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

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

```
[acadgild@localhost ~]$ hadoop fs -ls /
17/11/10 02:43:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library
or your platform... using builtin-java classes where applicable
Found 3 items
-rw-r--r--   1 acadgild supergroup       1366 2016-04-28 18:53 /README.txt
-rw-r--r--   1 acadgild supergroup        114 2017-11-10 02:42 /sample.txt
drwx-wx-wx   - acadgild supergroup          0 2016-04-28 18:53 /tmp
[acadgild@localhost ~]$ hadoop fs -cat /sample.txt
17/11/10 02:43:34 WARN util.NativeCodeLoader: Unable to load native-hadoop library
or your platform... using builtin-java classes where applicable
This-is-my-first-assignment.
It-will-count-the-number-of-lines-in-this-document.
The-total-number-of-lines-is-3
```

scala>val rows =sc.textFile("sample.txt")
rows.count()

```
scala> val rows =sc.textFile("sample.txt")
rows: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[13] at textFile at <cons
e>:27

scala> rows.count()
res7: Long = 3
```

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

val rows =sc.textFile("*("/home/acadgild/user/*sample.txt")
var flat_map= rows.flatMap(row=>row.split(" "))
var map=flat_map.map(word=>(word,1))
var count=map.reduceByKey(_+_)
count.saveAsTextFile("/home/acadgild/usernew/word_count_output/")

```
scala> val rows =sc.textFile("/home/acadgild/user/sample.txt")
rows: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[15] at textFile at <cons
e>:27

scala> var flat_map= rows.flatMap(row=>row.split(" "))
flat_map: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[16] at flatMap at <c
sole>:29

scala> var map=flat_map.map(word=>(word,1))
map: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[17] at map at <con
le>:31

scala> var count=map.reduceByKey(_+_)
count: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[18] at reduceByKey at
console>:33

scala> count.saveAsTextFile("/home/acadgild/usernew/word_count_output/")
```

cd usernew
ls
cd *word_count_output*
ls
cat part-00000

```
[acadgild@localhost ~]$ cd usernew
[acadgild@localhost usernew]$ ls
word_count_output
[acadgild@localhost usernew]$ cd word_count_output
[acadgild@localhost word_count_output]$ ls
part-00000  _SUCCESS
[acadgild@localhost word_count_output]$ cat part-00000
(This-is-my-first-assignment.,1)
(The-total-number-of-lines-is-3,1)
(It-will-count-the-number-of-lines-in-this-document.,1)
```

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.

val inputFile= sc.textFile("/home/acadgild/user/sample.txt")
val words= inputFile.flatMap(line=>line.split("-"))
val wordCounts= words.map(word=>(word,1)).reduceByKey{case(x,y)=>x+y}
wordCounts.saveAsTextFile("/home/acadgild/usernew/word_count_outputNew/")

```
scala> val inputFile= sc.textFile("/home/acadgild/user/sample.txt")
inputFile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[23] at textFile at <c
onsole>:27

scala> val words= inputFile.flatMap(line=>line.split("-"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[24] at flatMap at <consol
e>:29

scala> val wordCounts= words.map(word=>(word,1)).reduceByKey{case(x,y)=>x+y}
wordCounts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[26] at reduceByKey
 at <console>:31

scala> wordCounts.saveAsTextFile("/home/acadgild/usernew/word_count_outputNew/")
```

cd usernew
ls
cd word_count_outputNew
ls
cat part-00000

```
[acadgild@localhost usernew]$ ls
word_count_output  word_count_outputNew
[acadgild@localhost usernew]$ cd word_count_outputNew
[acadgild@localhost word_count_outputNew]$ ls
part-00000  _SUCCESS
[acadgild@localhost word_count_outputNew]$ ls
part-00000  _SUCCESS
[acadgild@localhost word_count_outputNew]$ cat part-00000
(this,1)
(lines,2)
(The,1)
(is,2)
(document. ,1)
(number,2)
(assignment. ,1)
(will,1)
(This,1)
(in,1)
(first,1)
(3,1)
(total,1)
(of,2)
(It,1)
(my,1)
(count,1)
(the,1)
```