Assignment 20.1

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.

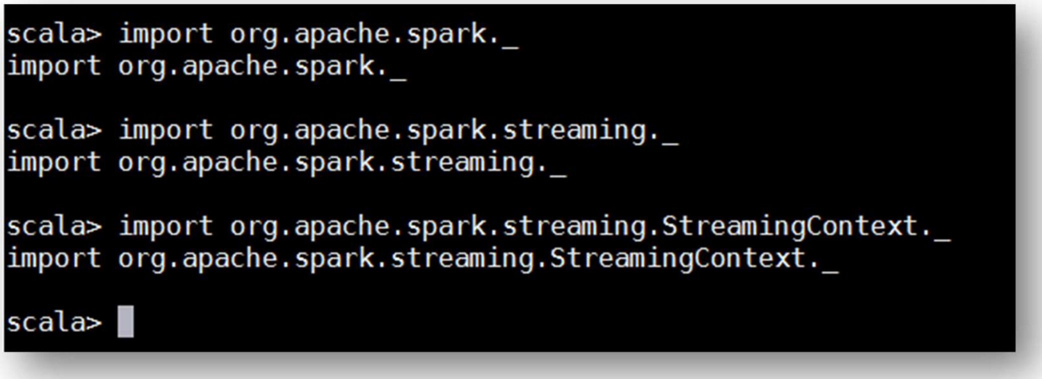
Provide the sum of all the remaining numbers in that batch.

1. Start the spark shell with 4 threads by the command : spark-shell –master local[4] and import the following packages:

import org.apache.spark._

import org.apache.spark.streaming._

mport org.apache.spark.streaming.StreamingContext._

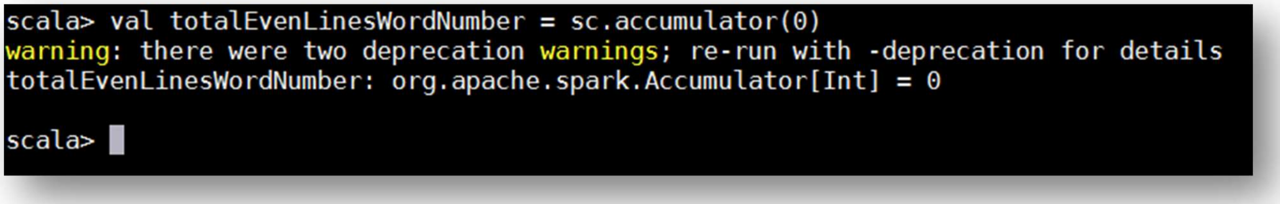


```
scala> import org.apache.spark._
import org.apache.spark._

scala> import org.apache.spark.streaming._
import org.apache.spark.streaming._

scala> import org.apache.spark.streaming.StreamingContext._
import org.apache.spark.streaming.StreamingContext._

scala> 
```

2. Declare accumulator totalEvenLinesWordNumber which will keep track of sum of number of word numbers in lines so far

val totalEvenLinesWordNumber = sc.accumulator(0)

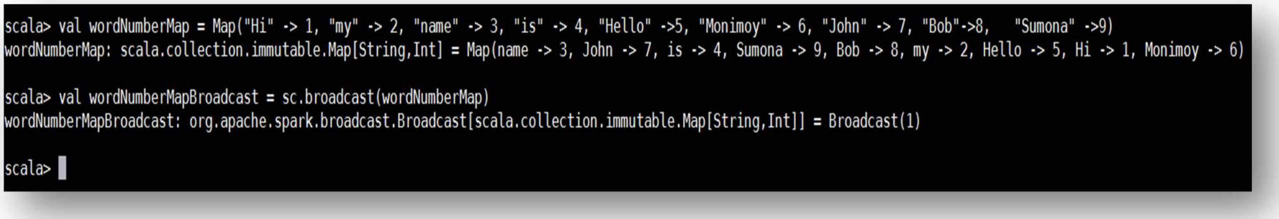
}



```
scala> val totalEvenLinesWordNumber = sc.accumulator(0)
warning: there were two deprecation warnings; re-run with -deprecation for details
totalEvenLinesWordNumber: org.apache.spark.Accumulator[Int] = 0

scala> 
```

3. Define a map for converting word to number. If word is not there in map then 0 will be returned. Broadcast the map. Code is as below

```
val wordNumberMap = Map("Hi" -> 1, "my" -> 2, "name" -> 3, "is" -> 4, "Hello" ->5,
"Monimoy" -> 6, "John" -> 7, "Bob"->8,      "Vibhu" ->9)

val wordNumberMapBroadcast = sc.broadcast(wordNumberMap)
```

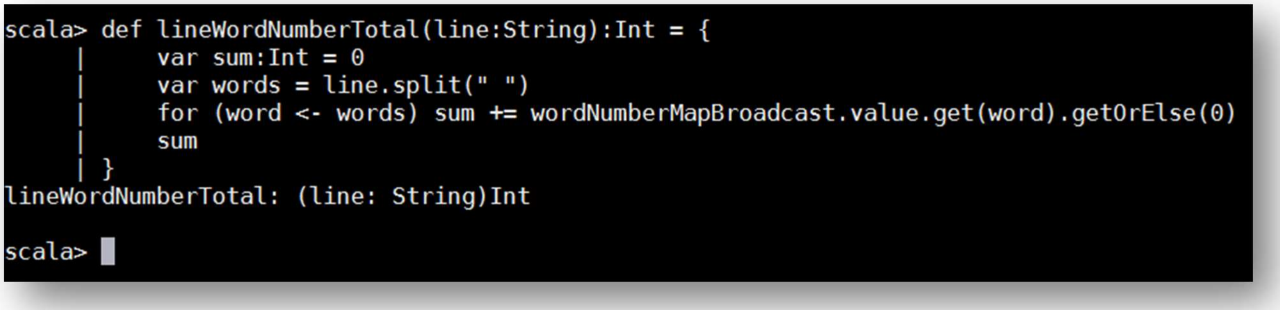


```
scala> val wordNumberMap = Map("Hi" -> 1, "my" -> 2, "name" -> 3, "is" -> 4, "Hello" ->5, "Monimoy" -> 6, "John" -> 7, "Bob"->8,   "Sumona" ->9)
wordNumberMap: scala.collection.immutable.Map[String,Int] = Map(name -> 3, John -> 7, is -> 4, Sumona -> 9, Bob -> 8, my -> 2, Hello -> 5, Hi -> 1, Monimoy -> 6)

scala> val wordNumberMapBroadcast = sc.broadcast(wordNumberMap)
wordNumberMapBroadcast: org.apache.spark.broadcast.Broadcast[scala.collection.immutable.Map[String,Int]] = Broadcast(1)

scala> 
```

4. Define a function lineWordNumberTotal which will split a line based on blank space to get all the words in a next. Next in the lookup wordNumberMapBroadcast , based on word, corresponding number is retrieved and sum all these numbers together.

Code is as below:

```
def lineWordNumberTotal(line:String):Int = {

var sum:Int = 0

 var words = line.split(" ")

for (word <- words) sum += wordNumberMapBroadcast.value.get(word).getOrElse(0)

sum

}
```

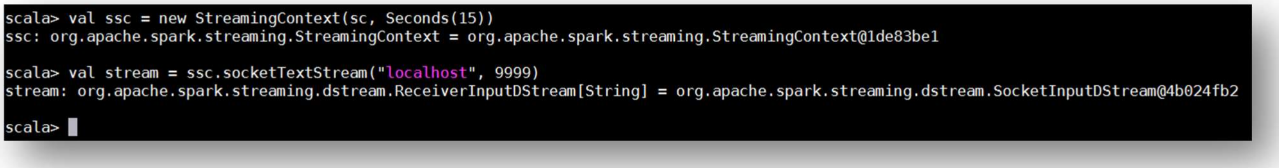


```
scala> def lineWordNumberTotal(line:String):Int = {
     |     var sum:Int = 0
     |     var words = line.split(" ")
     |     for (word <- words) sum += wordNumberMapBroadcast.value.get(word).getOrElse(0)
     |     sum
     | }
lineWordNumberTotal: (line: String)Int

scala> 
```

5. Start text streaming on localhost with port number 9999 and interval 15 seconds and return the stream. Code is as below:

val ssc = new StreamingContext(sc, Seconds(15))

val stream = ssc.socketTextStream("localhost", 9999)

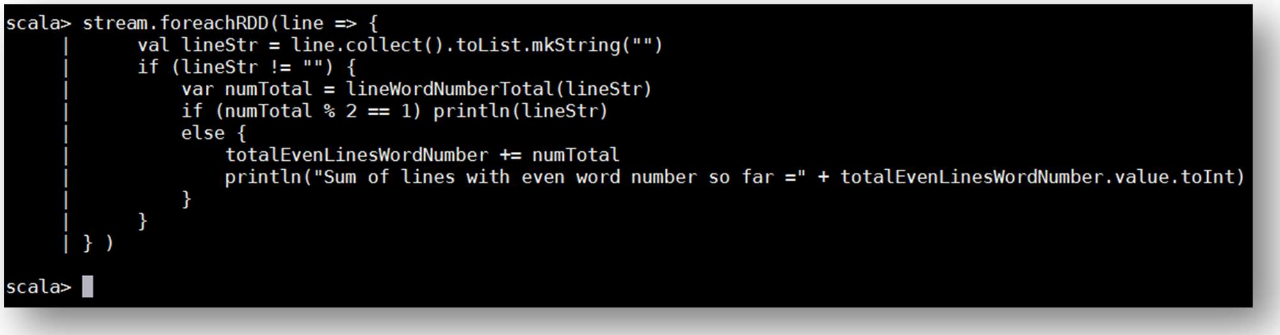


```
scala> val ssc = new StreamingContext(sc, Seconds(15))
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@1de83be1

scala> val stream = ssc.socketTextStream("localhost", 9999)
stream: org.apache.spark.streaming.dstream.ReceiverInputDStream[String] = org.apache.spark.streaming.dstream.SocketInputDStream@4b024fb2

scala> 
```

6. Process each RDD in stream. First convert the RDD to string. If it is not blank calculate word number for each word and sum them using function lineWordNumberTotal and put to variable numTotal. If numTotal is odd, print the corresponding line. Also, add numTotal to accumulator accu totalEvenLinesWordNumber and print the sum

```
stream.foreachRDD(line => {

val lineStr = line.collect().toList.mkString("")

if (lineStr != "") {

  var numTotal = lineWordNumberTotal(lineStr)

  if (numTotal % 2 == 1) println(lineStr)

   else {

          totalEvenLinesWordNumber += numTotal
```

println("Sum of lines with even word number so far =" + totalEvenLinesWordNumber.value.toInt) }}}



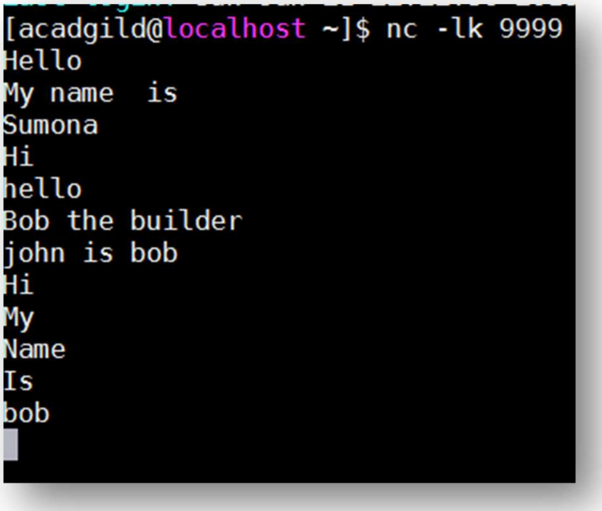
```
scala> stream.foreachRDD(line => {
     |      val lineStr = line.collect().toList.mkString("")
     |      if (lineStr != "") {
     |          var numTotal = lineWordNumberTotal(lineStr)
     |          if (numTotal % 2 == 1) println(lineStr)
     |          else {
     |              totalEvenLinesWordNumber += numTotal
     |              println("Sum of lines with even word number so far =" + totalEvenLinesWordNumber.value.toInt)
     |          }
     |      }
     | } )

scala> 
```

7. Start the streams and wait till its termination. Code is as below:

ssc.start()

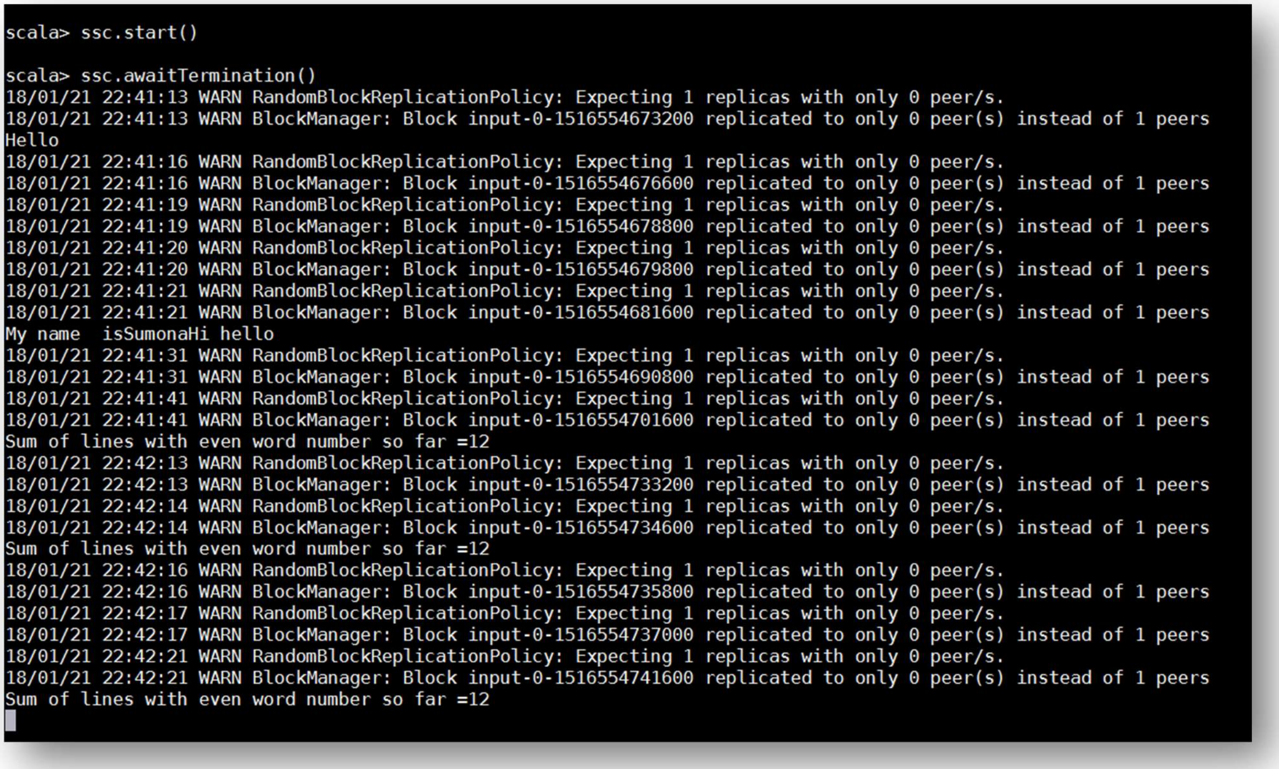
ssc.awaitTermination()

We will enter few string values in a new terminal **nc –lk 9999** and check the output accordingly



```
[acadgild@localhost ~]$ nc -lk 9999
Hello
My name  is
Sumona
Hi
hello
Bob the builder
john is bob
Hi
My
Name
Is
bob
```

Output:



```
scala> ssc.start()

scala> ssc.awaitTermination()
18/01/21 22:41:13 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:41:13 WARN BlockManager: Block input-0-1516554673200 replicated to only 0 peer(s) instead of 1 peers
Hello
18/01/21 22:41:16 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:41:16 WARN BlockManager: Block input-0-1516554676600 replicated to only 0 peer(s) instead of 1 peers
18/01/21 22:41:19 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:41:19 WARN BlockManager: Block input-0-1516554678800 replicated to only 0 peer(s) instead of 1 peers
18/01/21 22:41:20 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:41:20 WARN BlockManager: Block input-0-1516554679800 replicated to only 0 peer(s) instead of 1 peers
18/01/21 22:41:21 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:41:21 WARN BlockManager: Block input-0-1516554681600 replicated to only 0 peer(s) instead of 1 peers
My name  isSumonaHi hello
18/01/21 22:41:31 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:41:31 WARN BlockManager: Block input-0-1516554690800 replicated to only 0 peer(s) instead of 1 peers
18/01/21 22:41:41 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:41:41 WARN BlockManager: Block input-0-1516554701600 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even word number so far =12
18/01/21 22:42:13 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:42:13 WARN BlockManager: Block input-0-1516554733200 replicated to only 0 peer(s) instead of 1 peers
18/01/21 22:42:14 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:42:14 WARN BlockManager: Block input-0-1516554734600 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even word number so far =12
18/01/21 22:42:16 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:42:16 WARN BlockManager: Block input-0-1516554735800 replicated to only 0 peer(s) instead of 1 peers
18/01/21 22:42:17 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:42:17 WARN BlockManager: Block input-0-1516554737000 replicated to only 0 peer(s) instead of 1 peers
18/01/21 22:42:21 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/01/21 22:42:21 WARN BlockManager: Block input-0-1516554741600 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even word number so far =12
```