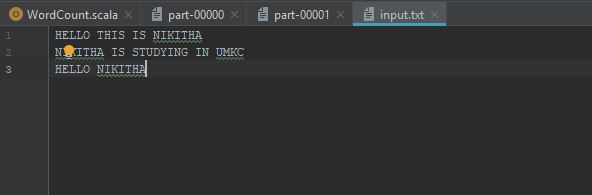
**Problem Statement 1:** **spark program using text data as the input and program should have at least Two Spark Transformations and Two Spark Actions.**

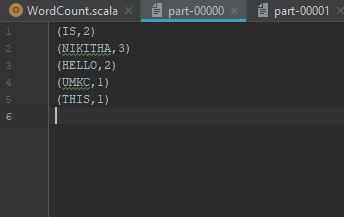
**WordCount:** **Scala Code:**

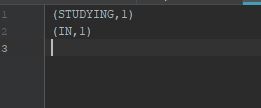
**import** org.apache.spark.\_  
  
**object** WordCount {  
 **def** main(args: Array[String]) {  
  
 System.*setProperty*("hadoop.home.dir","C:\\Users\\nikit\\OneDrive\\Desktop\\winutils" )  
 //val inputFile = args(0)  
 //val outputFile = args(1)  
 **val** conf = **new** SparkConf().setAppName("wordCount").setMaster("local[\*]")  
 // Create a Scala Spark Context.  
 **val** sc = **new** SparkContext(conf)  
 // Load our input data.  
 //val input = sc.textFile(inputFile)  
 **val** input = sc.textFile("input.txt")  
 // Split up into words.  
 **val** words = input.flatMap(line => line.split(" "))  
 // Transform into word and count.  
 **val** counts = words.map(word => (word, 1)).reduceByKey{**case** (x, y) => x + y}  
 // Save the word count back out to a text file, causing evaluation.  
 counts.saveAsTextFile("output")  
 }  
}

**Input:**



**Output:**





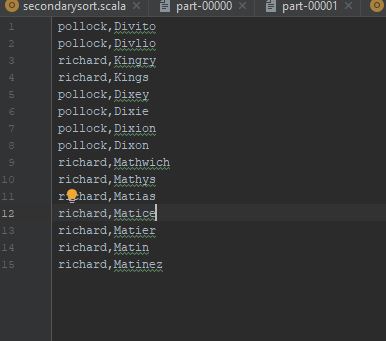
**Problem Statement 2:**

**Secondary Sorting in Map Reduce**

**Scala Code:**

**import** org.apache.spark.{ SparkConf, SparkContext }  
**import** org.apache.spark.sql.\_  
**import** org.apache.spark.sql.SQLContext.\_  
  
**object** secondarysort {  
 **def** main(args: Array[String]) {  
 System.*setProperty*("hadoop.home.dir", "C:\\Users\\nikit\\OneDrive\\Desktop\\winutils")  
 **val** conf = **new** SparkConf().setAppName("secondarysort").setMaster("local[\*]")  
 **val** sc = **new** SparkContext(conf)  
  
 **val** personRDD = sc.textFile("input.txt")  
 **val** pairsRDD = personRDD.map(\_.split(",")).map { k => (k(0), k(1)) }  
  
 **val** numReducers = 2;  
  
 **val** listRDD = pairsRDD.groupByKey(numReducers).mapValues(\_.toList.sortBy(r => r))  
  
  
 **val** resultRDD = listRDD.flatMap {  
 **case** (label, list) => {  
 list.map((label, \_))  
 }  
 }  
 resultRDD.saveAsTextFile(path= "output1")  
 }  
}

**Input:**



**Output:**

