CSEE5590 Big Data Programming

In Class Programming –8 Report
(Jongkook Son)

Project Overview:

Apache Spark is a unified analytics engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing.

Requirements/Task(s):

1. Spark Programming:

Write a spark program with an interesting use case using text data as the input and program should have at least Two Spark Transformations and Two Spark Actions.

Present your use case in map reduce paradigm as shown below (for word count).

2. Secondary Sorting in Map Reduce

Secondary sorting is used to sort the values in the reducer phase.

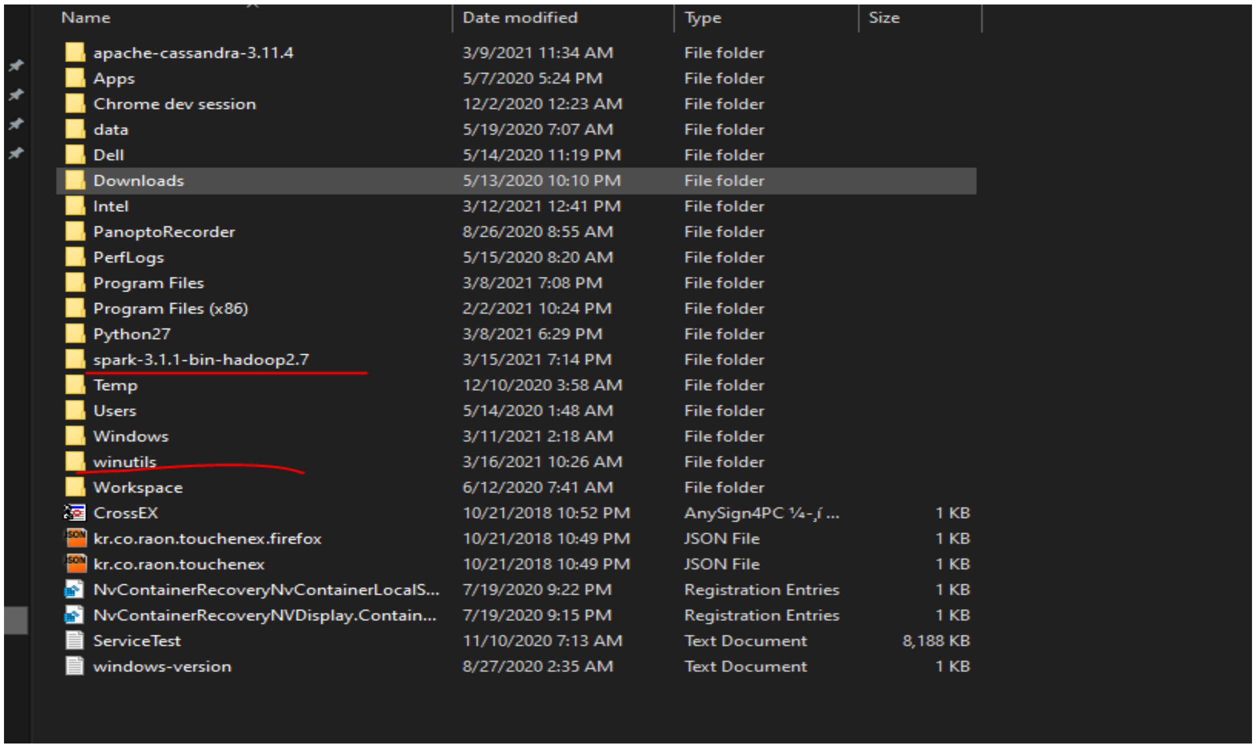
Take any input of your interest and perform secondary sorting on it.

What I learned in ICP:

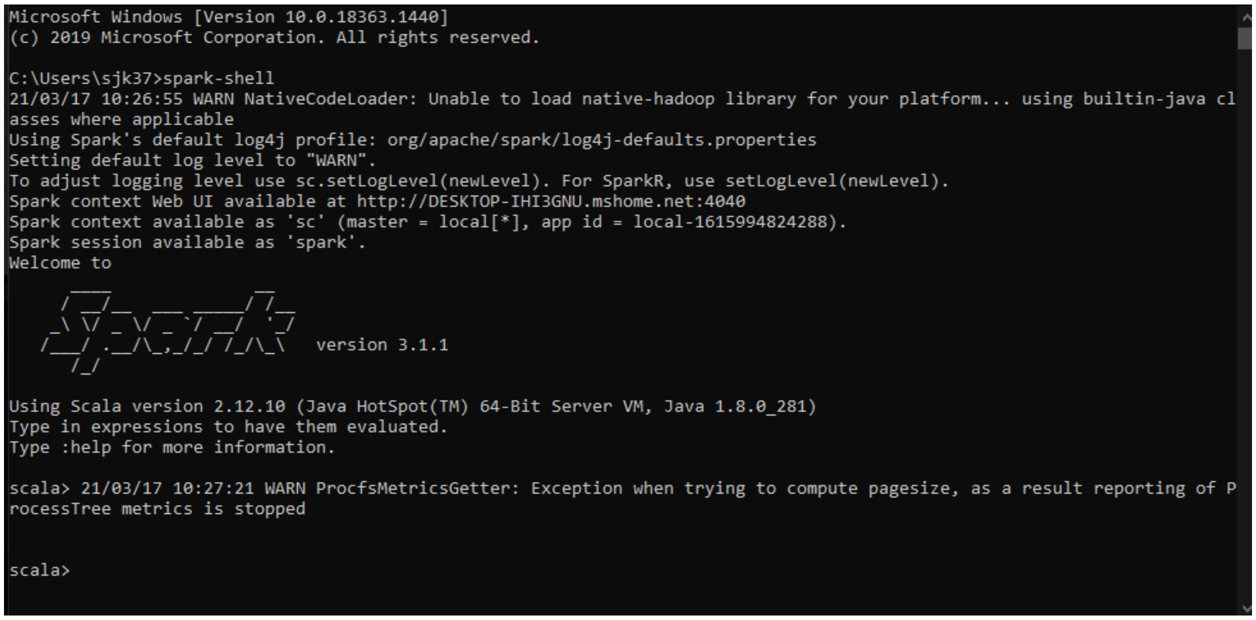
I could have learned how to set up Spark on my local machine and installing scala as a plugin for IntelliJ. It was a little tricky to install spark in my local but I did it by following steps. By fulfilling the first task, I could have learned some transformations and actions in spark and how can I implement it with scala. It seems better than I did on Hadoop. By Fulfilling the second task, I could have understood what secondary sort in map-reduce. The secondary sorting technique will enable us to sort the values (in ascending or descending order) passed to each reducer. The goal of the Secondary Sort pattern is to give some order to the values received by a reducer.

INSTALLATION OF Spark AND Scala

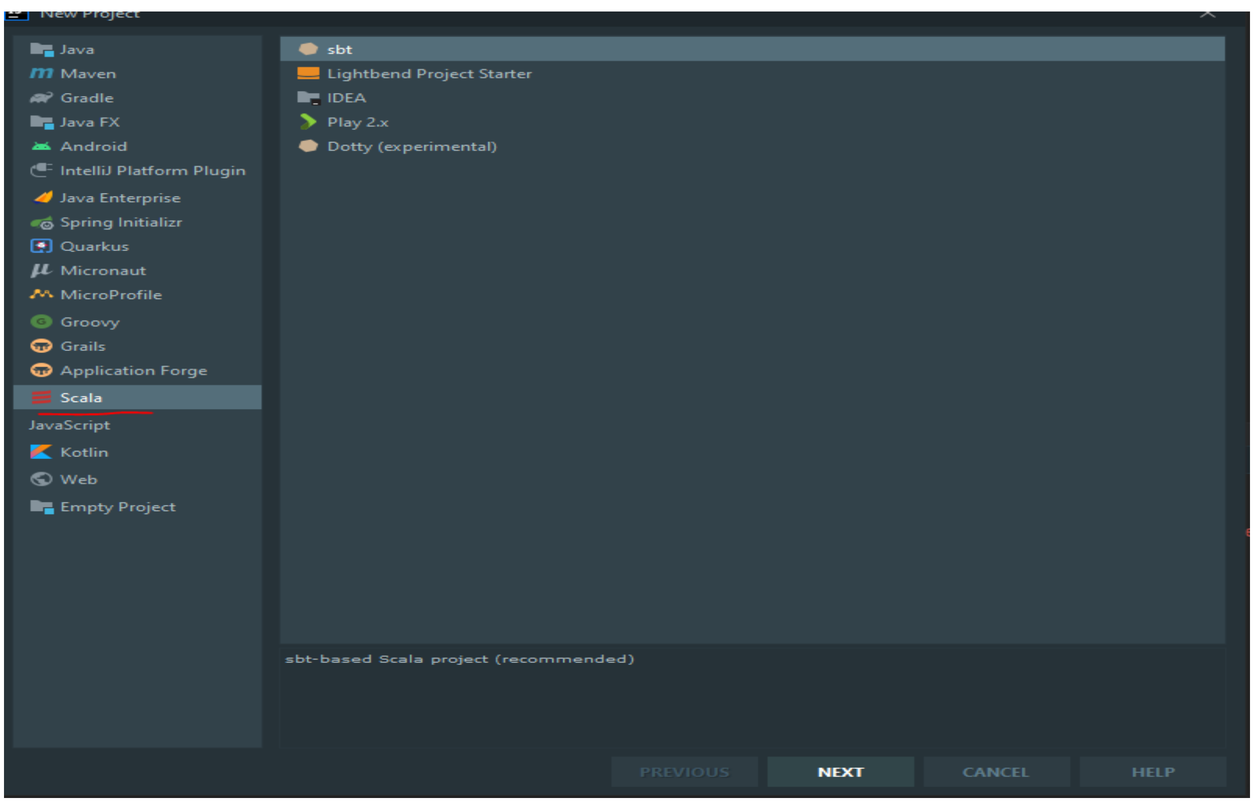
(I installed it on my local maschine and intellij.)



Name	Date modified	Type	Size
apache-cassandra-3.11.4	3/9/2021 11:34 AM	File folder	
Apps	5/7/2020 5:24 PM	File folder	
Chrome dev session	12/2/2020 12:23 AM	File folder	
data	5/19/2020 7:07 AM	File folder	
Dell	5/14/2020 11:19 PM	File folder	
Downloads	5/13/2020 10:10 PM	File folder	
Intel	3/12/2021 12:41 PM	File folder	
PanoptoRecorder	8/26/2020 8:55 AM	File folder	
PerfLogs	5/15/2020 8:20 AM	File folder	
Program Files	3/8/2021 7:08 PM	File folder	
Program Files (x86)	2/2/2021 10:24 PM	File folder	
Python27	3/8/2021 6:29 PM	File folder	
spark-3.1.1-bin-hadoop2.7	3/15/2021 7:14 PM	File folder	
Temp	12/10/2020 3:58 AM	File folder	
Users	5/14/2020 1:48 AM	File folder	
Windows	3/11/2021 2:18 AM	File folder	
winutils	3/16/2021 10:26 AM	File folder	
Workspace	6/12/2020 7:41 AM	File folder	
CrossEX	10/21/2018 10:52 PM	AnySign4PC ¼-í ...	1 KB
kr.co.raon.touchenex.firefox	10/21/2018 10:49 PM	JSON File	1 KB
kr.co.raon.touchenex	10/21/2018 10:49 PM	JSON File	1 KB
NvContainerRecoveryNvContainerLocalS...	7/19/2020 9:22 PM	Registration Entries	1 KB
NvContainerRecoveryNVDisplay.Contain...	7/19/2020 9:15 PM	Registration Entries	1 KB
ServiceTest	11/10/2020 7:13 AM	Text Document	8,188 KB
windows-version	8/27/2020 2:35 AM	Text Document	1 KB



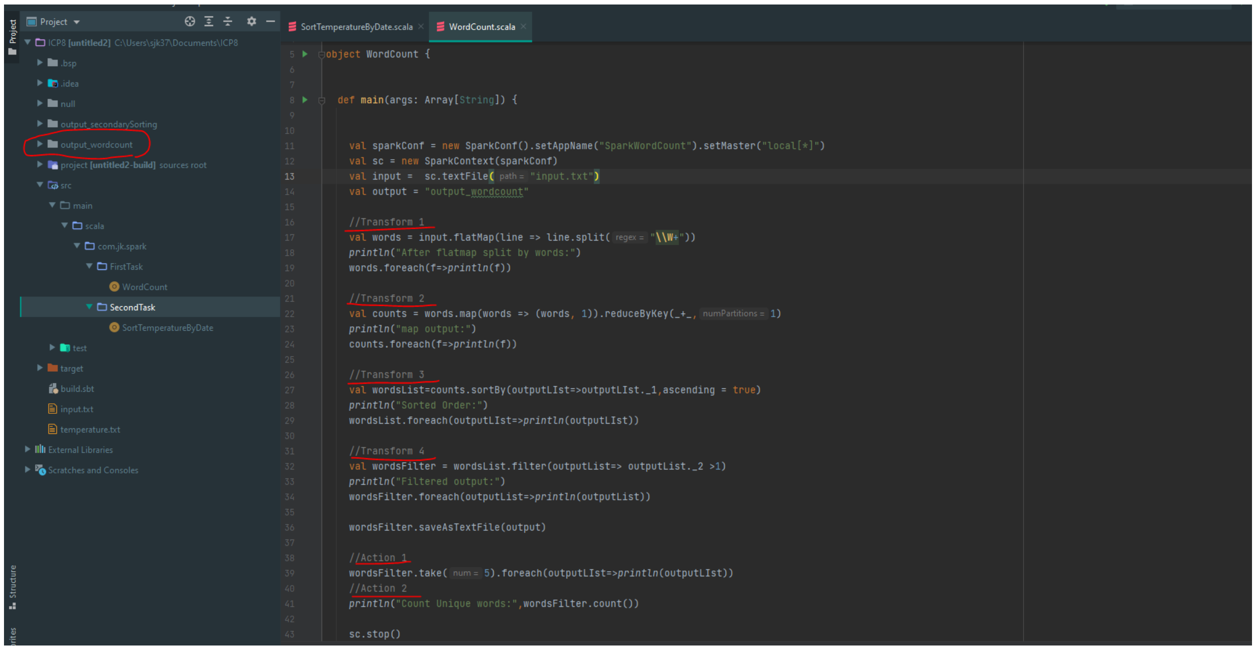
```
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\sjk37>spark-shell
21/03/17 10:26:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://DESKTOP-IHI3GNU.mshome.net:4040
Spark context available as 'sc' (master = local[*], app id = local-1615994824288).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_281)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 21/03/17 10:27:21 WARN ProcfsMetricsGetter: Exception when trying to compute pagesize, as a result reporting of P
rocessTree metrics is stopped


scala>
```



sbt-based Scala project (recommended)

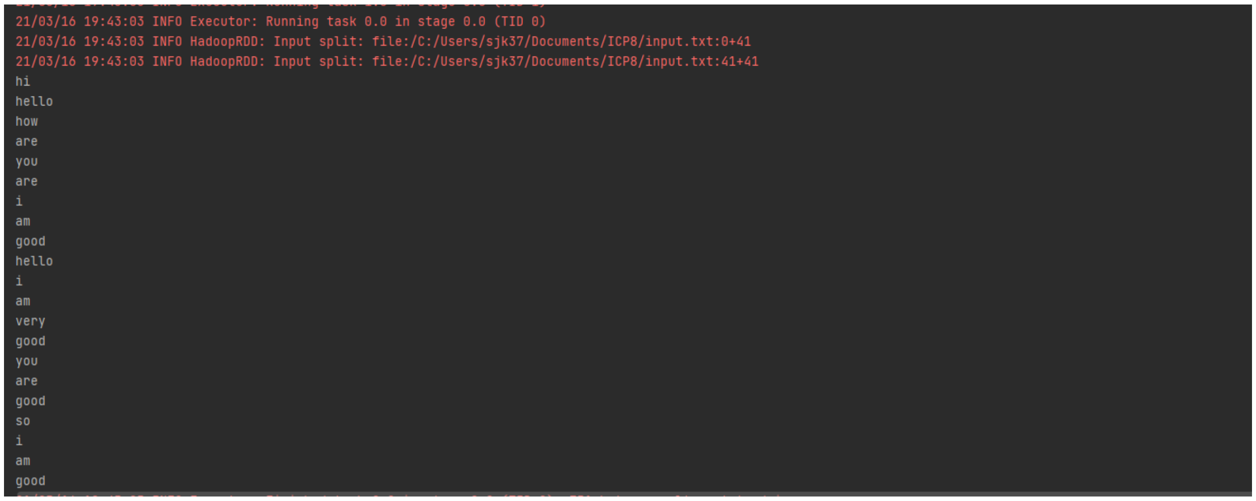
<Task1> Write a spark program with an interesting use case using text data as the input and program should have at least Two Spark Transformations and Two Spark Actions.

=> Here I used transformations map,flatmap,reducebykey,filter and in actions I used count, foreach, take.



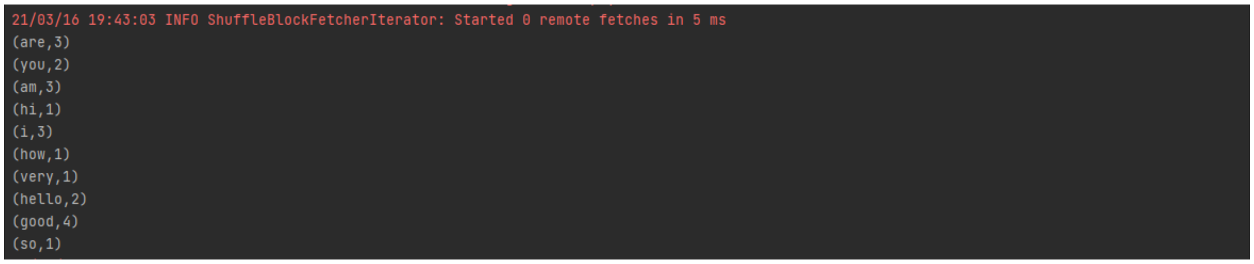
```scala
object WordCount {


  def main(args: Array[String]) {


    val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
    val sc = new SparkContext(sparkConf)
    val input =  sc.textFile( path = "input.txt")
    val output = "output_wordcount"

    //Transform 1
    val words = input.flatMap(line => line.split( regex = "\\W+"))
    println("After flatmap split by words:")
    words.foreach(f=>println(f))

    //Transform 2
    val counts = words.map(words => (words, 1)).reduceByKey(_+_, numPartitions = 1)
    println("map output:")
    counts.foreach(f=>println(f))

    //Transform 3
    val wordsList=counts.sortBy(outputLIst=>outputLIst._1,ascending = true)
    println("Sorted Order:")
    wordsList.foreach(outputLIst=>println(outputLIst))

    //Transform 4
    val wordsFilter = wordsList.filter(outputList=> outputList._2 >1)
    println("Filtered output:")
    wordsFilter.foreach(outputList=>println(outputList))

    wordsFilter.saveAsTextFile(output)

    //Action 1
    wordsFilter.take( num = 5).foreach(outputLIst=>println(outputLIst))
    //Action 2
    println("Count Unique words:",wordsFilter.count())

    sc.stop()
```

<Overall Code>



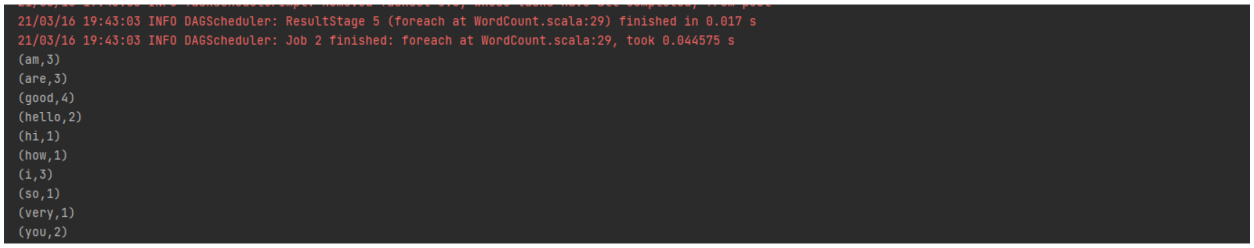
```
21/03/16 19:43:03 INFO Executor: Running task 0.0 in stage 0.0 (TID 0)
21/03/16 19:43:03 INFO HadoopRDD: Input split: file:/C:/Users/sjk37/Documents/ICP8/input.txt:0+41
21/03/16 19:43:03 INFO HadoopRDD: Input split: file:/C:/Users/sjk37/Documents/ICP8/input.txt:41+41
hi
hello
how
are
you
are
i
am
good
hello
i
am
very
good
you
are
good
so
i
am
good
```

<Transformation1> FlatMap



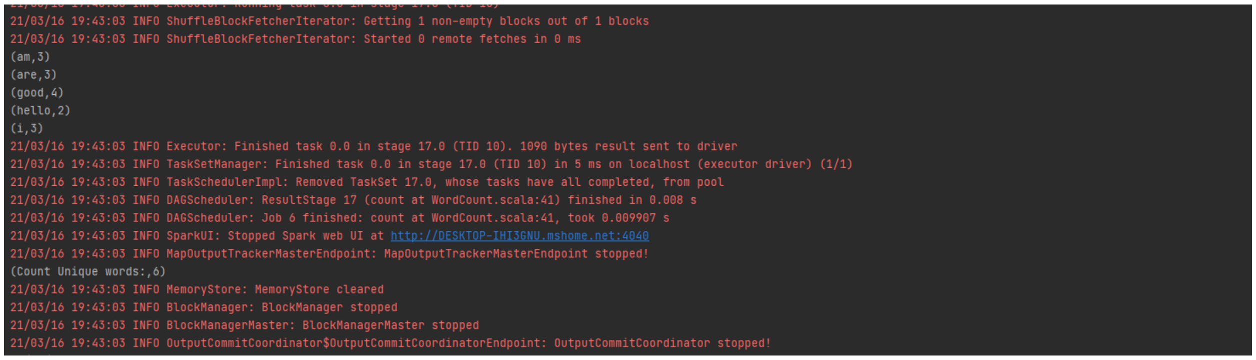
```
21/03/16 19:43:03 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 5 ms
(are,3)
(you,2)
(am,3)
(hi,1)
(i,3)
(how,1)
(very,1)
(hello,2)
(good,4)
(so,1)
```

<Transformation2> Map and Reduce



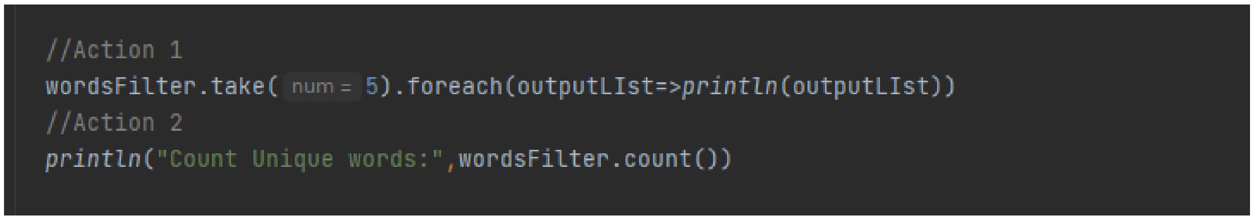
```
21/03/16 19:43:03 INFO DAGScheduler: ResultStage 5 (foreach at WordCount.scala:29) finished in 0.017 s
21/03/16 19:43:03 INFO DAGScheduler: Job 2 finished: foreach at WordCount.scala:29, took 0.044575 s
(am,3)
(are,3)
(good,4)
(hello,2)
(hi,1)
(how,1)
(i,3)
(so,1)
(very,1)
(you,2)
```

<Transformation3> Sort



```
21/03/16 19:43:03 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
21/03/16 19:43:03 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
(am,3)
(are,3)
(good,4)
(hello,2)
(i,3)
21/03/16 19:43:03 INFO Executor: Finished task 0.0 in stage 17.0 (TID 10). 1090 bytes result sent to driver
21/03/16 19:43:03 INFO TaskSetManager: Finished task 0.0 in stage 17.0 (TID 10) in 5 ms on localhost (executor driver) (1/1)
21/03/16 19:43:03 INFO TaskSchedulerImpl: Removed TaskSet 17.0, whose tasks have all completed, from pool
21/03/16 19:43:03 INFO DAGScheduler: ResultStage 17 (count at WordCount.scala:41) finished in 0.008 s
21/03/16 19:43:03 INFO DAGScheduler: Job 6 finished: count at WordCount.scala:41, took 0.009907 s
21/03/16 19:43:03 INFO SparkUI: Stopped Spark web UI at http://DESKTOP-IHI3GNU.mshome.net:4040
21/03/16 19:43:03 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
(Count Unique words:,6)
21/03/16 19:43:03 INFO MemoryStore: MemoryStore cleared
21/03/16 19:43:03 INFO BlockManager: BlockManager stopped
21/03/16 19:43:03 INFO BlockManagerMaster: BlockManagerMaster stopped
21/03/16 19:43:03 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
```

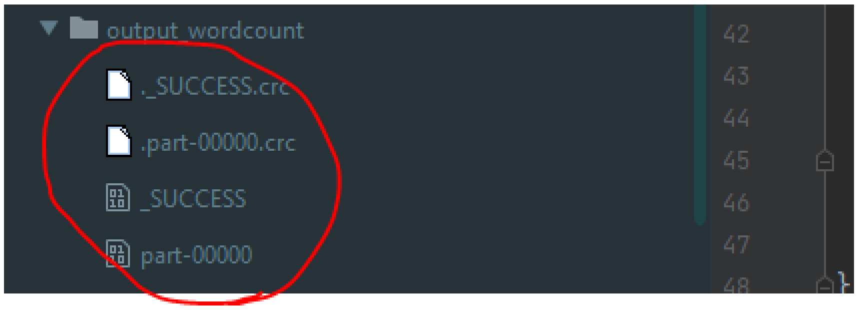
<Transformation4> Fliter



```
//Action 1
wordsFilter.take( num = 5).foreach(outputLIst=>println(outputLIst))
//Action 2
println("Count Unique words:",wordsFilter.count())
```

<Actions>

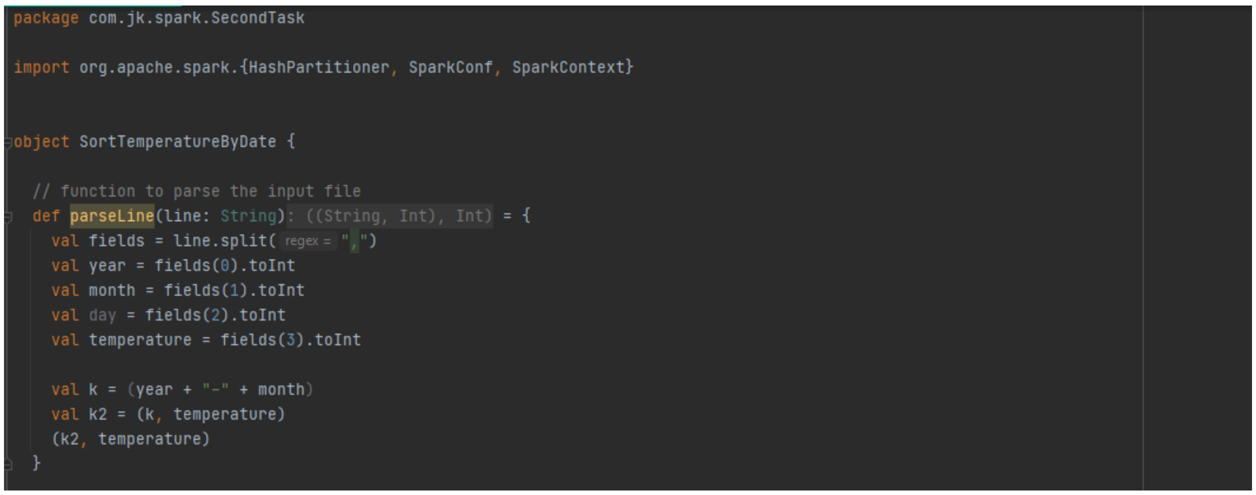
=>I take 5 out for the filtered list and count the number of words filter



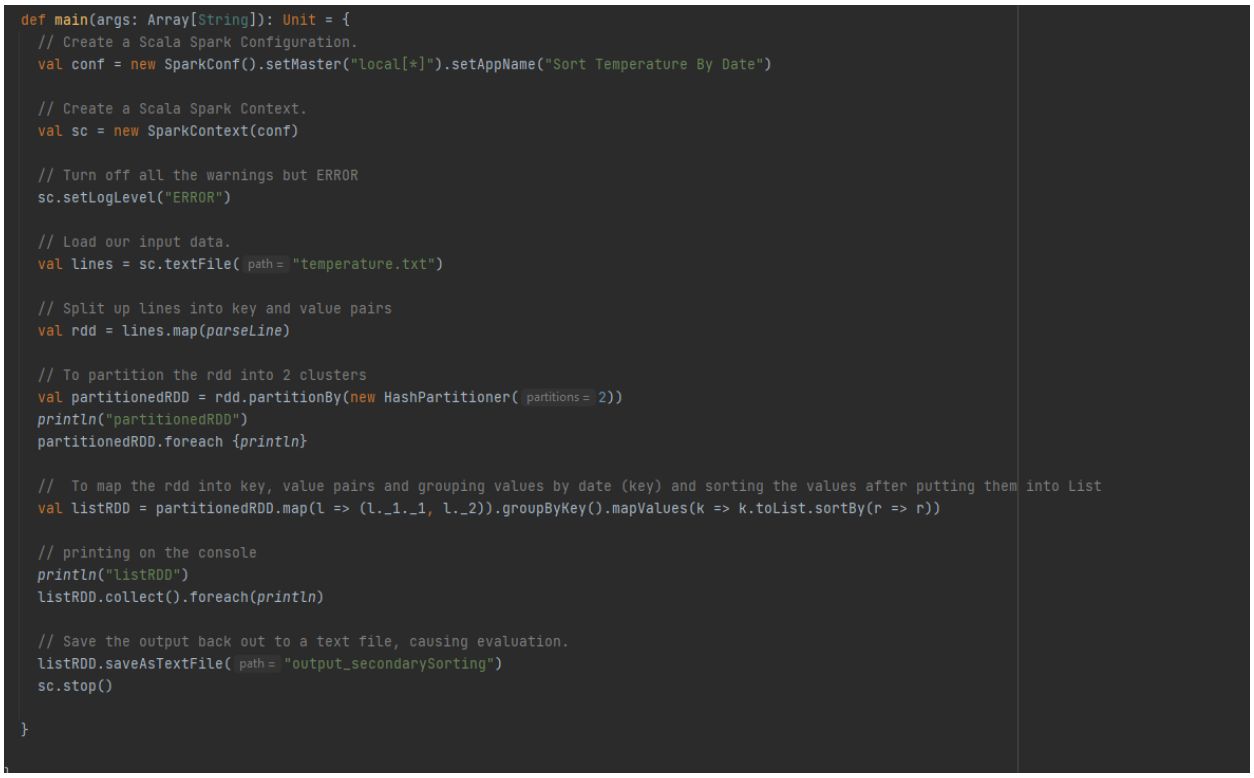
```
output_wordcount
    ._SUCCESS.crc
    .part-00000.crc
    _SUCCESS
    part-00000
```

<Output of the Task>

<Task2>Performing the partitioning of the map values which will do group by with comparing the common values as a KEY and the tuple values will be taken as VAlUES

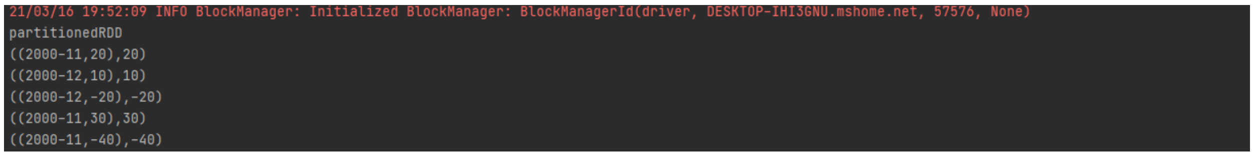


```
package com.jk.spark.SecondTask

import org.apache.spark.{HashPartitioner, SparkConf, SparkContext}


object SortTemperatureByDate {

  // function to parse the input file
  def parseLine(line: String): ((String, Int), Int) = {
    val fields = line.split( regex = ",")
    val year = fields(0).toInt
    val month = fields(1).toInt
    val day = fields(2).toInt
    val temperature = fields(3).toInt

    val k = (year + "-" + month)
    val k2 = (k, temperature)
    (k2, temperature)
  }
```



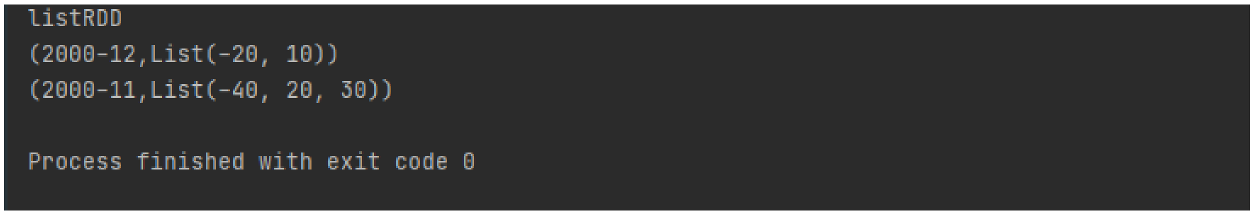
```scala
def main(args: Array[String]): Unit = {
  // Create a Scala Spark Configuration.
  val conf = new SparkConf().setMaster("local[*]").setAppName("Sort Temperature By Date")

  // Create a Scala Spark Context.
  val sc = new SparkContext(conf)

  // Turn off all the warnings but ERROR
  sc.setLogLevel("ERROR")

  // Load our input data.
  val lines = sc.textFile( path = "temperature.txt")

  // Split up lines into key and value pairs
  val rdd = lines.map(parseLine)

  // To partition the rdd into 2 clusters
  val partitionedRDD = rdd.partitionBy(new HashPartitioner( partitions = 2))
  println("partitionedRDD")
  partitionedRDD.foreach {println}

  //  To map the rdd into key, value pairs and grouping values by date (key) and sorting the values after putting them into List
  val listRDD = partitionedRDD.map(l => (l._1._1, l._2)).groupByKey().mapValues(k => k.toList.sortBy(r => r))

  // printing on the console
  println("listRDD")
  listRDD.collect().foreach(println)

  // Save the output back out to a text file, causing evaluation.
  listRDD.saveAsTextFile( path = "output_secondarySorting")
  sc.stop()

}
```

<Overall Code>



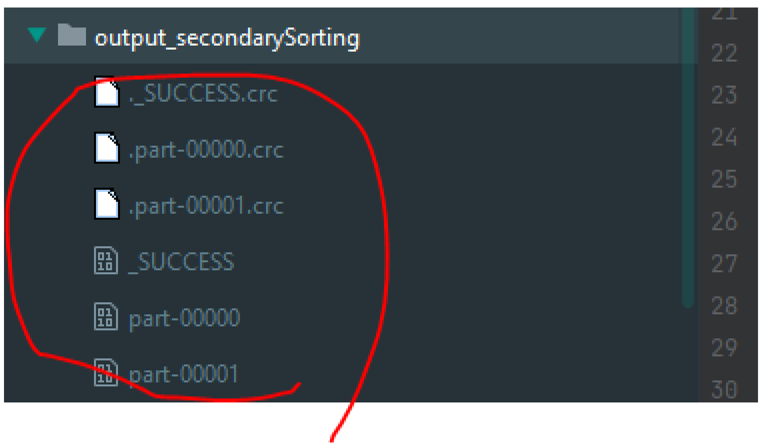
```
21/03/16 19:52:09 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, DESKTOP-IHI3GNU.mshome.net, 57576, None)
partitionedRDD
((2000-11,20),20)
((2000-12,10),10)
((2000-12,-20),-20)
((2000-11,30),30)
((2000-11,-40),-40)
```

<Partitioned Rdd>



```
listRDD
(2000-12,List(-20, 10))
(2000-11,List(-40, 20, 30))

Process finished with exit code 0
```

<List RDD >



output_secondarySorting
._SUCCESS.crc
.part-00000.crc
.part-00001.crc
_SUCCESS
part-00000
part-00001

<Output for Task2>