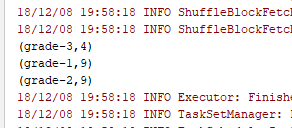
**4. What is the average score of students in each subject per grade?**

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

***Code logic***

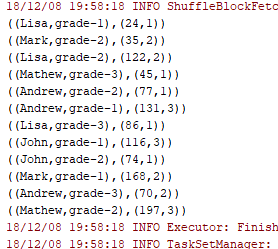
**package** core  
  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** SparkRDDDeepDiveTask3 {  
  
 **def** main(args: Array[String]): Unit = {  
  
 *println*(**"hey Scala"**)  
  
 **val** conf = **new** SparkConf().setMaster(**"local"**).setAppName(**"SimpleApp"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** x1 = sc.textFile(**"G:\\Bigdata\\Spark\\RDD\_DeepDive.txt"**);  
  
 *// 1. What is the count of students per grade in the school?* **val** splitvalue = x1.map(x => (x.split(**","**)(0), x.split(**","**)(1), x.split(**","**)(2), x.split(**","**)(3).toInt, x.split(**","**)(4).toInt))  
 **val** x3 = splitvalue.map(x => (x.\_3, 1)).  
 reduceByKey((a, b) => (a + b))  
 x3.foreach(*println*)  
  
 *//2. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in  
 //some other grade!)* **val** splitvalue1 = x1.map(x => ((x.split(**","**)(0), x.split(**","**)(2)), x.split(**","**)(3).toInt))  
 **val** x4 = splitvalue1.mapValues(y => (y,1))  
 **val** x5 = x4.reduceByKey((x,y) => (x.\_1 + y.\_1 , x.\_2 + y.\_2))  
 x5.foreach(*println*)  
 **val** x6 = x5.mapValues(x => ((1.0\*x.\_1)/x.\_2))  
 x6.foreach(*println*)  
  
 *//3. What is the average score of students in each subject across all grades?* **val** splitvalue2 = x1.map(x => ((x.split(**","**)(0),x.split(**","**)(1)), x.split(**","**)(3).toInt))  
 **val** x7 = splitvalue2.mapValues(y => (y,1))  
 **val** x8 = x7.reduceByKey((x,y) => (x.\_1 + y.\_1 , x.\_2 + y.\_2))  
 **val** x9 = x8.mapValues(x => ((1.0\*x.\_1)/x.\_2))  
 x9.foreach(*println*)  
  
 *//4. What is the average score of students in each subject per grade?* **val** splitvalue3 = x1.map(x => ((x.split(**","**)(1),x.split(**","**)(2)), x.split(**","**)(3).toInt))  
 **val** x10 = splitvalue3.mapValues(y => (y,1))  
 **val** x11 = x10.reduceByKey((x,y) => (x.\_1 + y.\_1 , x.\_2 + y.\_2))  
 **val** x12 = x11.mapValues(x => ((1.0\*x.\_1)/x.\_2))  
 x12.foreach(*println*)  
  
 *//5. For all students in grade-2, how many have average score greater than 50?* **val** splitvalue4 = x1.map(x => ((x.split(**","**)(0),x.split(**","**)(2)), x.split(**","**)(3).toInt))  
 **val** x13 = splitvalue4.mapValues(y => (y,1))  
 **val** x14 = x13.reduceByKey((x,y) => (x.\_1 + y.\_1 , x.\_2 + y.\_2))  
 **val** x15 = x14.mapValues(x => ((1.0\*x.\_1)/x.\_2))  
  
 **val** filteroutput =*println*(x15.filter(x => x.\_1.\_2 == **"grade-2"** && x.\_2 >50).count())  
 **val** filteroutput1 =x15.filter(x => x.\_1.\_2 == **"grade-2"** && x.\_2 >50)  
 filteroutput1.foreach(*println*)  
  
  
  
  
 }  
  
 }

**Output for What is the count of students per grade in the school?**

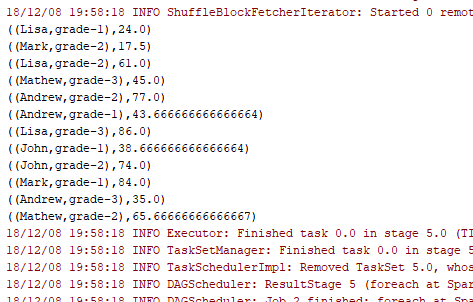
****

**Output for Find the average of each student (Note - Mathew is grade-1, is different from Mathew in**

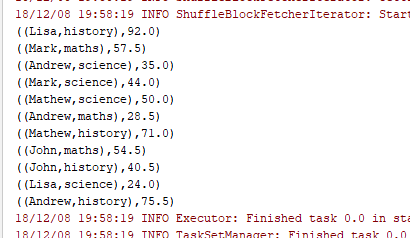
**some other grade!)**

****

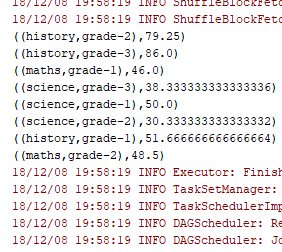
**Output for What is the average score of students in each subject across all grades?**

****

**Output for What is the average score of students in each subject per grade?**

****

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

****

**Problem Statement 3:**

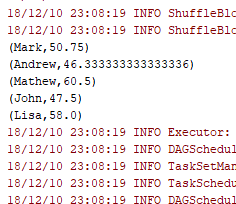
**Are there any students in the college that satisfy the below criteria:**

**1. Average score per student\_name across all grades is same as average score per Big Data Hadoop and Spark Development student\_name per grade**

**Code logic**

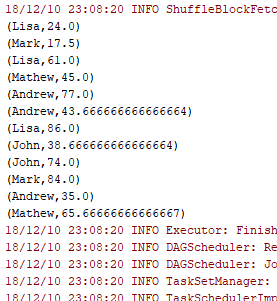
**package** core  
  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
**object** SparkRDDDeepDiveTask4 {  
  
 **def** main(args: Array[String]): Unit = {  
  
 *println*(**"hey Scala"**)  
  
 **val** conf = **new** SparkConf().setMaster(**"local"**).setAppName(**"SimpleApp"**)  
 **val** sc = **new** SparkContext(conf)  
  
 **val** x1 = sc.textFile(**"G:\\Bigdata\\Spark\\RDD\_DeepDive.txt"**);  
  
 **val** splitvalue2 = x1.map(x => (x.split(**","**)(0), x.split(**","**)(3).toInt))  
 **val** x7 = splitvalue2.mapValues(y => (y,1))  
 **val** x8 = x7.reduceByKey((x,y) => (x.\_1 + y.\_1 , x.\_2 + y.\_2))  
 **val** x9 = x8.mapValues(x => ((1.0\*x.\_1)/x.\_2))  
 x9.foreach(*println*)

**Output**



**val** splitvalue3 = x1.map(x => ((x.split(**","**)(0),x.split(**","**)(2)), x.split(**","**)(3).toInt))  
 **val** x10 = splitvalue3.mapValues(y => (y,1))  
 **val** x11 = x10.reduceByKey((x,y) => (x.\_1 + y.\_1 , x.\_2 + y.\_2))  
 **val** x12 = x11.mapValues(x => ((1.0\*x.\_1)/x.\_2))   
 **val** x13 = x12.map(x=>(x.\_1.\_1,x.\_2))  
 x13.foreach(*println*)

**Output**



**val** commonval = x9.intersection(x13)  
 commonval.foreach(*println*)  
}  
}

**No Common records to fetch**