Assignment-24

Task 1

Read a stream of Strings, fetch the words which can be converted to numbers. Filter out the rows, where the sum of numbers in that line is odd. Provide the sum of all the remaining numbers in that batch.

Below screenshot shows the spark application to filter the lines containing even numbers.

Even_Number_Line.scala | Offensive_Word_Count.scala

```scala
package com.scala

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

object Even_Number_Line{

  def main(args: Array[String]):Unit = {

      // create a function which
      def Get_Lines_Sum(input : String) : Double ={

          val line = input.split(" ")
              var number : Double = 0.0
              for (x <- line)
              {
                try{
                  val value = x.toDouble
                      number = number + value
                }
                catch
```

Even_Number_Line.scala | Offensive_Word_Count.scala

```scala
                {
                case ex : Exception => {}
                }
              }
          return number
      }

      println("Task1 of assignment 24")

      val conf = new SparkConf().setMaster("local[2]").setAppName("EvenNumberedL
      val sc = new SparkContext(conf)

      sc.setLogLevel("WARN")
      println("Spark Context Created")

      // Create a local StreamingContext with working thread and batch interval
      val ssc = new StreamingContext(sc, Seconds(20))
      println("Spark Streaming Context Created")

      // Create a DStream that will connect to hostname:port,localhost:9999
```

Even_Number_Line.scala | Offensive_Word_Count.scala

```scala
        println("Spark Streaming Context Created")

        // Create a DStream that will connect to hostname:port,localhost:9999
        val lines = ssc.socketTextStream( hostname="localhost", port= 9999)

        //filter the even string from input line by using Get_Lines function
        val lines_filter = lines.filter(x => Get_Lines_Sum(x)%2 == 0)
        //add all the numbers the even string from input line by using Get_Lines f
        val lines_sum = lines_filter.map(x => Get_Lines_Sum(x))
        println("Lines with even sum:")
        lines_filter.print()

        println("Sum of the numbers in even lines:")
        lines_sum.reduce(_+_).print()
        // Start the computation
        ssc.start()
        // Wait for the computation to terminate
        ssc.awaitTermination()
    }
}
```

Before starting the above application, we need start **'netcat'** to provide the inputs.

```
File  Edit  View  Search  Terminal  Tabs  Help
acadgild...  acadgild...  acadgild...  acadgil
[acadgild@localhost ~]$ nc -lk 9999
```

After the starting the application, input has been provided as shown in the below screenshot.

```
File  Edit  View  Search  Terminal  Tabs  Help
acadgild...  acadgild...  acadgild...  acadg
[acadgild@localhost ~]$ nc -lk 9999
Hello234 Everyone24
How35 are27 you23
manage45 all56
do24 like45 this22
```

In the below screenshot, we are able to see that lines containing sum of odd numbers are filtered and even number is displayed and the sum of the number is displayed in the next line.

```
Spark Context Created
Spark Streaming Context Created
Lines with even sum:
Sum of the numbers in even lines:

-------------------------------------------
Time: 1528791620000 ms
-------------------------------------------
How35 are27 you23
manage45 all56

-------------------------------------------
Time: 1528791620000 ms
-------------------------------------------
42.0

-------------------------------------------
Time: 1528791640000 ms
-------------------------------------------

-------------------------------------------
Time: 1528791640000 ms
-------------------------------------------
```

Task 2

Read two streams

1. List of strings input by user

2. Real-time set of offensive words

Find the word count of the offensive words inputted by the user as per the real-time set of offensive words

Below screenshot shows the spark application to filter the offensive words from input string entered by user.

```scala
package com.scala

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

object Offensive_Word_Count {
  def main(args: Array[String]):Unit = {
    println("This is the task2 of assignment session 26")

  val conf = new SparkConf().setMaster("local[2]").setAppName("SparkStreamingExample"
  val sc = new SparkContext(conf)
  sc.setLogLevel("WARN")
  println("Spark Context Created")

  //create a set of offensive words which we use to compare and filter these words fr
    val offensive_word_list: Set[String] = Set("Hello", "BDHS", "Hi", "Spark")
  //print the list of these offensive words
  println(s"$offensive_word_list")
  // Create a local StreamingContext with working thread and batch interval of 20 sec
  val ssc = new StreamingContext(sc, Seconds(20))

```

In the above screenshot, we have set of words that we considered as offensive words.

"Hello","BDHS","Hi","Spark"

```scala
  println(s"$offensive_word_list")
  // Create a local StreamingContext with working thread and batch interval of 20 sec
  val ssc = new StreamingContext(sc, Seconds(20))

  println("Spark Streaming Context Created !")
  // Create a DStream that will connect to hostname:port,localhost:9999
  val lines = ssc.socketTextStream( hostname="localhost", port= 9999)
    // Split each line into words
    val words = lines.flatMap(_.split(" ")).map (x => x)
    //words.print()
    // filter the offensive words from input string by using set and count words
  val Offensive_Word_Count = words.filter(x => offensive_word_list.contains(x)).map(x
  Offensive_Word_Count.print()
    // Start the computation
  ssc.start()
  // Wait for the computation to terminate
  ssc.awaitTermination()
  } }
```

'netcat' is started using the command shown in the below screenshot and input has been provided.

```
acadgild@localhost:~
[acadgild@localhost ~]$ nc -lk 9999
Hello Barath
Hi BDHS Acadgild
```

```
18/12/09 20:22:37 INFO BlockManager: Initialized
Spark Context Created
Set(Hello, BDHS, Hi, Spark)
Spark Streaming Context Created !
18/12/09 20:22:51 WARN ReceiverSupervisorImpl: R
```

```
18/12/09 20:22:59 WARN BlockManager: Block input-0-1544367178800 repl
-------------------------------------------
Time: 1544367180000 ms
-------------------------------------------
(Hello,1)

18/12/09 20:23:11 WARN RandomBlockReplicationPolicy: Expecting 1 repl
18/12/09 20:23:11 WARN BlockManager: Block input-0-1544367191600 repl
-------------------------------------------
Time: 1544367200000 ms
-------------------------------------------
(Hi,1)
(BDHS,1)

-------------------------------------------
Time: 1544367220000 ms
-------------------------------------------
```

In the above screenshot, we are able to see that spark application has count the number of offensive words occur as per the input string provided by the user.