SPARK ASSIGNMENT 17.1

Problem Statement

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.

Sample document :
This-is-my-first-assignment.
It-will-count-the-number-of-lines-in-this-document.
The-total-number-of-lines-is-3

//starting the HDFS

```
[acadgild@localhost sparkdata]$ start-dfs.sh
```

//creating and editing the input file at LOCAL filesystem

```
[acadgild@localhost sparkdata]$ gedit chhaya.txt
```

//Browsing the contents of text file

```
[acadgild@localhost sparkdata]$ cat chhaya.txt
```

//creation of user directory at HDFS

```
[acadgild@localhost sparkdata]$ hadoop fs -mkdir -p /user/acadgild/spark/
```

//Browsing through the HDFS user directory

```
[acadgild@localhost sparkdata]$ hadoop fs -ls /user/acadgild/spark/
```

//moving file from local filesystem to HDFS

```
[acadgild@localhost sparkdata]$ hadoop fs -copyFromLocal chhaya.txt
/user/acadgild/spark/
```

//Browsing through hdfs directory to verify the file

```
[acadgild@localhost sparkdata]$ hadoop fs -ls /user/acadgild/spark/
Found 1 items
-rw-r--r--   1 acadgild supergroup        192 2017-11-21 02:35
/user/acadgild/spark/chhaya.txt
```

//View the HDFS file

[acadgild@localhost sparkdata]$ hadoop fs -cat /user/acadgild/spark/chhaya.txt

// Initiating the spark shell prompt

acadgild@localhost sparkdata]$ spark-shell

//read the text file from hdfs via spark context object

val rdd1 = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/chhaya.txt",1)

//flattening the text file and splitting it linewise.

val rdd2 = rdd1.flatMap(lines => if (lines.equals("")) Array[String]() else lines.split("\n"))

//removing empty lines to get exact amount if the file contains empty lines.

val rdd3 = rdd2.filter(lines => !lines.equals(""))

//calculating the total count of non empty lines in document

val totallines = rdd3.count()

//display total nr of non empty lines

println(totallines)

Input file Screenshot:

```
scala> rdd1.foreach(println)
A-APPLE
B-BAT-BREATHTAKING-BAT
C-COW

D-DAY
E-EYES

F-FISH
G-GOOD
H-HONEY

I-IOTA
J-JUSTICE
K-KALE
L-LOVE-LAMBDA

M-MALIGNANT-MONKEY-MONEY-MILES
N-NAIL
O-OX

P-PALE
Q-QUIECE
R-RAIN-REAL-ROAMING-ROLL-RUN

T-TAME
U-USEFUL

W-WORLD
X-XMEN

Z-ZEN
```

Output_1 :

```
scala> val rdd1 = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/chhaya.txt",1)
rdd1: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[52] at textFile at <console>:27

scala> val rdd2 = rdd1.flatMap(lines => if (lines.equals("")) Array[String]() else lines.split("\n")
)
rdd2: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[53] at flatMap at <console>:29

scala> val rdd3 = rdd2.filter(lines => !lines.equals(""))
rdd3: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[54] at filter at <console>:31

scala> val totallines = rdd3.count()
totallines: Long = 23

scala> println(totallines)
23

scala>
```

```
//read the text file from hdfs via spark context object

val rdd1 = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/chhaya.txt",1)

//flattening the text file and splitting it space delimiter.

val rdd2 = rdd1.flatMap(lines => if (lines.equals("")) Array[String]()
else lines.split(" ")).count()

// print the number of words in the document separated by space delimiter.

println(rdd2)
```

Output_2 :

```
scala> val rdd1 = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/chhaya.txt",1)
rdd1: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[56] at textFile at <console>:27

scala> val rdd2 = rdd1.flatMap(lines => if (lines.equals("")) Array[String]() else lines.split(" "))
.count()
rdd2: Long = 23

scala> println(rdd2)
23

scala>
```

```
//read the text file from hdfs via spark context object

val rdd1 = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/chhaya.txt",1)

//flattening the text file and splitting it hyphen delimiter.

val rdd2 = rdd1.flatMap(lines => if (lines.equals("")) Array[String]()
 else lines.split("-")).count()

// print the number of words in the document separated by hyphen delimiter.

println(rdd2)
```

Output_3 :

```
scala> val rdd1 = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/chhaya.txt",1)
rdd1: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[59] at textFile at <console>:27

scala> val rdd2 = rdd1.flatMap(lines => if (lines.equals("")) Array[String]() else lines.split("-"))
.count()
rdd2: Long = 56

scala> println(rdd2)
56

scala>
```
