ADVANCED SPARK ASSIGNMENT 21.1

PROBLEM STATEMENT:

Count the number of blank lines in a text file, by using accumulators

Sample file :

Hello World

It's a sunny day

<blank_line>

When will it rain?

Will it rain today!

//moving to local user directory

[acadgild@localhost ~]$ cd Downloads/sparkdata/

//checking the files present in current directory

[acadgild@localhost sparkdata]$ ls

//putting input file at hdfs

[acadgild@localhost sparkdata]$ hadoop fs -put accumulatordemo.txt /user/acadgild/spark/

//Browsing the file contents at HDFS

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

Hello World

It's a sunny day

When will it rain?

Will it rain today!

Be careful with your words

and how do you want it to convey

cause once it is said

it can be forgiven but

it cannot be forgotten instead.

```
//Reading the HDFS File and putting the same into SPARK RDD
val file = sc.textFile("hdfs://localhost:9000//user/acadgild/spark/accumulatordemo.txt")
//Declaring the accumulator variables with the help of SPARK CONTEXT object
val blankLines = sc.accumulator(0)
val validLines = sc.accumulator(0)
//Defining function to calculate blank and non blank lines and setting a counter on accumulators
def countoperation(line:String):Array[String] = {
   if (line.equals("")) {
      blankLines += 1
      Array[String]()
    }
   else {
      validLines += 1
      line.split(" ")
```

```
  }
}
```

```
//map & reduce operation to invoke the user defined function countoperation

val wordRDD = file.flatMap(countoperation).map(x => (x,1)).reduceByKey((x,y) => x+y)

//Expected Output

println(blankLines)
```

Screenshots:

```
scala> val blankLines = sc.accumulator(0)
blankLines: org.apache.spark.Accumulator[Int] = 0

scala> val validLines = sc.accumulator(0)
validLines: org.apache.spark.Accumulator[Int] = 0

scala> def countoperation(line:String):Array[String] = {
     |     if (line.equals("")) {
     |         blankLines += 1
     |         Array[String]()
     |      }
     |     else {
     |         validLines += 1
     |         line.split(" ")
     |     }
     | }
countoperation: (line: String)Array[String]

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

```
scala> file.foreach(println)
Hello World
It's a sunny day

When will it rain?
Will it rain today!

Be careful with your words

and how do you want it to convey

cause once it is said

it can be forgiven but

it cannot be forgotten instead.

scala> val wordRDD = file.flatMap(countoperation).map(x => (x,1)).reduceByKey((x,y) => x+y)
wordRDD: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:35

scala> wordRDD.collect()
res1: Array[(String, Int)] = Array((it,6), (is,1), (convey,1), (you,1), (cannot,1), (Hello,1), (can,
1), (rain,1), (rain?,1), (a,1), (instead.,1), (be,2), (do,1), (forgotten,1), (how,1), (with,1), (Whe
n,1), (Will,1), (today!,1), (to,1), (said,1), (day,1), (will,1), (forgiven,1), (World,1), (once,1),
(careful,1), (words,1), (Be,1), (cause,1), (but,1), (and,1), (sunny,1), (want,1), (your,1), (It's,1)
)

scala> blanklines.collect()
<console>:26: error: not found: value blanklines
              blanklines.collect()
              ^

scala> println(blankLines)
6
```