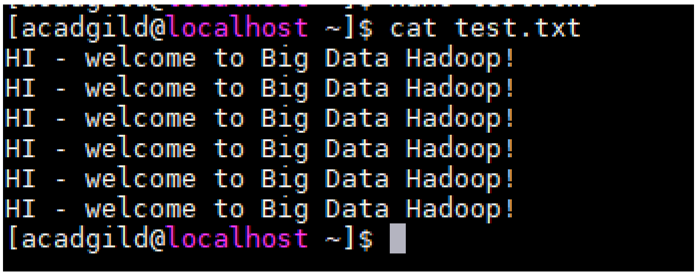
Assignment - 19.1

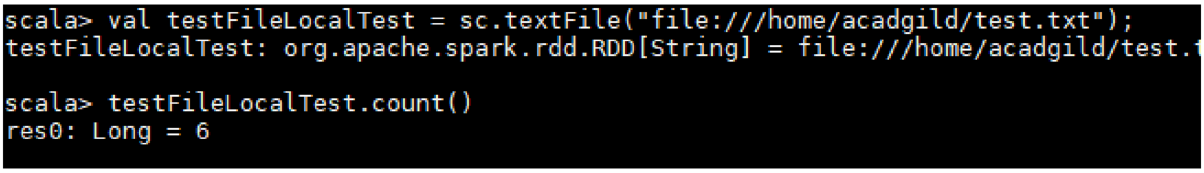
Task1.1: Write a program to read a text file and print the number of rows of data in the document.

Answer: Now first we create a file in local and note the no. of records over here to be 6 rows of records as shown in the below screenshot.



```
[acadgild@localhost ~]$ cat test.txt
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
[acadgild@localhost ~]$
```

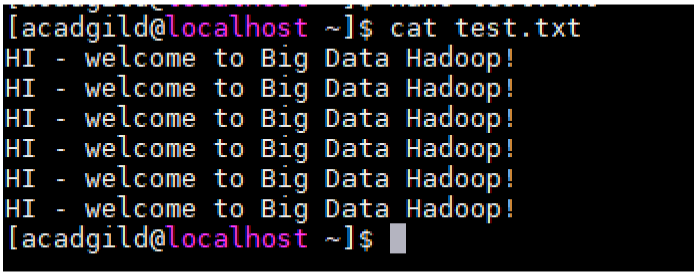
Now we will load this file to spark and get the no. of rows of record using the below commands as shown in below screenshot.



```
scala> val testFileLocalTest = sc.textFile("file:///home/acadgild/test.txt");
testFileLocalTest: org.apache.spark.rdd.RDD[String] = file:///home/acadgild/test.t

scala> testFileLocalTest.count()
res0: Long = 6
```

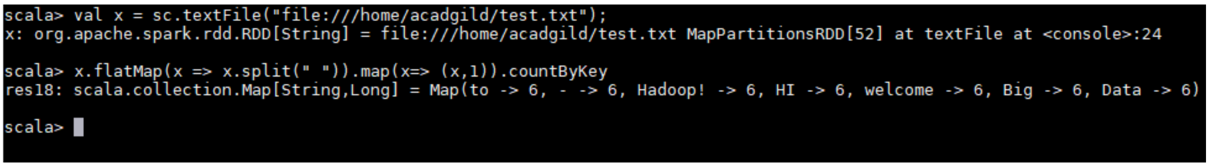
Task1.2: Write a program to read a text file and print the number of words in the document.

Answer: Input file content is as below.



```
[acadgild@localhost ~]$ cat test.txt
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
HI - welcome to Big Data Hadoop!
[acadgild@localhost ~]$
```

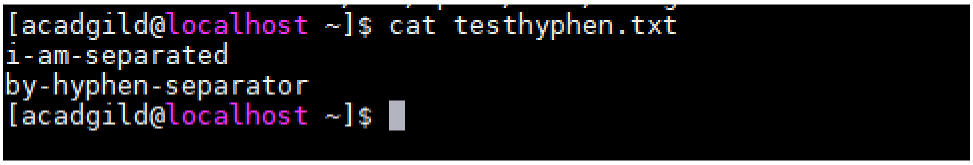
Wordcount is as mentioned below.



```
scala> val x = sc.textFile("file:///home/acadgild/test.txt");
x: org.apache.spark.rdd.RDD[String] = file:///home/acadgild/test.txt MapPartitionsRDD[52] at textFile at <console>:24

scala> x.flatMap(x => x.split(" ")).map(x=> (x,1)).countByKey
res18: scala.collection.Map[String,Long] = Map(to -> 6, - -> 6, Hadoop! -> 6, HI -> 6, welcome -> 6, Big -> 6, Data -> 6)

scala>
```

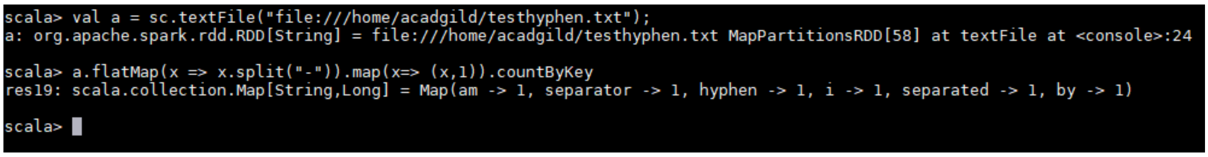
Task1.3: Write a program to read a text file and print the number of words in the document. Write a spark code, to obtain the count of the total number of words present in the document.

Answer: input file we can see that the words are separated by hyphen



```
[acadgild@localhost ~]$ cat testhyphen.txt
i-am-separated
by-hyphen-separator
[acadgild@localhost ~]$
```

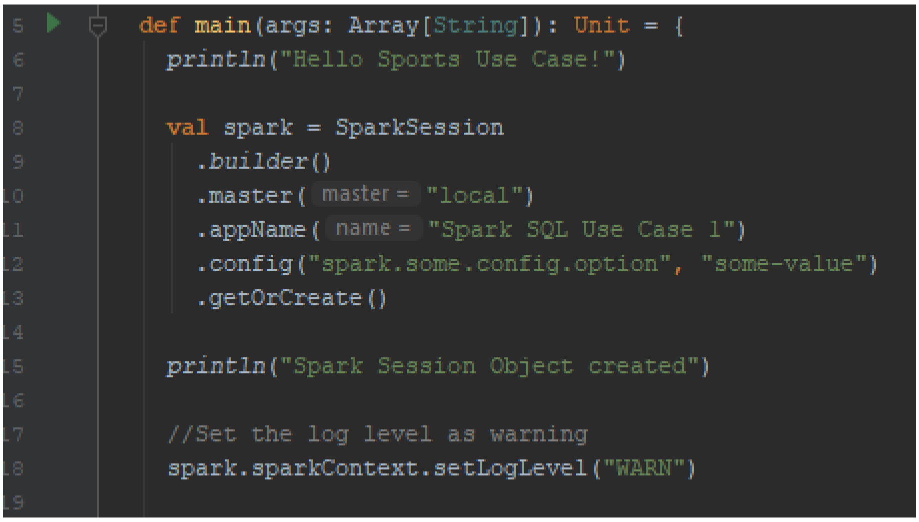
Word count is as follows.



```
scala> val a = sc.textFile("file:///home/acadgild/testhyphen.txt");
a: org.apache.spark.rdd.RDD[String] = file:///home/acadgild/testhyphen.txt MapPartitionsRDD[58] at textFile at <console>:24

scala> a.flatMap(x => x.split("-")).map(x=> (x,1)).countByKey
res19: scala.collection.Map[String,Long] = Map(am -> 1, separator -> 1, hyphen -> 1, i -> 1, separated -> 1, by -> 1)

scala>
```

Task 2 and Task 3 – Student Management System

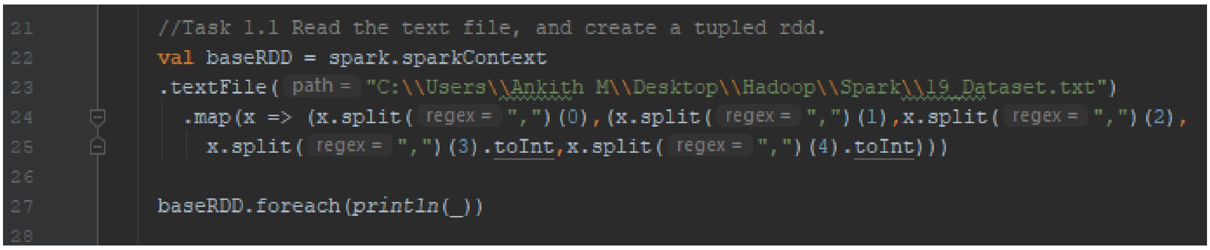
Here the below set of Problems are solved using **Idea IntelliJ.** So initially we have created the Spark Session Object as mentioned below.



```
def main(args: Array[String]): Unit = {
  println("Hello Sports Use Case!")

  val spark = SparkSession
    .builder()
    .master( master = "local")
    .appName( name = "Spark SQL Use Case 1")
    .config("spark.some.config.option", "some-value")
    .getOrCreate()

  println("Spark Session Object created")

  //Set the log level as warning
  spark.sparkContext.setLogLevel("WARN")
```

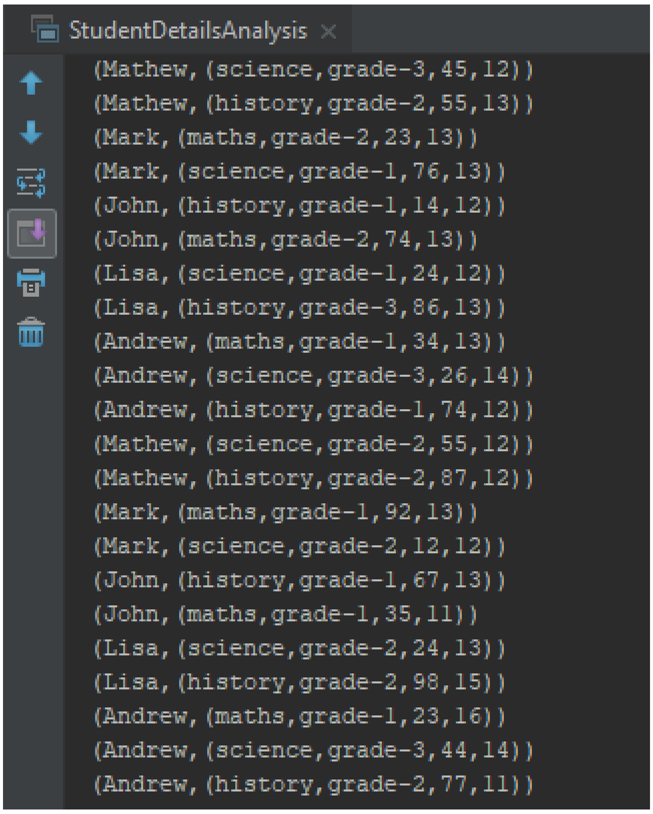
Problem1.1: Read the text file, and create a tupled rdd.

Answer:



```
//Task 1.1 Read the text file, and create a tupled rdd.
val baseRDD = spark.sparkContext
.textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x => (x.split( regex = ",")(0),(x.split( regex = ",")(1),x.split( regex = ",")(2),
    x.split( regex = ",")(3).toInt,x.split( regex = ",")(4).toInt)))

baseRDD.foreach(println(_))
```

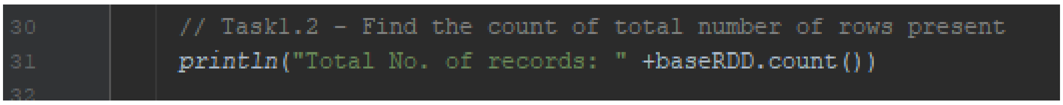
Output:



```
(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))
```

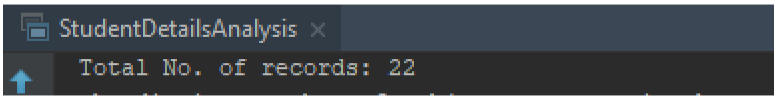
__Problem 1.2:__ Find the count of total number of rows present.

__Answer:__



```
// Task1.2 - Find the count of total number of rows present
println("Total No. of records: " +baseRDD.count())
```

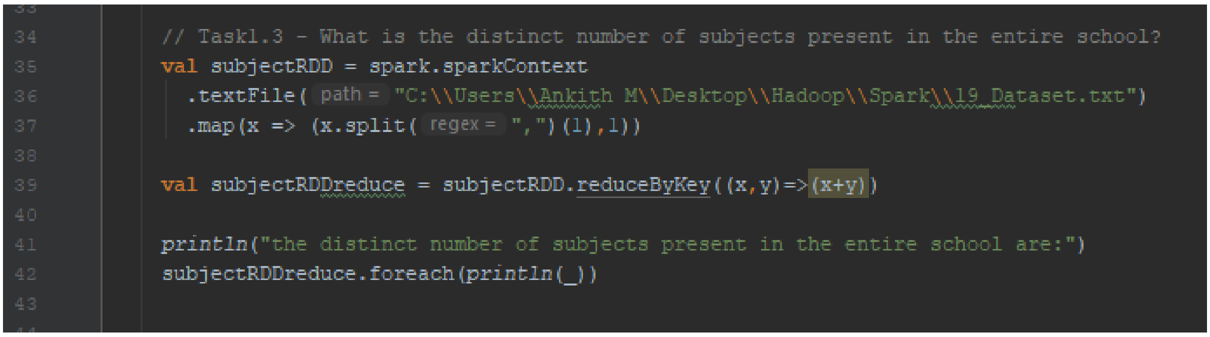
Output:



```
Total No. of records: 22
```

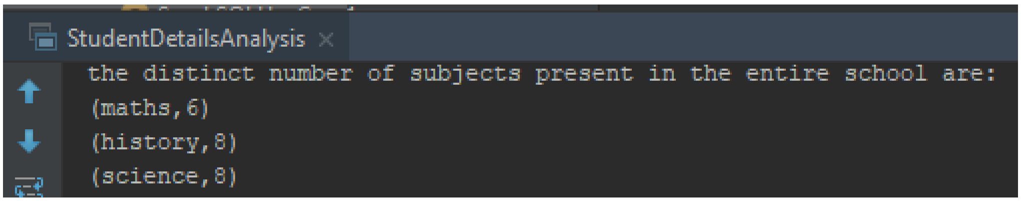
__Problem1.3:__ What is the distinct number of subjects present in the entire school?

__Answer:__



```
// Task1.3 - What is the distinct number of subjects present in the entire school?
val subjectRDD = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x => (x.split( regex = ",")(1),1))

val subjectRDDreduce = subjectRDD.reduceByKey((x,y)=>(x+y))

println("the distinct number of subjects present in the entire school are:")
subjectRDDreduce.foreach(println(_))
```

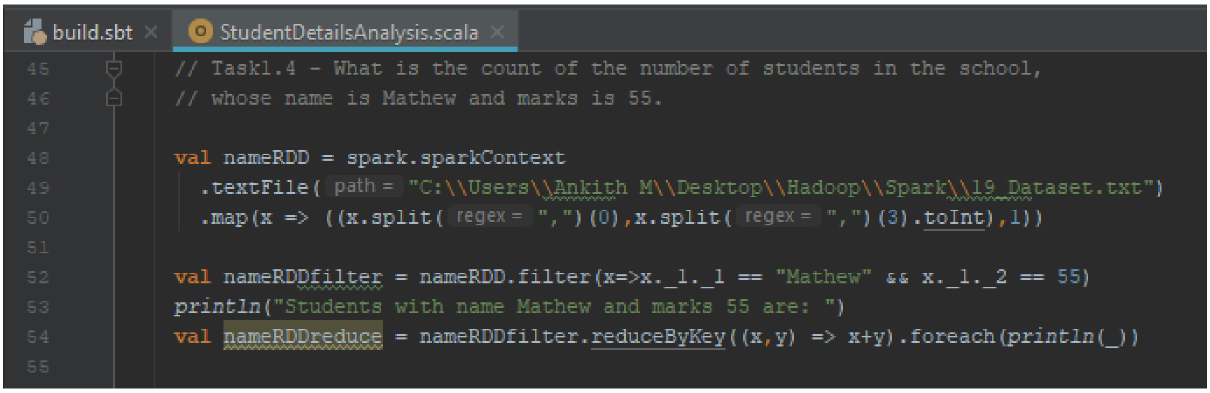
Output:



```
the distinct number of subjects present in the entire school are:
(maths,6)
(history,8)
(science,8)
```

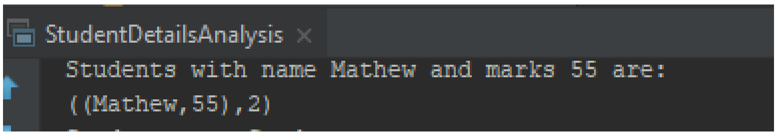
Problem1.4: What is the count of the number of students in the school, whose name is Mathew and marks is 55?

Answer:



```
// Task1.4 - What is the count of the number of students in the school,
// whose name is Mathew and marks is 55.

val nameRDD = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x => ((x.split( regex = ",")(0),x.split( regex = ",")(3).toInt),1))

val nameRDDfilter = nameRDD.filter(x=>x._1._1 == "Mathew" && x._1._2 == 55)
println("Students with name Mathew and marks 55 are: ")
val nameRDDreduce = nameRDDfilter.reduceByKey((x,y) => x+y).foreach(println(_))
```

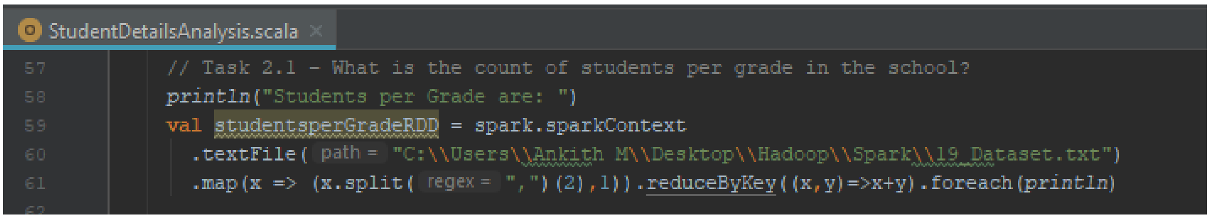
Output:



```
Students with name Mathew and marks 55 are:
((Mathew,55),2)
```

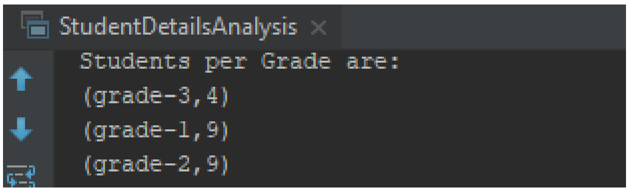
Problem2.1: What is the count of students per grade in the school?

Answer:



```
// Task 2.1 - What is the count of students per grade in the school?
println("Students per Grade are: ")
val studentsperGradeRDD = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x => (x.split( regex = ",")(2),1)).reduceByKey((x,y)=>x+y).foreach(println)
```

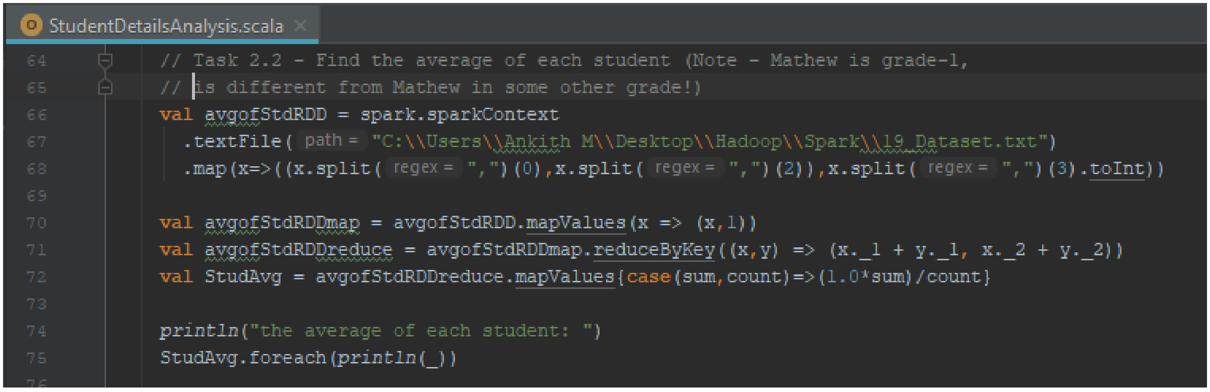
Output:



```
Students per Grade are:
(grade-3,4)
(grade-1,9)
(grade-2,9)
```

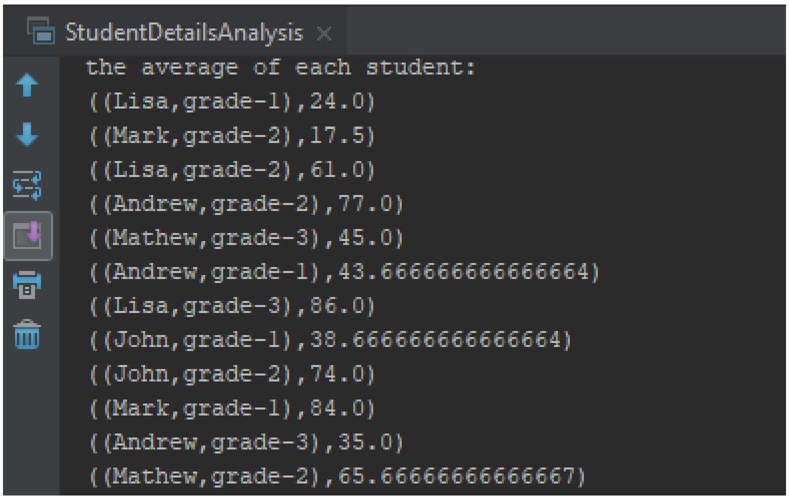
Problem2.2: Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

Answer:



```
// Task 2.2 - Find the average of each student (Note - Mathew is grade-1,
// is different from Mathew in some other grade!)
val avgofStdRDD = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x=>((x.split( regex = ",")(0),x.split( regex = ",")(2)),x.split( regex = ",")(3).toInt))

val avgofStdRDDmap = avgofStdRDD.mapValues(x => (x,1))
val avgofStdRDDreduce = avgofStdRDDmap.reduceByKey((x,y) => (x._1 + y._1, x._2 + y._2))
val StudAvg = avgofStdRDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

println("the average of each student: ")
StudAvg.foreach(println(_))
```

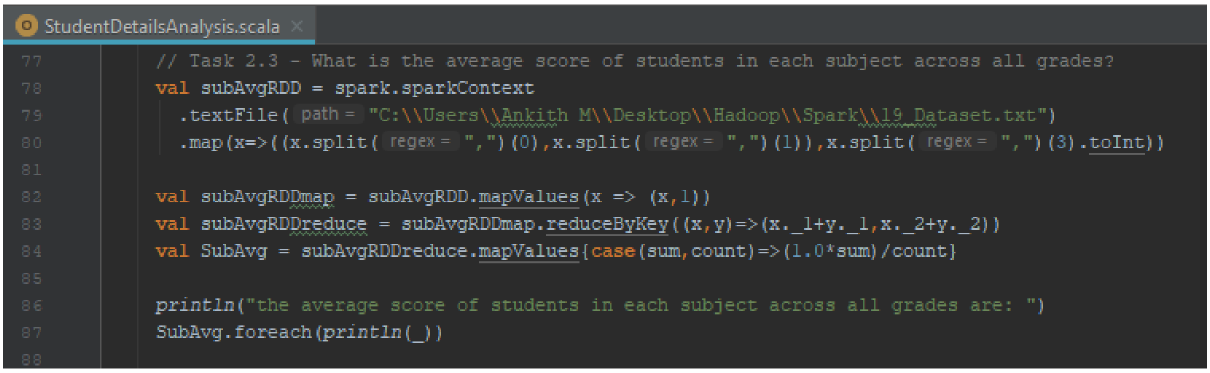
Output:



```
the average of each student:
((Lisa,grade-1),24.0)
((Mark,grade-2),17.5)
((Lisa,grade-2),61.0)
((Andrew,grade-2),77.0)
((Mathew,grade-3),45.0)
((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)
```

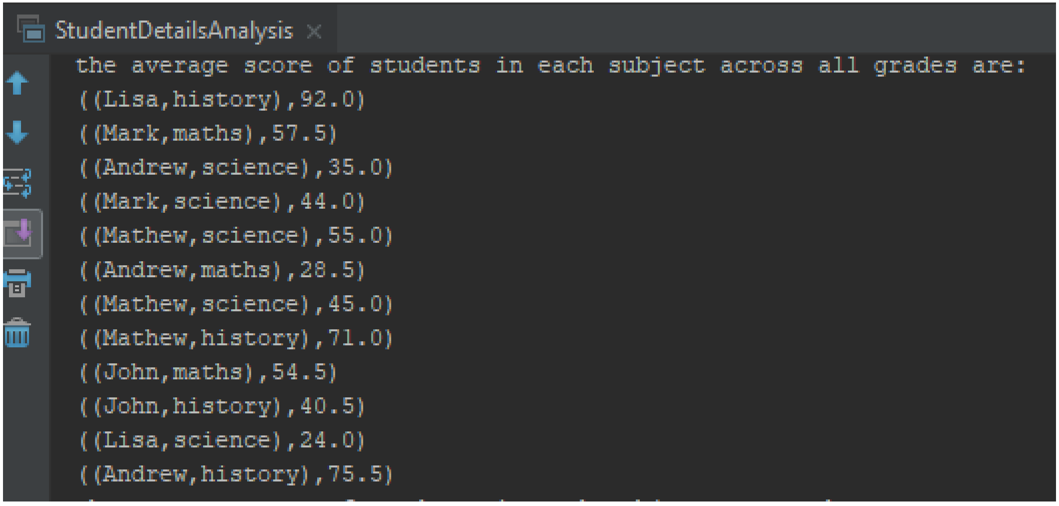
Problem2.3: What is the average score of students in each subject across all grades?

Answer:



```
// Task 2.3 - What is the average score of students in each subject across all grades?
val subAvgRDD = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x=>((x.split( regex = ",")(0),x.split( regex = ",")(1)),x.split( regex = ",")(3).toInt))

val subAvgRDDmap = subAvgRDD.mapValues(x => (x,1))
val subAvgRDDreduce = subAvgRDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
val SubAvg = subAvgRDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

println("the average score of students in each subject across all grades are: ")
SubAvg.foreach(println(_))
```

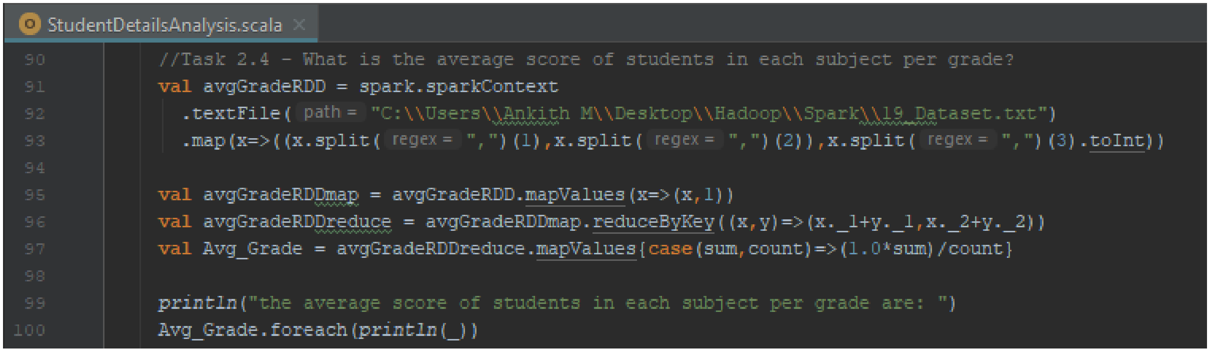
Output:



```
the average score of students in each subject across all grades are:
((Lisa,history),92.0)
((Mark,maths),57.5)
((Andrew,science),35.0)
((Mark,science),44.0)
((Mathew,science),55.0)
((Andrew,maths),28.5)
((Mathew,science),45.0)
((Mathew,history),71.0)
((John,maths),54.5)
((John,history),40.5)
((Lisa,science),24.0)
((Andrew,history),75.5)
```

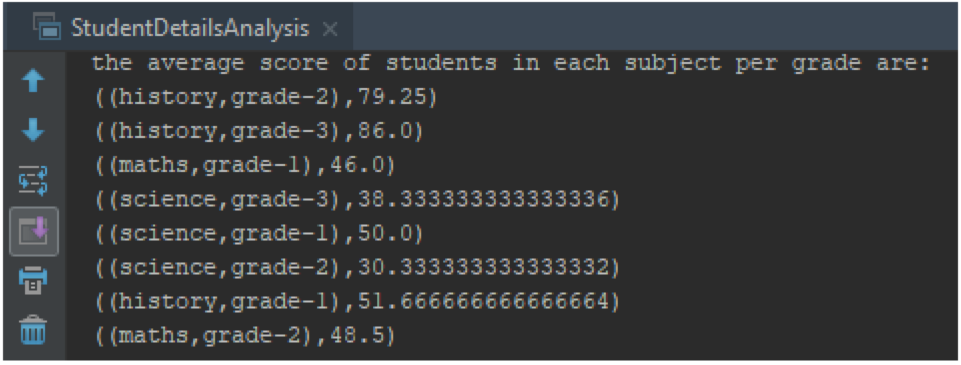
__Problem2.4:__ What is the average score of students in each subject per grade?

__**Answer:**__



```scala
//Task 2.4 - What is the average score of students in each subject per grade?
val avgGradeRDD = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x=>((x.split( regex = ",")(1),x.split( regex = ",")(2)),x.split( regex = ",")(3).toInt))

val avgGradeRDDmap = avgGradeRDD.mapValues(x=>(x,1))
val avgGradeRDDreduce = avgGradeRDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
val Avg_Grade = avgGradeRDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

println("the average score of students in each subject per grade are: ")
Avg_Grade.foreach(println(_))
```

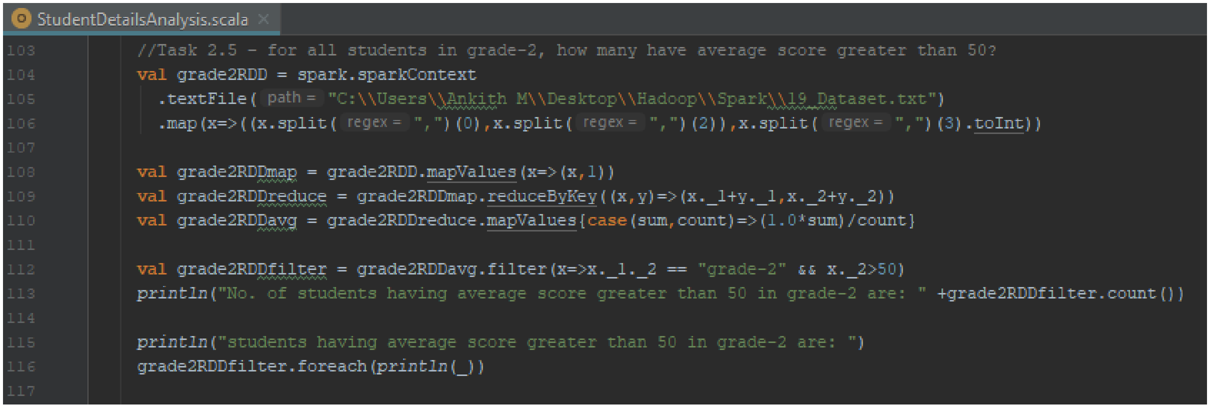
Output:



```
the average score of students in each subject per grade are:
((history,grade-2),79.25)
((history,grade-3),86.0)
((maths,grade-1),46.0)
((science,grade-3),38.333333333333336)
((science,grade-1),50.0)
((science,grade-2),30.333333333333332)
((history,grade-1),51.666666666666664)
((maths,grade-2),48.5)
```

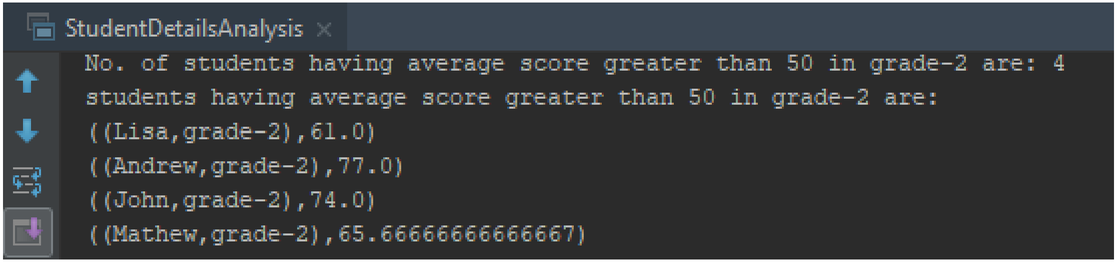
Problem2.5: For all students in grade-2, how many have average score greater than 50?

Answer:



```
//Task 2.5 - for all students in grade-2, how many have average score greater than 50?
val grade2RDD = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x=>((x.split( regex = ",")(0),x.split( regex = ",")(2)),x.split( regex = ",")(3).toInt))

val grade2RDDmap = grade2RDD.mapValues(x=>(x,1))
val grade2RDDreduce = grade2RDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
val grade2RDDavg = grade2RDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

val grade2RDDfilter = grade2RDDavg.filter(x=>x._1._2 == "grade-2" && x._2>50)
println("No. of students having average score greater than 50 in grade-2 are: " +grade2RDDfilter.count())

println("students having average score greater than 50 in grade-2 are: ")
grade2RDDfilter.foreach(println(_))
```

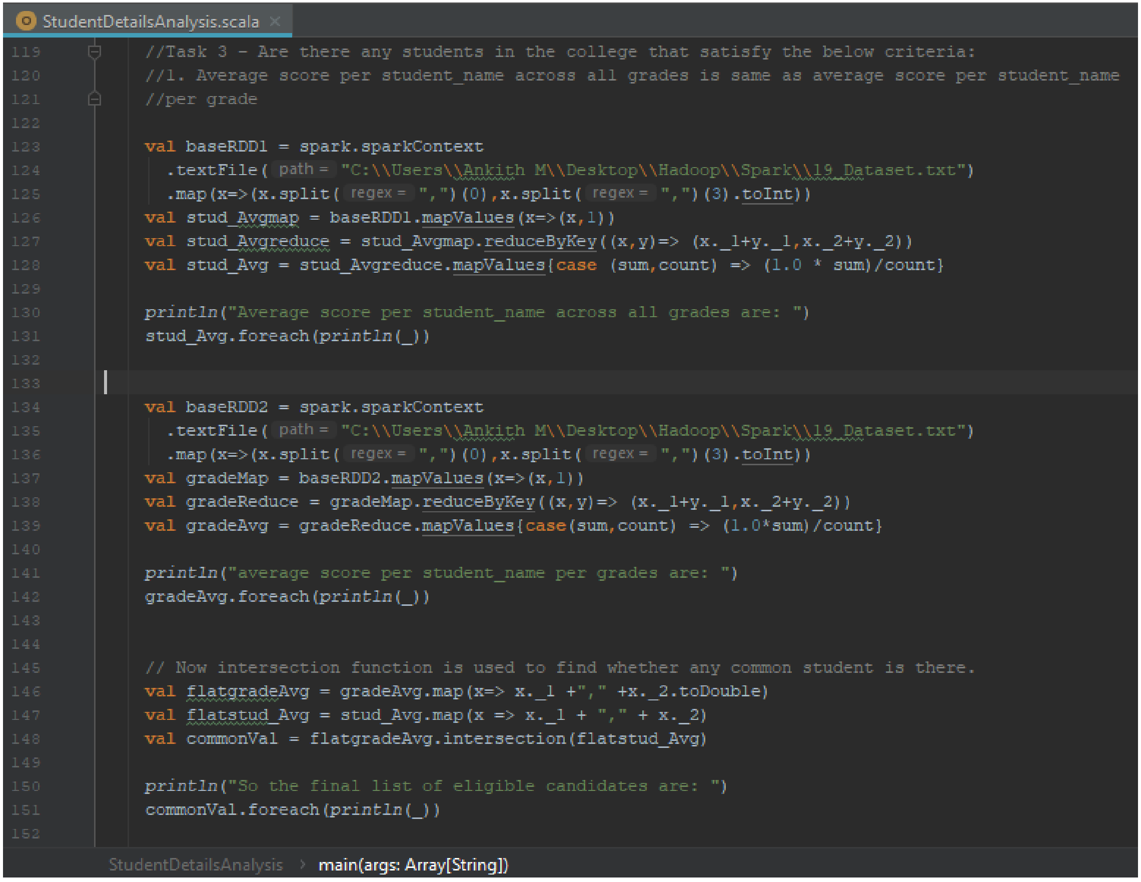
Output:



```
No. of students having average score greater than 50 in grade-2 are: 4
students having average score greater than 50 in grade-2 are:
((Lisa,grade-2),61.0)
((Andrew,grade-2),77.0)
((John,grade-2),74.0)
((Mathew,grade-2),65.66666666666667)
```

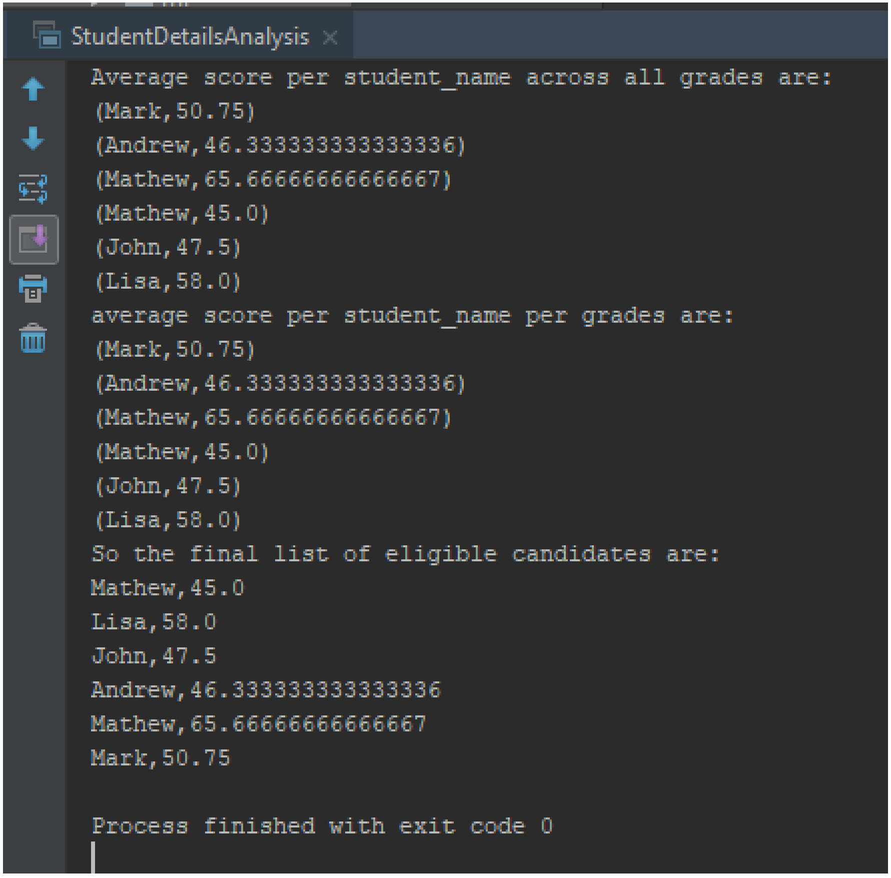
Problem 3: What is the average score of students in each subject per grade?

Answer: here we are using intersection concept to get the common records.



```
//Task 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 = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x=>(x.split( regex = ",")(0),x.split( regex = ",")(3).toInt))
val stud_Avgmap = baseRDD1.mapValues(x=>(x,1))
val stud_Avgreduce = stud_Avgmap.reduceByKey((x,y)=> (x._1+y._1,x._2+y._2))
val stud_Avg = stud_Avgreduce.mapValues{case (sum,count) => (1.0 * sum)/count}

println("Average score per student_name across all grades are: ")
stud_Avg.foreach(println(_))


val baseRDD2 = spark.sparkContext
  .textFile( path = "C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
  .map(x=>(x.split( regex = ",")(0),x.split( regex = ",")(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}

println("average score per student_name per grades are: ")
gradeAvg.foreach(println(_))


// Now intersection function is used to find whether any common student is there.
val flatgradeAvg = gradeAvg.map(x=> x._1 +"," +x._2.toDouble)
val flatstud_Avg = stud_Avg.map(x => x._1 + "," + x._2)
val commonVal = flatgradeAvg.intersection(flatstud_Avg)

println("So the final list of eligible candidates are: ")
commonVal.foreach(println(_))
```

Output:

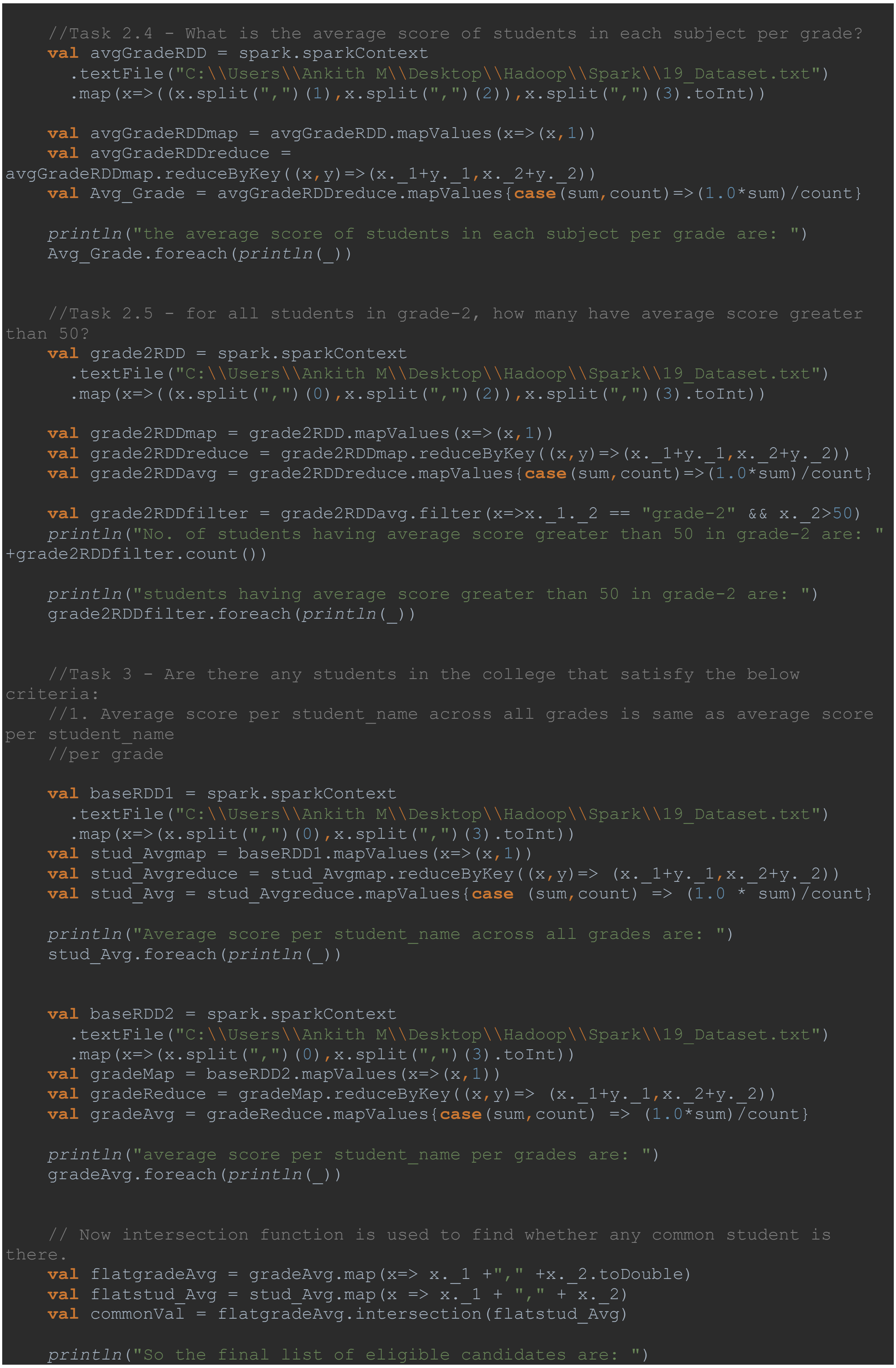


```
Average score per student_name across all grades are:
(Mark,50.75)
(Andrew,46.333333333333336)
(Mathew,65.66666666666667)
(Mathew,45.0)
(John,47.5)
(Lisa,58.0)
average score per student_name per grades are:
(Mark,50.75)
(Andrew,46.333333333333336)
(Mathew,65.66666666666667)
(Mathew,45.0)
(John,47.5)
(Lisa,58.0)
So the final list of eligible candidates are:
Mathew,45.0
Lisa,58.0
John,47.5
Andrew,46.333333333333336
Mathew,65.66666666666667
Mark,50.75

Process finished with exit code 0
```

================================End of Assignment =================================

The whole code dump used to complete "Student Use case" is mentioned below.

```
import org.apache.spark.sql.SparkSession

object StudentDetailsAnalysis {

  def main(args: Array[String]): Unit = {
    println("Hello Sports Use Case!")

    val spark = SparkSession
      .builder()
      .master("local")
      .appName("Spark SQL Use Case 1")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")

    //Set the log level as warning
    spark.sparkContext.setLogLevel("WARN")


    //Task 1.1 Read the text file, and create a tupled rdd.
    val baseRDD = spark.sparkContext
    .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x => (x.split(",")(0),(x.split(",")(1),x.split(",")(2),
        x.split(",")(3).toInt,x.split(",")(4).toInt)))

    baseRDD.foreach(println(_))


    // Task1.2 - Find the count of total number of rows present
    println("Total No. of records: " +baseRDD.count())


    // Task1.3 - What is the distinct number of subjects present in the entire
school?
    val subjectRDD = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x => (x.split(",")(1),1))

    val subjectRDDreduce = subjectRDD.reduceByKey((x,y)=>(x+y))

    println("the distinct number of subjects present in the entire school are:")
    subjectRDDreduce.foreach(println(_))


    // Task1.4 - What is the count of the number of students in the school,
    // whose name is Mathew and marks is 55.

    val nameRDD = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x => ((x.split(",")(0),x.split(",")(3).toInt),1))

    val nameRDDfilter = nameRDD.filter(x=>x._1._1 == "Mathew" && x._1._2 == 55)
    println("Students with name Mathew and marks 55 are: ")
    val nameRDDreduce = nameRDDfilter.reduceByKey((x,y) => x+y).foreach(println(_))


    // Task 2.1 - What is the count of students per grade in the school?
    println("Students per Grade are: ")
    val studentsperGradeRDD = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x => (x.split(",")(2),1)).reduceByKey((x,y)=>x+y).foreach(println)


    // Task 2.2 - Find the average of each student (Note - Mathew is grade-1,
    // is different from Mathew in some other grade!)
    val avgofStdRDD = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x=>((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))

    val avgofStdRDDmap = avgofStdRDD.mapValues(x => (x,1))
    val avgofStdRDDreduce = avgofStdRDDmap.reduceByKey((x,y) => (x._1 + y._1, x._2
+ y._2))
    val StudAvg = avgofStdRDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

    println("the average of each student: ")
    StudAvg.foreach(println(_))

    // Task 2.3 - What is the average score of students in each subject across all
grades?
    val subAvgRDD = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x=>((x.split(",")(0),x.split(",")(1)),x.split(",")(3).toInt))

    val subAvgRDDmap = subAvgRDD.mapValues(x => (x,1))
    val subAvgRDDreduce = subAvgRDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val SubAvg = subAvgRDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

    println("the average score of students in each subject across all grades are:
")
    SubAvg.foreach(println(_))
```



```
    //Task 2.4 - What is the average score of students in each subject per grade?
    val avgGradeRDD = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x=>((x.split(",")(1),x.split(",")(2)),x.split(",")(3).toInt))

    val avgGradeRDDmap = avgGradeRDD.mapValues(x=>(x,1))
    val avgGradeRDDreduce =
avgGradeRDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val Avg_Grade = avgGradeRDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

    println("the average score of students in each subject per grade are: ")
    Avg_Grade.foreach(println(_))


    //Task 2.5 - for all students in grade-2, how many have average score greater
than 50?
    val grade2RDD = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x=>((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))

    val grade2RDDmap = grade2RDD.mapValues(x=>(x,1))
    val grade2RDDreduce = grade2RDDmap.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val grade2RDDavg = grade2RDDreduce.mapValues{case(sum,count)=>(1.0*sum)/count}

    val grade2RDDfilter = grade2RDDavg.filter(x=>x._1._2 == "grade-2" && x._2>50)
    println("No. of students having average score greater than 50 in grade-2 are: "
+grade2RDDfilter.count())

    println("students having average score greater than 50 in grade-2 are: ")
    grade2RDDfilter.foreach(println(_))


    //Task 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 = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x=>(x.split(",")(0),x.split(",")(3).toInt))
    val stud_Avgmap = baseRDD1.mapValues(x=>(x,1))
    val stud_Avgreduce = stud_Avgmap.reduceByKey((x,y)=> (x._1+y._1,x._2+y._2))
    val stud_Avg = stud_Avgreduce.mapValues{case (sum,count) => (1.0 * sum)/count}

    println("Average score per student_name across all grades are: ")
    stud_Avg.foreach(println(_))


    val baseRDD2 = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\19_Dataset.txt")
      .map(x=>(x.split(",")(0),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}

    println("average score per student_name per grades are: ")
    gradeAvg.foreach(println(_))


    // Now intersection function is used to find whether any common student is
there.
    val flatgradeAvg = gradeAvg.map(x=> x._1 +"," +x._2.toDouble)
    val flatstud_Avg = stud_Avg.map(x => x._1 + "," + x._2)
    val commonVal = flatgradeAvg.intersection(flatstud_Avg)

    println("So the final list of eligible candidates are: ")
    commonVal.foreach(println(_))



  }
}
```

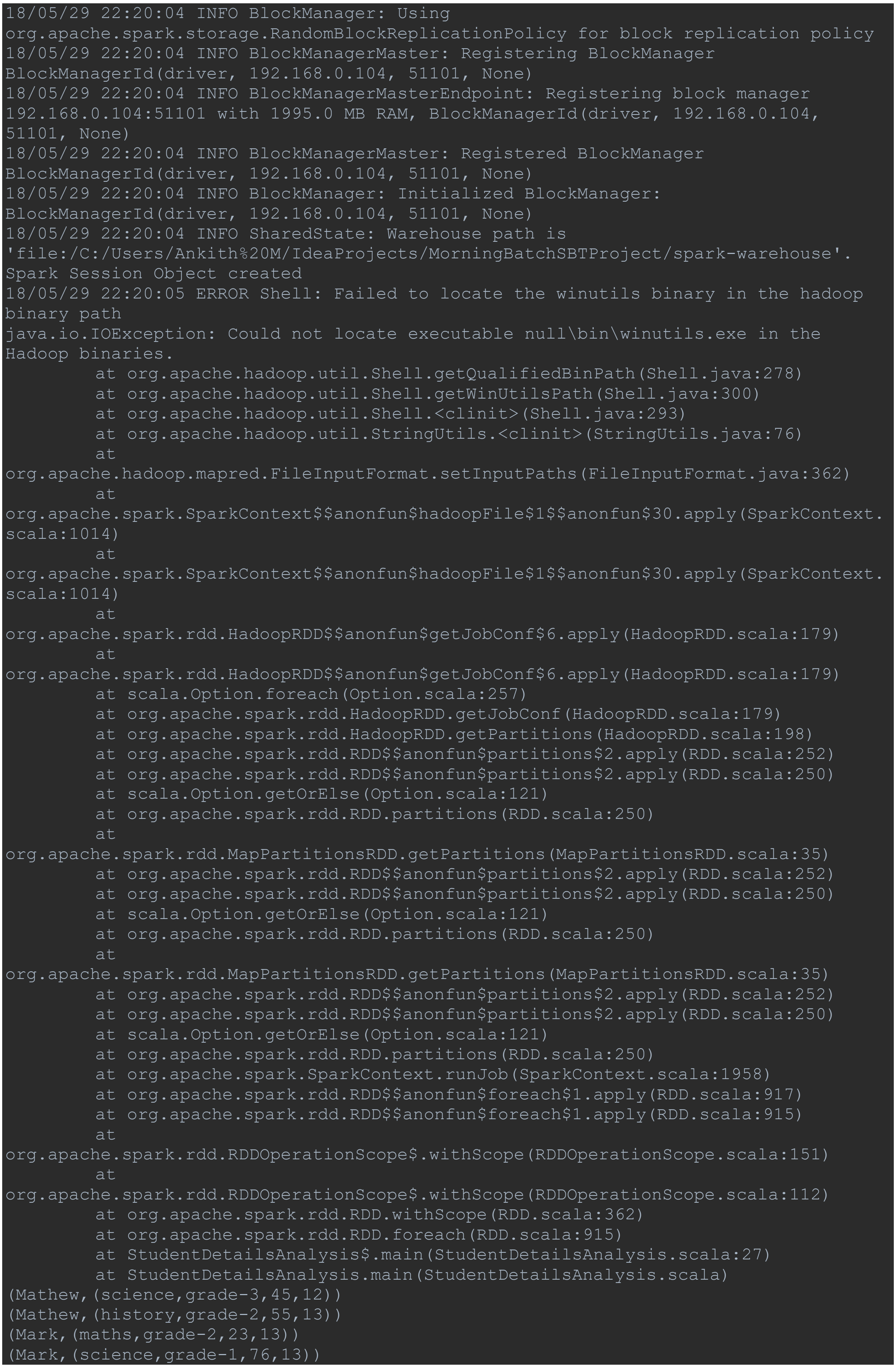
output interface for the above code is as below.

```
"C:\Program Files\Java\jdk1.8.0_141\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition
2018.1.1\lib\idea_rt.jar=51071:C:\Program Files\JetBrains\IntelliJ IDEA Community
Edition 2018.1.1\bin" -Dfile.encoding=UTF-8 -classpath "C:\Program
Files\Java\jdk1.8.0_141\jre\lib\charsets.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\deploy.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\access-bridge-64.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\cldrdata.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\dnsns.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\jaccess.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\jfxrt.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\localedata.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\nashorn.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\sunec.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\sunjce_provider.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\sunmscapi.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\sunpkcs11.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\ext\zipfs.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\javaws.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\jce.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\jfr.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\jfxswt.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\jsse.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\management-agent.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\plugin.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\resources.jar;C:\Program
Files\Java\jdk1.8.0_141\jre\lib\rt.jar;C:\Users\Ankith
M\IdeaProjects\MorningBatchSBTProject\target\scala-2.11\classes;C:\Users\Ankith
M\.ivy2\cache\org.scala-lang\scala-library\jars\scala-library-
2.11.7.jar;C:\Users\Ankith M\.ivy2\cache\org.spire-math\spire_2.11\jars\spire_2.11-
0.7.4.jar;C:\Users\Ankith M\.ivy2\cache\org.spire-math\spire-
macros_2.11\jars\spire-macros_2.11-0.7.4.jar;C:\Users\Ankith
M\.ivy2\cache\org.scalanlp\breeze_2.11\jars\breeze_2.11-0.12.jar;C:\Users\Ankith
M\.ivy2\cache\org.scalanlp\breeze-macros_2.11\jars\breeze-macros_2.11-
0.12.jar;C:\Users\Ankith M\.ivy2\cache\org.jpmml\pmml-schema\jars\pmml-schema-
1.2.15.jar;C:\Users\Ankith M\.ivy2\cache\org.jpmml\pmml-model\jars\pmml-model-
1.2.15.jar;C:\Users\Ankith M\.ivy2\cache\org.codehaus.janino\janino\jars\janino-
3.0.0.jar;C:\Users\Ankith M\.ivy2\cache\org.codehaus.janino\commons-
compiler\jars\commons-compiler-3.0.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.spark\spark-streaming_2.11\jars\spark-streaming_2.11-
2.1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.spark\spark-sql_2.11\jars\spark-
sql_2.11-2.1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.spark\spark-
sketch_2.11\jars\spark-sketch_2.11-2.1.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.spark\spark-mllib_2.11\jars\spark-mllib_2.11-
2.1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.spark\spark-mllib-
local_2.11\jars\spark-mllib-local_2.11-2.1.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.spark\spark-graphx_2.11\jars\spark-graphx_2.11-
2.1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.spark\spark-
catalyst_2.11\jars\spark-catalyst_2.11-2.1.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.parquet\parquet-jackson\jars\parquet-jackson-
1.8.1.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.parquet\parquet-
hadoop\jars\parquet-hadoop-1.8.1.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.parquet\parquet-format\jars\parquet-format-2.3.0-
incubating.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.parquet\parquet-
encoding\jars\parquet-encoding-1.8.1.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.parquet\parquet-common\jars\parquet-common-
1.8.1.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.parquet\parquet-
column\jars\parquet-column-1.8.1.jar;C:\Users\Ankith
```

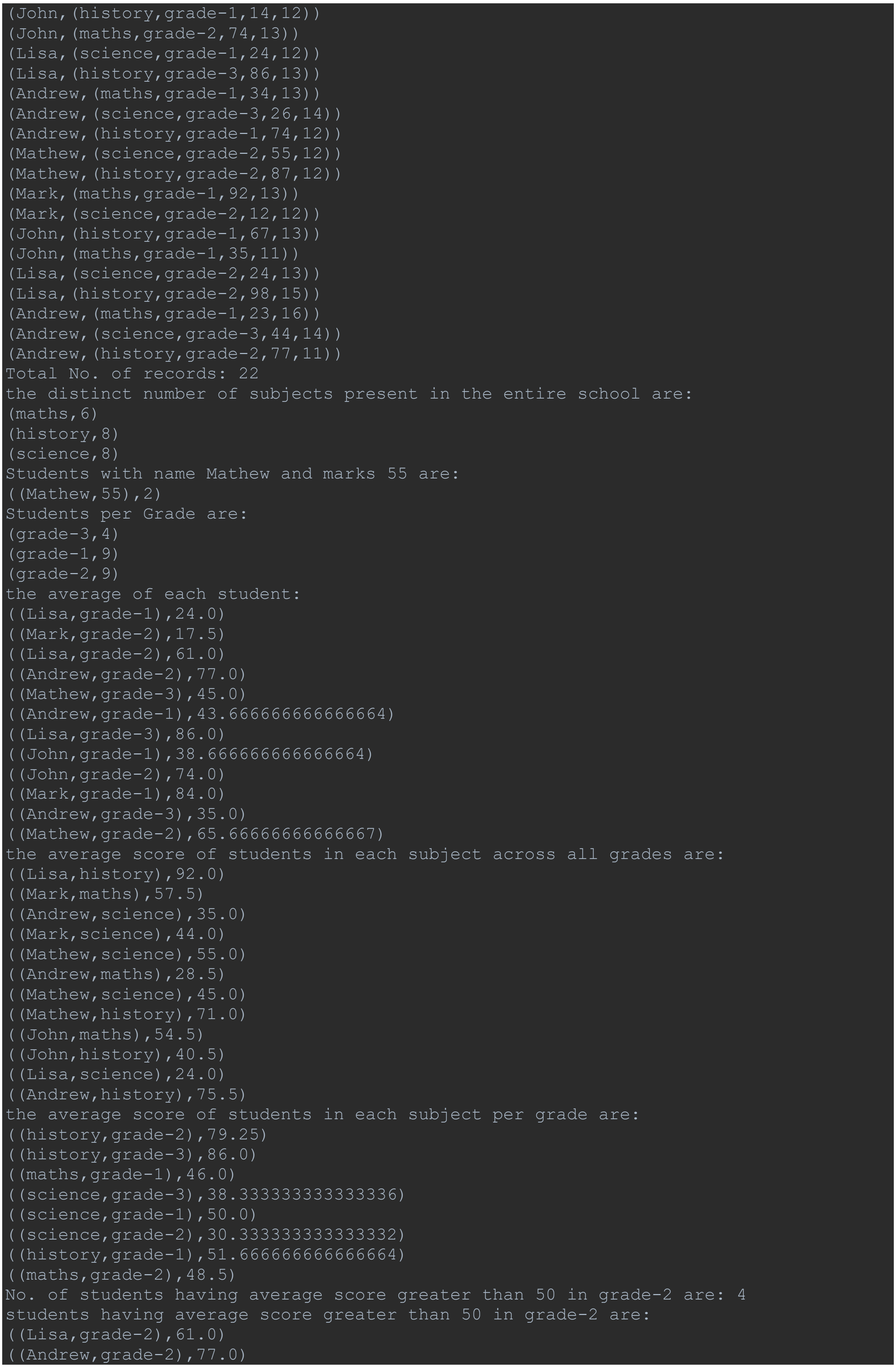
```
M\.ivy2\cache\org.antlr\antlr4-runtime\jars\antlr4-runtime-
4.5.3.jar;C:\Users\Ankith
M\.ivy2\cache\net.sourceforge.f2j\arpack_combined_all\jars\arpack_combined_all-
0.1.jar;C:\Users\Ankith M\.ivy2\cache\net.sf.opencsv\opencsv\jars\opencsv-
2.3.jar;C:\Users\Ankith M\.ivy2\cache\commons-codec\commons-codec\jars\commons-
codec-1.10.jar;C:\Users\Ankith M\.ivy2\cache\com.univocity\univocity-
parsers\jars\univocity-parsers-2.2.1.jar;C:\Users\Ankith
M\.ivy2\cache\com.github.rwl\jtransforms\jars\jtransforms-2.4.0.jar;C:\Users\Ankith
M\.ivy2\cache\com.github.fommil.netlib\core\jars\core-1.1.2.jar;C:\Users\Ankith
M\.ivy2\cache\com.chuusai\shapeless_2.11\bundles\shapeless_2.11-
2.0.0.jar;C:\Users\Ankith M\.ivy2\cache\xmlenc\xmlenc\jars\xmlenc-
0.52.jar;C:\Users\Ankith M\.ivy2\cache\oro\oro\jars\oro-2.0.8.jar;C:\Users\Ankith
M\.ivy2\cache\org.xerial.snappy\snappy-java\bundles\snappy-java-
1.1.2.6.jar;C:\Users\Ankith M\.ivy2\cache\org.tukaani\xz\jars\xz-
1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.spark-project.spark\unused\jars\unused-
1.0.0.jar;C:\Users\Ankith M\.ivy2\cache\org.sonatype.sisu.inject\cglib\jars\cglib-
2.2.1-v20090111.jar;C:\Users\Ankith M\.ivy2\cache\org.slf4j\slf4j-
log4j12\jars\slf4j-log4j12-1.7.16.jar;C:\Users\Ankith
M\.ivy2\cache\org.slf4j\slf4j-api\jars\slf4j-api-1.7.16.jar;C:\Users\Ankith
M\.ivy2\cache\org.slf4j\jul-to-slf4j\jars\jul-to-slf4j-1.7.16.jar;C:\Users\Ankith
M\.ivy2\cache\org.slf4j\jcl-over-slf4j\jars\jcl-over-slf4j-
1.7.16.jar;C:\Users\Ankith M\.ivy2\cache\org.scala-lang.modules\scala-
xml_2.11\bundles\scala-xml_2.11-1.0.4.jar;C:\Users\Ankith M\.ivy2\cache\org.scala-
lang.modules\scala-parser-combinators_2.11\bundles\scala-parser-combinators_2.11-
1.0.4.jar;C:\Users\Ankith M\.ivy2\cache\org.scala-lang\scalap\jars\scalap-
2.11.7.jar;C:\Users\Ankith M\.ivy2\cache\org.scala-lang\scala-reflect\jars\scala-
reflect-2.11.7.jar;C:\Users\Ankith M\.ivy2\cache\org.scala-lang\scala-
compiler\jars\scala-compiler-2.11.7.jar;C:\Users\Ankith
M\.ivy2\cache\org.roaringbitmap\RoaringBitmap\bundles\RoaringBitmap-
0.5.11.jar;C:\Users\Ankith M\.ivy2\cache\org.objenesis\objenesis\jars\objenesis-
2.1.jar;C:\Users\Ankith M\.ivy2\cache\org.mortbay.jetty\jetty-util\jars\jetty-util-
6.1.26.jar;C:\Users\Ankith M\.ivy2\cache\org.json4s\json4s-
jackson_2.11\jars\json4s-jackson_2.11-3.2.11.jar;C:\Users\Ankith
M\.ivy2\cache\org.json4s\json4s-core_2.11\jars\json4s-core_2.11-
3.2.11.jar;C:\Users\Ankith M\.ivy2\cache\org.json4s\json4s-ast_2.11\jars\json4s-
ast_2.11-3.2.11.jar;C:\Users\Ankith
M\.ivy2\cache\org.javassist\javassist\bundles\javassist-3.18.1-
GA.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.jersey.media\jersey-media-
jaxb\jars\jersey-media-jaxb-2.22.2.jar;C:\Users\Ankith
M\.ivy2\cache\org.glassfish.jersey.core\jersey-server\jars\jersey-server-
2.22.2.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.jersey.core\jersey-
common\jars\jersey-common-2.22.2.jar;C:\Users\Ankith
M\.ivy2\cache\org.glassfish.jersey.core\jersey-client\jars\jersey-client-
2.22.2.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.jersey.containers\jersey-
container-servlet-core\jars\jersey-container-servlet-core-
2.22.2.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.jersey.containers\jersey-
container-servlet\jars\jersey-container-servlet-2.22.2.jar;C:\Users\Ankith
M\.ivy2\cache\org.glassfish.jersey.bundles.repackaged\jersey-guava\bundles\jersey-
guava-2.22.2.jar;C:\Users\Ankith
M\.ivy2\cache\org.glassfish.hk2.external\javax.inject\jars\javax.inject-2.4.0-
b34.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.hk2.external\aopalliance-
repackaged\jars\aopalliance-repackaged-2.4.0-b34.jar;C:\Users\Ankith
M\.ivy2\cache\org.glassfish.hk2\osgi-resource-locator\jars\osgi-resource-locator-
1.0.1.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.hk2\hk2-utils\jars\hk2-utils-
2.4.0-b34.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.hk2\hk2-locator\jars\hk2-
locator-2.4.0-b34.jar;C:\Users\Ankith M\.ivy2\cache\org.glassfish.hk2\hk2-
api\jars\hk2-api-2.4.0-b34.jar;C:\Users\Ankith
M\.ivy2\cache\org.fusesource.leveldbjni\leveldbjni-all\bundles\leveldbjni-all-
1.8.jar;C:\Users\Ankith M\.ivy2\cache\org.codehaus.jackson\jackson-mapper-
asl\jars\jackson-mapper-asl-1.9.13.jar;C:\Users\Ankith
M\.ivy2\cache\org.codehaus.jackson\jackson-core-asl\jars\jackson-core-asl-
1.9.13.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.xbean\xbean-asm5-
shaded\bundles\xbean-asm5-shaded-4.4.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.ivy\ivy\jars\ivy-2.4.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.commons\commons-math3\jars\commons-math3-
3.4.1.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.commons\commons-
lang3\jars\commons-lang3-3.5.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.commons\commons-crypto\jars\commons-crypto-
```

```
1.0.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.commons\commons-
compress\jars\commons-compress-1.4.1.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.avro\avro-mapred\jars\avro-mapred-1.7.7-
hadoop2.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.avro\avro-ipc\jars\avro-ipc-
1.7.7-tests.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.avro\avro-ipc\jars\avro-
ipc-1.7.7.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.avro\avro\jars\avro-
1.7.7.jar;C:\Users\Ankith M\.ivy2\cache\net.sf.py4j\py4j\jars\py4j-
0.10.4.jar;C:\Users\Ankith M\.ivy2\cache\net.razorvine\pyrolite\jars\pyrolite-
4.13.jar;C:\Users\Ankith M\.ivy2\cache\net.jpountz.lz4\lz4\jars\lz4-
1.3.0.jar;C:\Users\Ankith M\.ivy2\cache\log4j\log4j\bundles\log4j-
1.2.17.jar;C:\Users\Ankith M\.ivy2\cache\javax.ws.rs\javax.ws.rs-
api\jars\javax.ws.rs-api-2.0.1.jar;C:\Users\Ankith
M\.ivy2\cache\javax.validation\validation-api\jars\validation-api-
1.1.0.Final.jar;C:\Users\Ankith M\.ivy2\cache\javax.servlet\javax.servlet-
api\jars\javax.servlet-api-3.1.0.jar;C:\Users\Ankith
M\.ivy2\cache\javax.inject\javax.inject\jars\javax.inject-1.jar;C:\Users\Ankith
M\.ivy2\cache\javax.annotation\javax.annotation-api\jars\javax.annotation-api-
1.2.jar;C:\Users\Ankith M\.ivy2\cache\io.dropwizard.metrics\metrics-
jvm\bundles\metrics-jvm-3.1.2.jar;C:\Users\Ankith
M\.ivy2\cache\io.dropwizard.metrics\metrics-json\bundles\metrics-json-
3.1.2.jar;C:\Users\Ankith M\.ivy2\cache\io.dropwizard.metrics\metrics-
graphite\bundles\metrics-graphite-3.1.2.jar;C:\Users\Ankith
M\.ivy2\cache\io.dropwizard.metrics\metrics-core\bundles\metrics-core-
3.1.2.jar;C:\Users\Ankith M\.ivy2\cache\commons-net\commons-net\jars\commons-net-
2.2.jar;C:\Users\Ankith M\.ivy2\cache\commons-httpclient\commons-
httpclient\jars\commons-httpclient-3.1.jar;C:\Users\Ankith M\.ivy2\cache\commons-
digester\commons-digester\jars\commons-digester-1.8.jar;C:\Users\Ankith
M\.ivy2\cache\commons-configuration\commons-configuration\jars\commons-
configuration-1.6.jar;C:\Users\Ankith M\.ivy2\cache\commons-cli\commons-
cli\jars\commons-cli-1.2.jar;C:\Users\Ankith M\.ivy2\cache\commons-
beanutils\commons-beanutils-core\jars\commons-beanutils-core-
1.8.0.jar;C:\Users\Ankith M\.ivy2\cache\commons-beanutils\commons-
beanutils\jars\commons-beanutils-1.7.0.jar;C:\Users\Ankith
M\.ivy2\cache\com.twitter\chill_2.11\jars\chill_2.11-0.8.0.jar;C:\Users\Ankith
M\.ivy2\cache\com.twitter\chill-java\jars\chill-java-0.8.0.jar;C:\Users\Ankith
M\.ivy2\cache\com.thoughtworks.paranamer\paranamer\jars\paranamer-
2.6.jar;C:\Users\Ankith M\.ivy2\cache\com.ning\compress-lzf\bundles\compress-lzf-
1.0.3.jar;C:\Users\Ankith M\.ivy2\cache\com.google.protobuf\protobuf-
java\bundles\protobuf-java-2.5.0.jar;C:\Users\Ankith
M\.ivy2\cache\com.google.inject\guice\jars\guice-3.0.jar;C:\Users\Ankith
M\.ivy2\cache\com.google.guava\guava\bundles\guava-14.0.1.jar;C:\Users\Ankith
M\.ivy2\cache\com.google.code.findbugs\jsr305\jars\jsr305-1.3.9.jar;C:\Users\Ankith
M\.ivy2\cache\com.fasterxml.jackson.module\jackson-module-
scala_2.11\bundles\jackson-module-scala_2.11-2.6.5.jar;C:\Users\Ankith
M\.ivy2\cache\com.fasterxml.jackson.module\jackson-module-
paranamer\bundles\jackson-module-paranamer-2.6.5.jar;C:\Users\Ankith
M\.ivy2\cache\com.fasterxml.jackson.core\jackson-databind\bundles\jackson-databind-
2.6.5.jar;C:\Users\Ankith M\.ivy2\cache\com.fasterxml.jackson.core\jackson-
core\bundles\jackson-core-2.6.5.jar;C:\Users\Ankith
M\.ivy2\cache\com.fasterxml.jackson.core\jackson-annotations\bundles\jackson-
annotations-2.6.5.jar;C:\Users\Ankith
M\.ivy2\cache\com.esotericsoftware\minlog\bundles\minlog-1.3.0.jar;C:\Users\Ankith
M\.ivy2\cache\com.esotericsoftware\kryo-shaded\bundles\kryo-shaded-
3.0.3.jar;C:\Users\Ankith
M\.ivy2\cache\com.clearspring.analytics\stream\jars\stream-
2.7.0.jar;C:\Users\Ankith M\.ivy2\cache\aopalliance\aopalliance\jars\aopalliance-
1.0.jar;C:\Users\Ankith M\.ivy2\cache\commons-collections\commons-
collections\jars\commons-collections-3.2.1.jar;C:\Users\Ankith
M\.ivy2\cache\commons-io\commons-io\jars\commons-io-2.1.jar;C:\Users\Ankith
M\.ivy2\cache\commons-lang\commons-lang\jars\commons-lang-2.5.jar;C:\Users\Ankith
M\.ivy2\cache\io.netty\netty\bundles\netty-3.8.0.Final.jar;C:\Users\Ankith
M\.ivy2\cache\io.netty\netty-all\jars\netty-all-4.0.42.Final.jar;C:\Users\Ankith
M\.ivy2\cache\net.java.dev.jets3t\jets3t\jars\jets3t-0.7.1.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.commons\commons-math\jars\commons-math-
2.1.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.curator\curator-
client\bundles\curator-client-2.4.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.curator\curator-framework\bundles\curator-framework-
2.4.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.curator\curator-
```

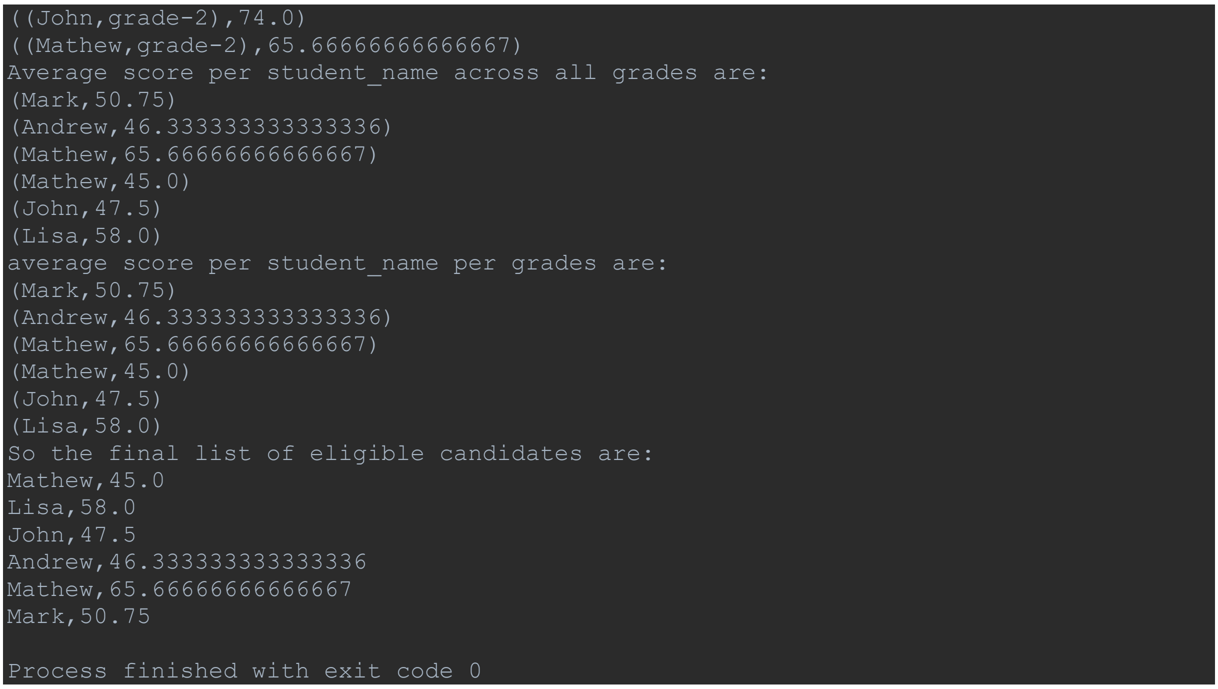
```
recipes\bundles\curator-recipes-2.4.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.hadoop\hadoop-annotations\jars\hadoop-annotations-
2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-auth\jars\hadoop-
auth-2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-
client\jars\hadoop-client-2.2.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.hadoop\hadoop-common\jars\hadoop-common-
2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-hdfs\jars\hadoop-
hdfs-2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-mapreduce-
client-app\jars\hadoop-mapreduce-client-app-2.2.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.hadoop\hadoop-mapreduce-client-common\jars\hadoop-
mapreduce-client-common-2.2.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.hadoop\hadoop-mapreduce-client-core\jars\hadoop-mapreduce-
client-core-2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-
mapreduce-client-jobclient\jars\hadoop-mapreduce-client-jobclient-
2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-mapreduce-client-
shuffle\jars\hadoop-mapreduce-client-shuffle-2.2.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.hadoop\hadoop-yarn-api\jars\hadoop-yarn-api-
2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-yarn-
client\jars\hadoop-yarn-client-2.2.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.hadoop\hadoop-yarn-common\jars\hadoop-yarn-common-
2.2.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.hadoop\hadoop-yarn-server-
common\jars\hadoop-yarn-server-common-2.2.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.spark\spark-core_2.11\jars\spark-core_2.11-
2.1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.spark\spark-
launcher_2.11\jars\spark-launcher_2.11-2.1.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.spark\spark-network-common_2.11\jars\spark-network-
common_2.11-2.1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.spark\spark-network-
shuffle_2.11\jars\spark-network-shuffle_2.11-2.1.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.spark\spark-tags_2.11\jars\spark-tags_2.11-
2.1.0.jar;C:\Users\Ankith M\.ivy2\cache\org.apache.spark\spark-
unsafe_2.11\jars\spark-unsafe_2.11-2.1.0.jar;C:\Users\Ankith
M\.ivy2\cache\org.apache.zookeeper\zookeeper\jars\zookeeper-
3.4.5.jar;C:\Users\Ankith
M\.ivy2\cache\org.scalatest\scalatest_2.11\bundles\scalatest_2.11-2.2.6.jar"
StudentDetailsAnalysis
Hello Sports Use Case!
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/05/29 22:20:02 INFO SparkContext: Running Spark version 2.1.0
18/05/29 22:20:03 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
18/05/29 22:20:03 INFO SecurityManager: Changing view acls to: Ankith M
18/05/29 22:20:03 INFO SecurityManager: Changing modify acls to: Ankith M
18/05/29 22:20:03 INFO SecurityManager: Changing view acls groups to:
18/05/29 22:20:03 INFO SecurityManager: Changing modify acls groups to:
18/05/29 22:20:03 INFO SecurityManager: SecurityManager: authentication disabled;
ui acls disabled; users  with view permissions: Set(Ankith M); groups with view
permissions: Set(); users  with modify permissions: Set(Ankith M); groups with
modify permissions: Set()
18/05/29 22:20:04 INFO Utils: Successfully started service 'sparkDriver' on port
51092.
18/05/29 22:20:04 INFO SparkEnv: Registering MapOutputTracker
18/05/29 22:20:04 INFO SparkEnv: Registering BlockManagerMaster
18/05/29 22:20:04 INFO BlockManagerMasterEndpoint: Using
org.apache.spark.storage.DefaultTopologyMapper for getting topology information
18/05/29 22:20:04 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
18/05/29 22:20:04 INFO DiskBlockManager: Created local directory at C:\Users\Ankith
M\AppData\Local\Temp\blockmgr-0fccc8aa-19e7-418f-b637-94c37dc006f7
18/05/29 22:20:04 INFO MemoryStore: MemoryStore started with capacity 1995.0 MB
18/05/29 22:20:04 INFO SparkEnv: Registering OutputCommitCoordinator
18/05/29 22:20:04 INFO Utils: Successfully started service 'SparkUI' on port 4040.
18/05/29 22:20:04 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at
http://192.168.0.104:4040
18/05/29 22:20:04 INFO Executor: Starting executor ID driver on host localhost
18/05/29 22:20:04 INFO Utils: Successfully started service
'org.apache.spark.network.netty.NettyBlockTransferService' on port 51101.
18/05/29 22:20:04 INFO NettyBlockTransferService: Server created on
192.168.0.104:51101
```



```
18/05/29 22:20:04 INFO BlockManager: Using
org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
18/05/29 22:20:04 INFO BlockManagerMaster: Registering BlockManager
BlockManagerId(driver, 192.168.0.104, 51101, None)
18/05/29 22:20:04 INFO BlockManagerMasterEndpoint: Registering block manager
192.168.0.104:51101 with 1995.0 MB RAM, BlockManagerId(driver, 192.168.0.104,
51101, None)
18/05/29 22:20:04 INFO BlockManagerMaster: Registered BlockManager
BlockManagerId(driver, 192.168.0.104, 51101, None)
18/05/29 22:20:04 INFO BlockManager: Initialized BlockManager:
BlockManagerId(driver, 192.168.0.104, 51101, None)
18/05/29 22:20:04 INFO SharedState: Warehouse path is
'file:/C:/Users/Ankith%20M/IdeaProjects/MorningBatchSBTProject/spark-warehouse'.
Spark Session Object created
18/05/29 22:20:05 ERROR Shell: Failed to locate the winutils binary in the hadoop
binary path
java.io.IOException: Could not locate executable null\bin\winutils.exe in the
Hadoop binaries.
	at org.apache.hadoop.util.Shell.getQualifiedBinPath(Shell.java:278)
	at org.apache.hadoop.util.Shell.getWinUtilsPath(Shell.java:300)
	at org.apache.hadoop.util.Shell.<clinit>(Shell.java:293)
	at org.apache.hadoop.util.StringUtils.<clinit>(StringUtils.java:76)
	at
org.apache.hadoop.mapred.FileInputFormat.setInputPaths(FileInputFormat.java:362)
	at
org.apache.spark.SparkContext$$anonfun$hadoopFile$1$$anonfun$30.apply(SparkContext.
scala:1014)
	at
org.apache.spark.SparkContext$$anonfun$hadoopFile$1$$anonfun$30.apply(SparkContext.
scala:1014)
	at
org.apache.spark.rdd.HadoopRDD$$anonfun$getJobConf$6.apply(HadoopRDD.scala:179)
	at
org.apache.spark.rdd.HadoopRDD$$anonfun$getJobConf$6.apply(HadoopRDD.scala:179)
	at scala.Option.foreach(Option.scala:257)
	at org.apache.spark.rdd.HadoopRDD.getJobConf(HadoopRDD.scala:179)
	at org.apache.spark.rdd.HadoopRDD.getPartitions(HadoopRDD.scala:198)
	at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:252)
	at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:250)
	at scala.Option.getOrElse(Option.scala:121)
	at org.apache.spark.rdd.RDD.partitions(RDD.scala:250)
	at
org.apache.spark.rdd.MapPartitionsRDD.getPartitions(MapPartitionsRDD.scala:35)
	at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:252)
	at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:250)
	at scala.Option.getOrElse(Option.scala:121)
	at org.apache.spark.rdd.RDD.partitions(RDD.scala:250)
	at
org.apache.spark.rdd.MapPartitionsRDD.getPartitions(MapPartitionsRDD.scala:35)
	at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:252)
	at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:250)
	at scala.Option.getOrElse(Option.scala:121)
	at org.apache.spark.rdd.RDD.partitions(RDD.scala:250)
	at org.apache.spark.SparkContext.runJob(SparkContext.scala:1958)
	at org.apache.spark.rdd.RDD$$anonfun$foreach$1.apply(RDD.scala:917)
	at org.apache.spark.rdd.RDD$$anonfun$foreach$1.apply(RDD.scala:915)
	at
org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
	at
org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
	at org.apache.spark.rdd.RDD.withScope(RDD.scala:362)
	at org.apache.spark.rdd.RDD.foreach(RDD.scala:915)
	at StudentDetailsAnalysis$.main(StudentDetailsAnalysis.scala:27)
	at StudentDetailsAnalysis.main(StudentDetailsAnalysis.scala)
(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))
Total No. of records: 22
the distinct number of subjects present in the entire school are:
(maths,6)
(history,8)
(science,8)
Students with name Mathew and marks 55 are:
((Mathew,55),2)
Students per Grade are:
(grade-3,4)
(grade-1,9)
(grade-2,9)
the average of each student:
((Lisa,grade-1),24.0)
((Mark,grade-2),17.5)
((Lisa,grade-2),61.0)
((Andrew,grade-2),77.0)
((Mathew,grade-3),45.0)
((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)
the average score of students in each subject across all grades are:
((Lisa,history),92.0)
((Mark,maths),57.5)
((Andrew,science),35.0)
((Mark,science),44.0)
((Mathew,science),55.0)
((Andrew,maths),28.5)
((Mathew,science),45.0)
((Mathew,history),71.0)
((John,maths),54.5)
((John,history),40.5)
((Lisa,science),24.0)
((Andrew,history),75.5)
the average score of students in each subject per grade are:
((history,grade-2),79.25)
((history,grade-3),86.0)
((maths,grade-1),46.0)
((science,grade-3),38.333333333333336)
((science,grade-1),50.0)
((science,grade-2),30.333333333333332)
((history,grade-1),51.666666666666664)
((maths,grade-2),48.5)
No. of students having average score greater than 50 in grade-2 are: 4
students having average score greater than 50 in grade-2 are:
((Lisa,grade-2),61.0)
((Andrew,grade-2),77.0)
```



```
((John,grade-2),74.0)
((Mathew,grade-2),65.66666666666667)
Average score per student_name across all grades are:
(Mark,50.75)
(Andrew,46.333333333333336)
(Mathew,65.66666666666667)
(Mathew,45.0)
(John,47.5)
(Lisa,58.0)
average score per student_name per grades are:
(Mark,50.75)
(Andrew,46.333333333333336)
(Mathew,65.66666666666667)
(Mathew,45.0)
(John,47.5)
(Lisa,58.0)
So the final list of eligible candidates are:
Mathew,45.0
Lisa,58.0
John,47.5
Andrew,46.333333333333336
Mathew,65.66666666666667
Mark,50.75

Process finished with exit code 0
```