**For reference:**

Spark Installation Steps

<https://www.dataneb.com/post/installing-apache-spark-and-scala-windows>

Software to install spark

<https://drive.google.com/drive/folders/1BP9Zs5uh_LJHa5x6eDCObqFQ3ayyj3gC?usp=sharing>

**Prog1:**

**Print Hello Word at scala shell.**

$scala

scala> println("Hello World!");

Hello World!

Prog2:

1. Execute Hello World Program in SCALA IDE. Follow the steps given in

<https://www.dataneb.com/post/hello-world-with-scala-ide>

Prog3:

**Program to run wordcount on scala shell**

**Note- Create a textfile sparkdata.txt locally and give appropriate path while loading the data using sc.textFile**

val data=sc.textFile("sparkdata.txt")

data.collect;

val splitdata = data.flatMap(line =**>** line.split(" "));

splitdata.collect;

val mapdata = splitdata.map(word =**>** (word,1));

mapdata.collect;

val reducedata = mapdata.reduceByKey(\_+\_);

reducedata.collect;

Prog 5. Execute any four transformations and four actions by following the commands given in

<https://www.dataneb.com/post/spark-rdd-transformations-and-actions-example>

2. Execute wordcount program on scala shell and scala IDE

**Program to run wordcount on scala IDE**

**Note: Follow the Helloworld program steps to create scala object and type in the following program and execute**

**package** **wordcount**

**import** **org.apache.spark.SparkConf**

**import** **org.apache.spark.SparkContext**

**import** **org.apache.spark.rdd.RDD.rddToPairRDDFunctions**

**object** **WordCount** {

**def**

main(args**:** **Array**[**String**]) **=** {

//Start the Spark context

**val**

conf **=** **new** **SparkConf**()

.setAppName("WordCount")

.setMaster("local")

**val**

sc **=** **new** **SparkContext**(conf)

//Read some example file to a test RDD

**val**

test **=**

sc.textFile("input.txt")

test.flatMap {

line **=>** //for

each line

line.split(" ") //split

the line in word by word.

}

.map {

word **=>** //for

each word

(word, **1**) //Return

a key/value tuple, with the word as key and 1 as valuw

.reduceByKey(**\_** + **\_**) //Sum

all of the value with same key

.saveAsTextFile("output.txt") //Save

to a text file

//Stop the Spark context

sc.stop

}

}

**Prog 5:**

**Lab Program -**

**Using RDD and FlaMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.**

val textFile = sc.textFile("/home/bhoom/Desktop/wc.txt")

val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_ + \_)

import scala.collection.immutable.ListMap

val sorted=ListMap(counts.collect.sortWith(\_.\_2 > \_.\_2):\_\*)// sort in descending order based on values

println(sorted)

for((k,v)<-sorted)

{

if(v>4)

{

print(k+",")

print(v)

println()

}

}