ASSIGNMENT 20.1

Problem Statement:

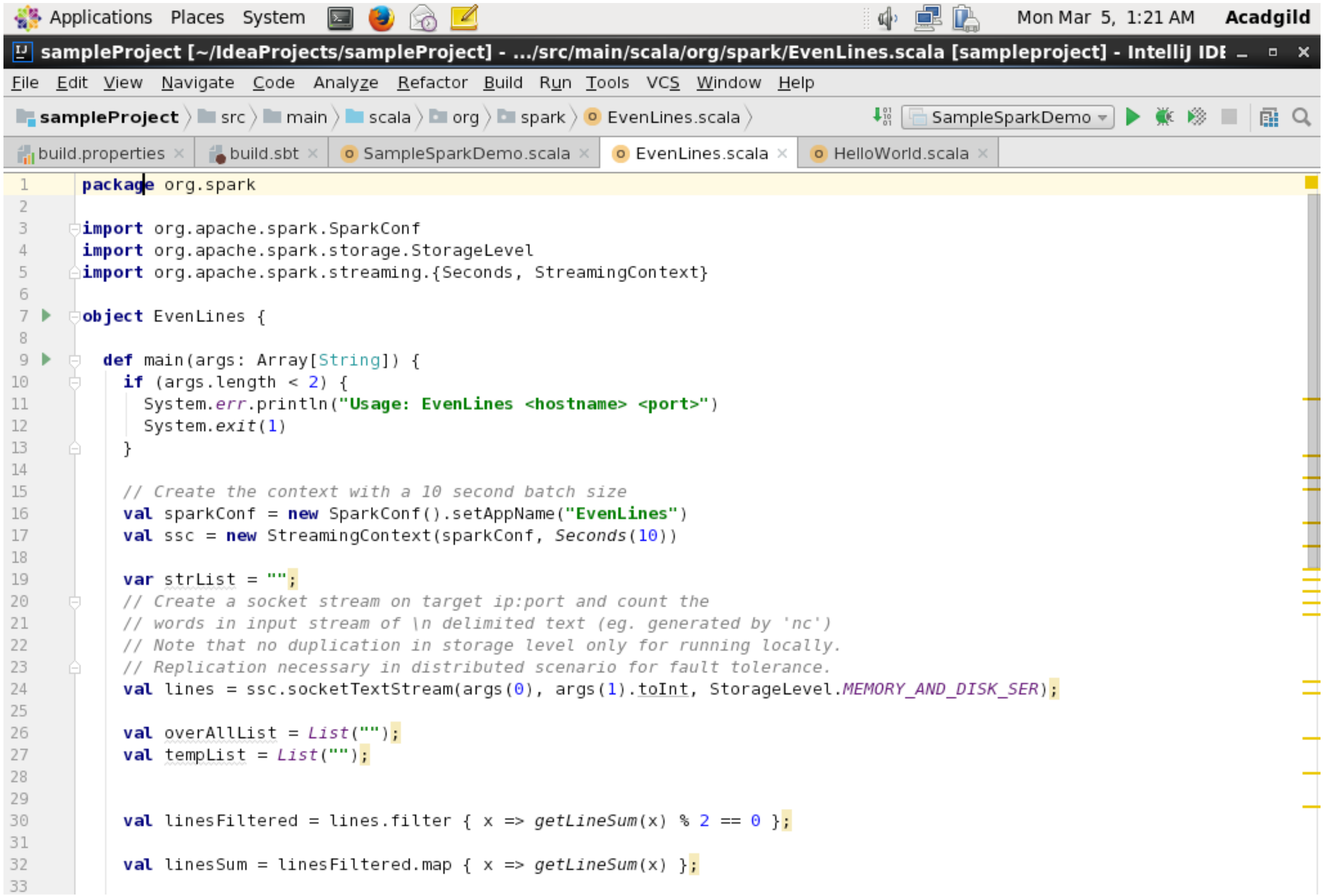
Read a stream of Strings, fetch the words which can be converted to numbers. Filter out the rows, where the sum of numbers in that line is odd.
Provide the sum of all the remaining numbers in that batch.

Solution:

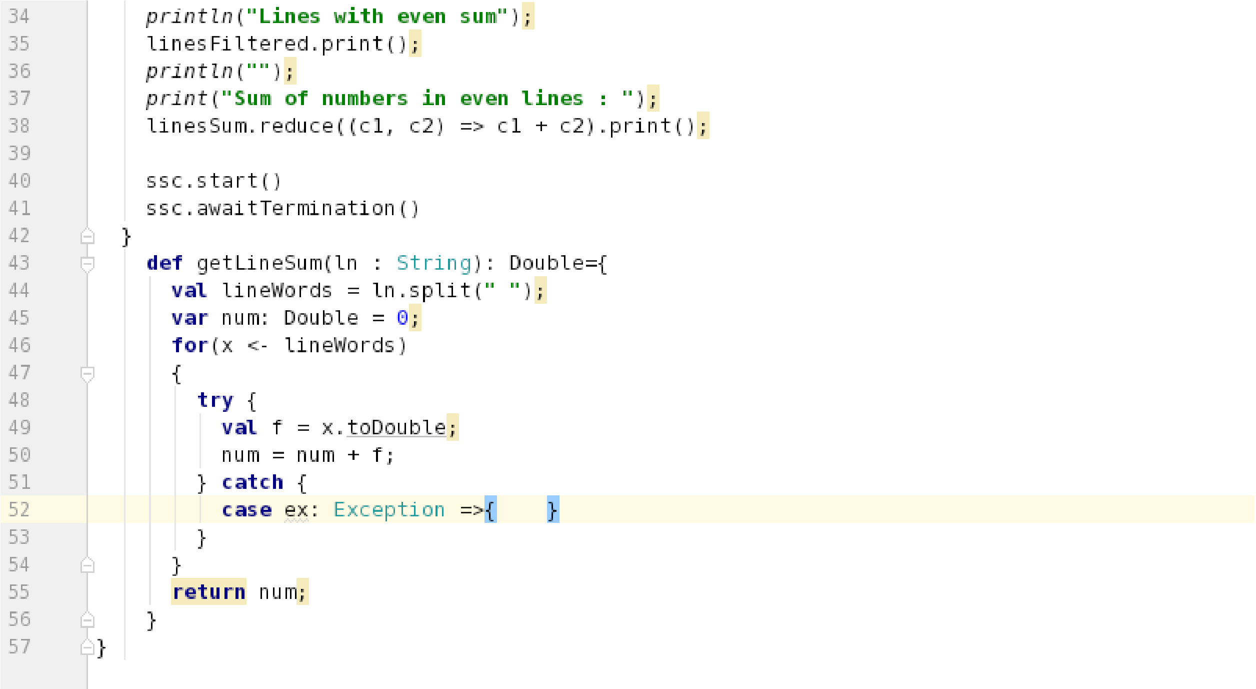
Here is the Spark code in Scala to create a stream of strings over socket connection:

EvenLines.scala:

```
package org.spark

// import required Spark packages

import org.apache.spark.SparkConf
import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.{Seconds, StreamingContext}


/*
 * Counts words in UTF8 encoded, '\n' delimited text received from the network every second.
 * Usage: EvenLines <hostname> <port>
 * <hostname> and <port> describe the TCP server that Spark Streaming would connect to receive data.
 * To run this on your local machine, you need to first run a Netcat server with this command:
 *    `$ nc -lk 9999`
 * and then run the example
 *   `$ bin/run-example org.apache.spark.examples.streaming.NetworkWordCount localhost 9999`
 */
object EvenLines {
  def main(args: Array[String]) {
// check whether user has passed sufficient command line arguments for execution of the program
    if (args.length < 2) {
      System.err.println("Usage: EvenLines <hostname> <port>")
      System.exit(1)
    }

    // create a streaming context with a 10 second batch size
    val sparkConf = new SparkConf().setAppName("EvenLines")
    val ssc = new StreamingContext(sparkConf, Seconds(10))

    // Create a socket stream on target ip:port and count the
    // words in input stream of \n delimited text
    // Note that no duplication in storage level only for running locally.
    // Replication necessary in distributed scenario for fault tolerance.
    val lines = ssc.socketTextStream(args(0), args(1).toInt, StorageLevel. MEMORY_AND_
DISK_SER)

   // retain only those lines in the current batch in which the sum of all numbers is even
    val linesFiltered = lines.filter { x => getLineSum(x)%2==0 }

     // compute sum for the retained lines with even sum
    val linesSum = linesFiltered.map { x => getLineSum(x) }

    // print the resulting lines followed by their sum
    println("Lines with even sum")
    linesFiltered.print()
    println("")
    print("Sum of numbers in even lines : ")
    linesSum.reduce( (c1, c2) => c1 + c2).print()

    // start the streaming operation until the program gets terminated
    ssc.start()
    ssc.awaitTermination()
   }

   // a routine to compute sum of numbers in a line passed by user over a socket connection
   def getLineSum(ln : String): Double={
    val lineWords = ln.split(" ")
    var sum: Double = 0
    for(x <- lineWords)
    {
      try {
        val f = x.toDouble
        sum = sum + f
     } catch {
        case ex: Exception =>{    }
     }
    }
   sum                                   // return sum of numbers present in a line
   }
}
```


Applications Places System
Mon Mar 5, 1:21 AM Acadgild
sampleProject [~/IdeaProjects/sampleProject] - .../src/main/scala/org/spark/EvenLines.scala [sampleproject] - IntelliJ IDE
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
sampleProject src main scala org spark EvenLines.scala
SampleSparkDemo
build.properties
build.sbt
SampleSparkDemo.scala
EvenLines.scala
HelloWorld.scala

```
package org.spark

import org.apache.spark.SparkConf
import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.{Seconds, StreamingContext}

object EvenLines {

  def main(args: Array[String]) {
    if (args.length < 2) {
      System.err.println("Usage: EvenLines <hostname> <port>")
      System.exit(1)
    }

    // Create the context with a 10 second batch size
    val sparkConf = new SparkConf().setAppName("EvenLines")
    val ssc = new StreamingContext(sparkConf, Seconds(10))

    var strList = "";
    // Create a socket stream on target ip:port and count the
    // words in input stream of \n delimited text (eg. generated by 'nc')
    // Note that no duplication in storage level only for running locally.
    // Replication necessary in distributed scenario for fault tolerance.
    val lines = ssc.socketTextStream(args(0), args(1).toInt, StorageLevel.MEMORY_AND_DISK_SER);

    val overAllList = List("");
    val tempList = List("");


    val linesFiltered = lines.filter { x => getLineSum(x) % 2 == 0 };

    val linesSum = linesFiltered.map { x => getLineSum(x) };

```

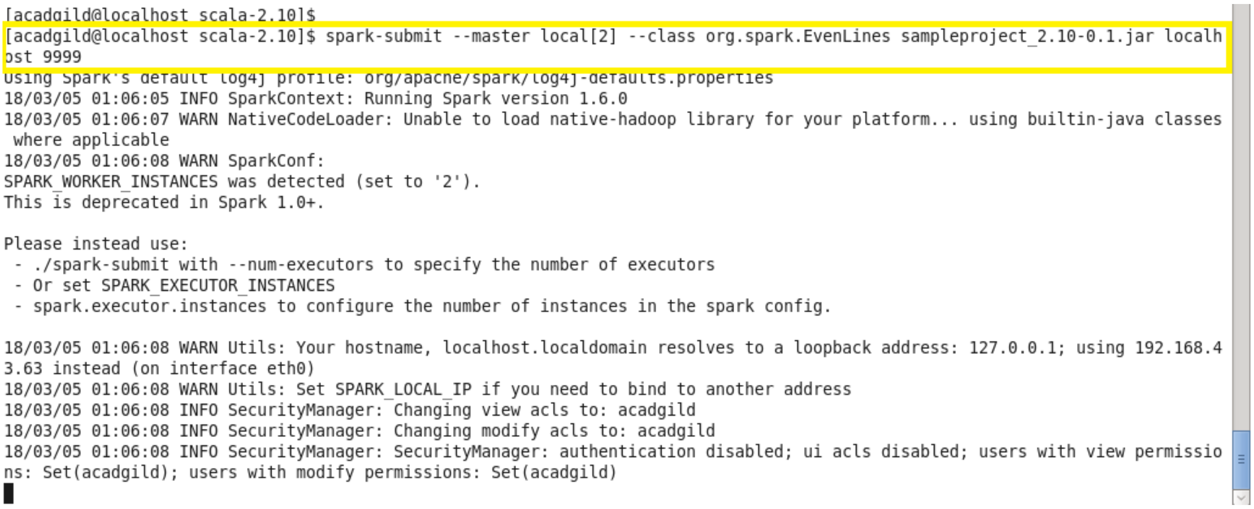


```
    println("Lines with even sum");
    linesFiltered.print();
    println("");
    print("Sum of numbers in even lines : ");
    linesSum.reduce((c1, c2) => c1 + c2).print();

    ssc.start()
    ssc.awaitTermination()
  }
    def getLineSum(ln : String): Double={
      val lineWords = ln.split(" ");
      var num: Double = 0;
      for(x <- lineWords)
      {
        try {
          val f = x.toDouble;
          num = num + f;
        } catch {
          case ex: Exception =>{     }
        }
      }
      return num;
    }
}
```

Output:

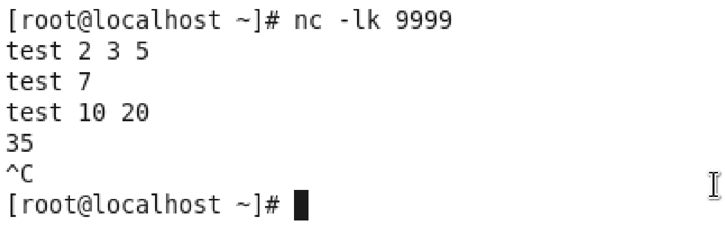
Here is the command to submit the jar file for the execution of the above code:

$ spark-submit --master local[2] -class org.scala.EvenLines Assignment20_1.jar localhost 9999



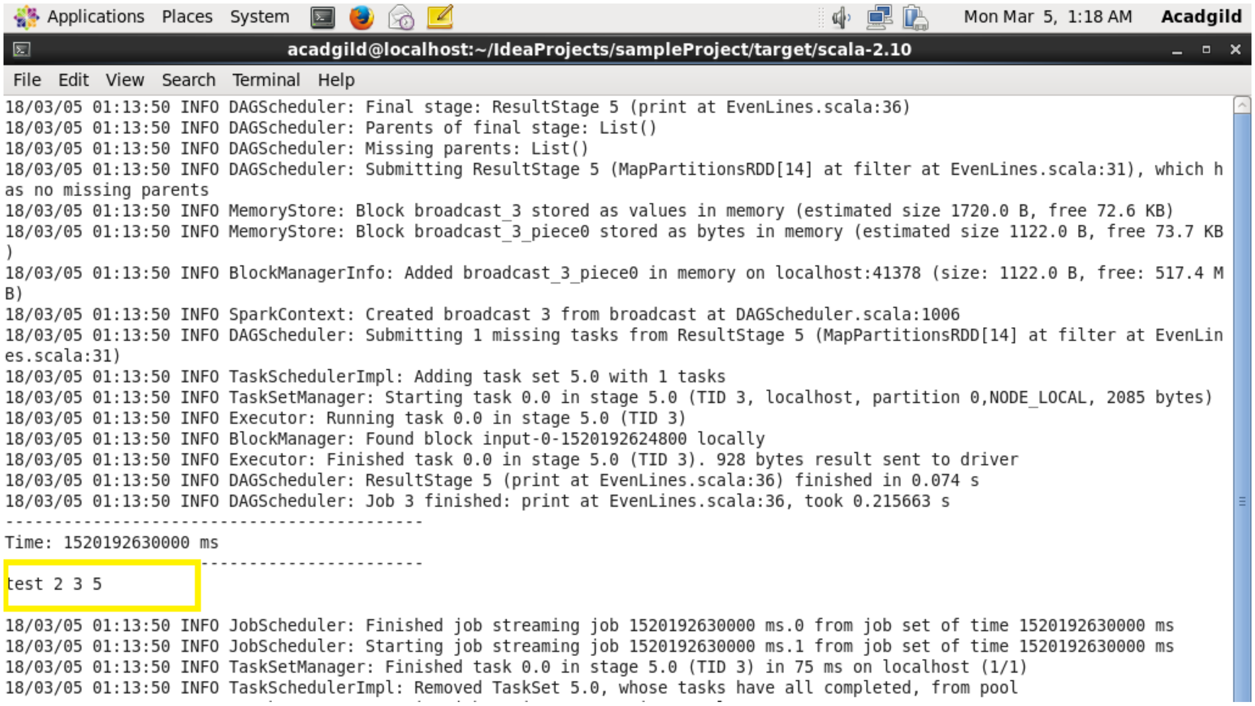
```
[acadgild@localhost scala-2.10]$
[acadgild@localhost scala-2.10]$ spark-submit --master local[2] --class org.spark.EvenLines sampleproject_2.10-0.1.jar localh
ost 9999
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/03/05 01:06:05 INFO SparkContext: Running Spark version 1.6.0
18/03/05 01:06:07 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
 where applicable
18/03/05 01:06:08 WARN SparkConf:
SPARK_WORKER_INSTANCES was detected (set to '2').
This is deprecated in Spark 1.0+.

Please instead use:
 - ./spark-submit with --num-executors to specify the number of executors
 - Or set SPARK_EXECUTOR_INSTANCES
 - spark.executor.instances to configure the number of instances in the spark config.

18/03/05 01:06:08 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.4
3.63 instead (on interface eth0)
18/03/05 01:06:08 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
18/03/05 01:06:08 INFO SecurityManager: Changing view acls to: acadgild
18/03/05 01:06:08 INFO SecurityManager: Changing modify acls to: acadgild
18/03/05 01:06:08 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissio
ns: Set(acadgild); users with modify permissions: Set(acadgild)
█
```

On another terminal we need to run netcat command to start entering some numbers which will be captured by the streaming application

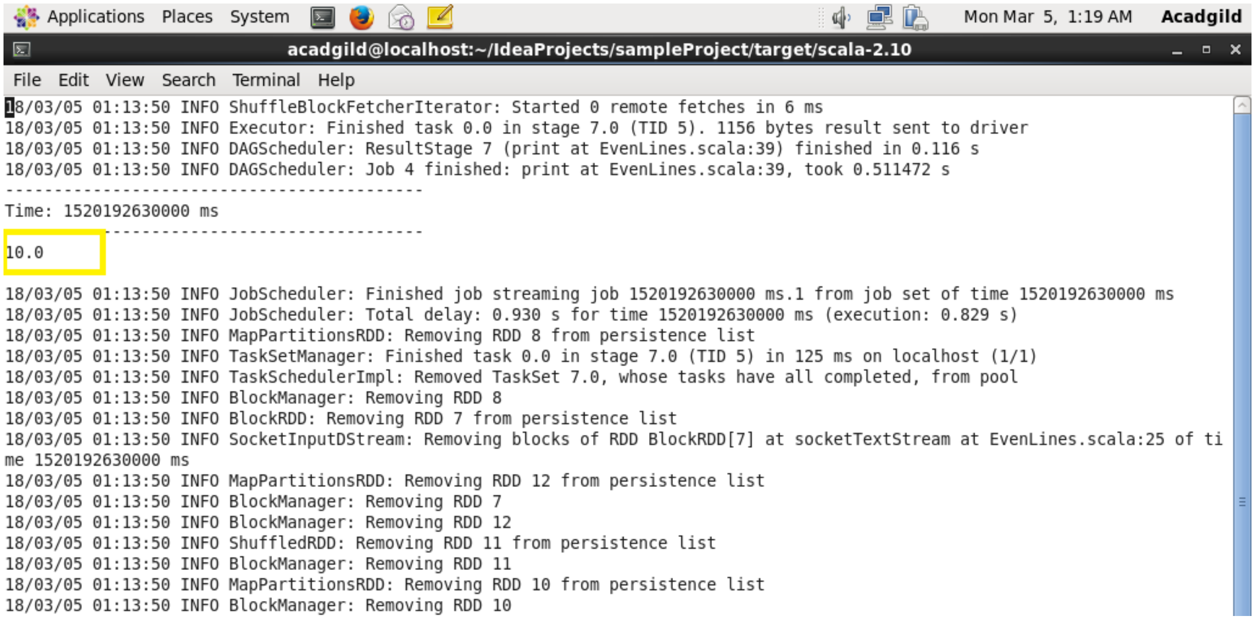


```
[root@localhost ~]# nc -lk 9999
test 2 3 5
test 7
test 10 20
35
^C
[root@localhost ~]# 
```

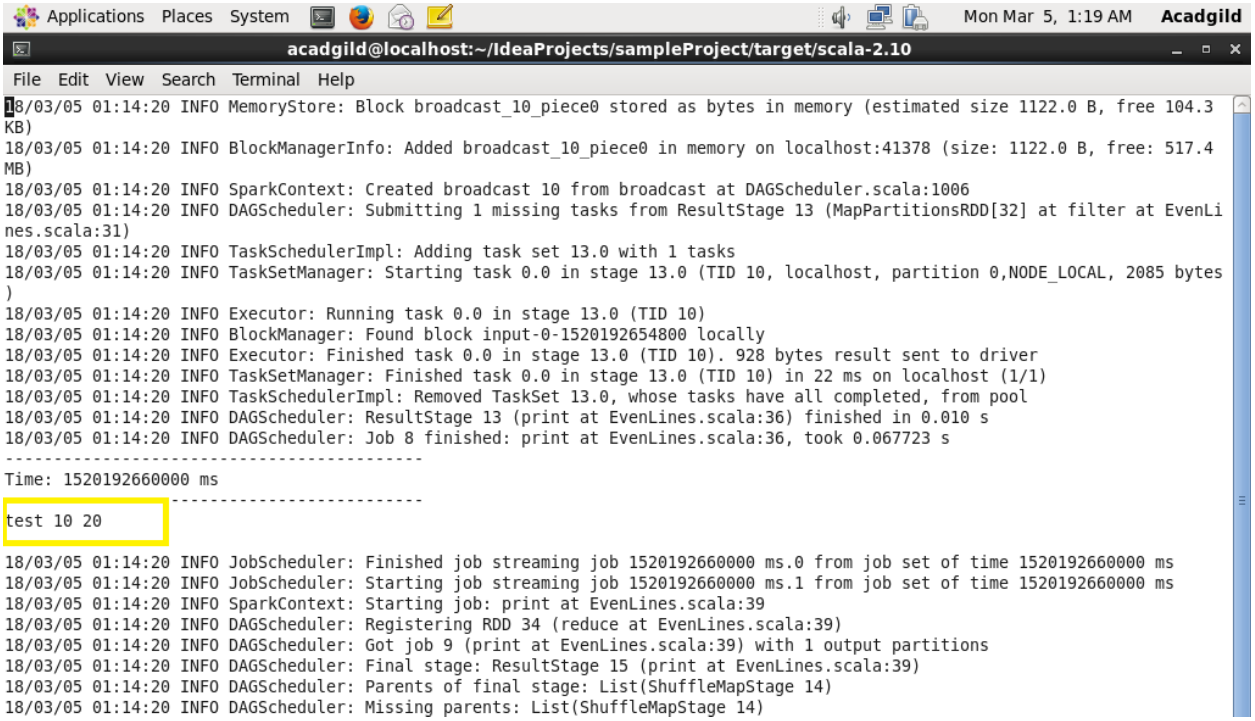
Let’s observe our streaming application run to see the output:



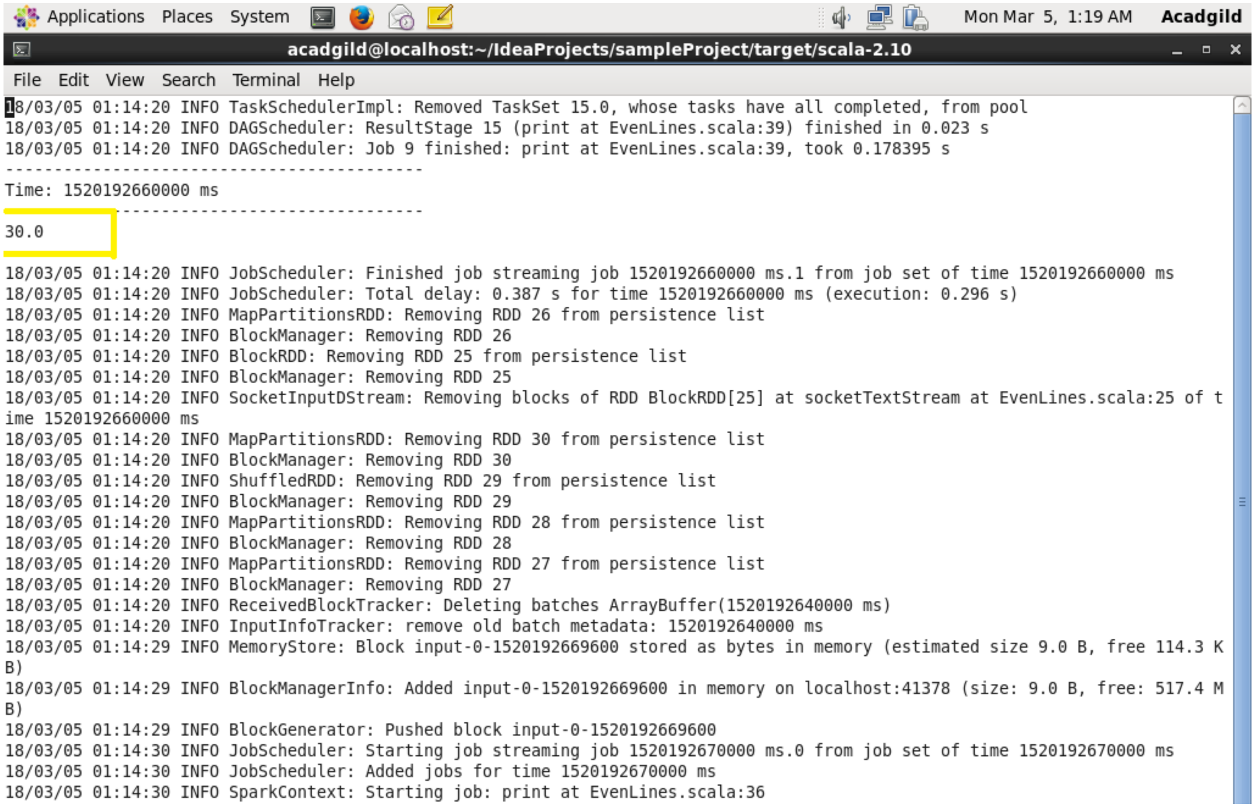
```
18/03/05 01:13:50 INFO DAGScheduler: Final stage: ResultStage 5 (print at EvenLines.scala:36)
18/03/05 01:13:50 INFO DAGScheduler: Parents of final stage: List()
18/03/05 01:13:50 INFO DAGScheduler: Missing parents: List()
18/03/05 01:13:50 INFO DAGScheduler: Submitting ResultStage 5 (MapPartitionsRDD[14] at filter at EvenLines.scala:31), which h
as no missing parents
18/03/05 01:13:50 INFO MemoryStore: Block broadcast_3 stored as values in memory (estimated size 1720.0 B, free 72.6 KB)
18/03/05 01:13:50 INFO MemoryStore: Block broadcast_3_piece0 stored as bytes in memory (estimated size 1122.0 B, free 73.7 KB
)
18/03/05 01:13:50 INFO BlockManagerInfo: Added broadcast_3_piece0 in memory on localhost:41378 (size: 1122.0 B, free: 517.4 M
B)
18/03/05 01:13:50 INFO SparkContext: Created broadcast 3 from broadcast at DAGScheduler.scala:1006
18/03/05 01:13:50 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 5 (MapPartitionsRDD[14] at filter at EvenLin
es.scala:31)
18/03/05 01:13:50 INFO TaskSchedulerImpl: Adding task set 5.0 with 1 tasks
18/03/05 01:13:50 INFO TaskSetManager: Starting task 0.0 in stage 5.0 (TID 3, localhost, partition 0,NODE_LOCAL, 2085 bytes)
18/03/05 01:13:50 INFO Executor: Running task 0.0 in stage 5.0 (TID 3)
18/03/05 01:13:50 INFO BlockManager: Found block input-0-1520192624800 locally
18/03/05 01:13:50 INFO Executor: Finished task 0.0 in stage 5.0 (TID 3). 928 bytes result sent to driver
18/03/05 01:13:50 INFO DAGScheduler: ResultStage 5 (print at EvenLines.scala:36) finished in 0.074 s
18/03/05 01:13:50 INFO DAGScheduler: Job 3 finished: print at EvenLines.scala:36, took 0.215663 s
-------------------------------------------
Time: 1520192630000 ms
-------------------------------------------
test 2 3 5

18/03/05 01:13:50 INFO JobScheduler: Finished job streaming job 1520192630000 ms.0 from job set of time 1520192630000 ms
18/03/05 01:13:50 INFO JobScheduler: Starting job streaming job 1520192630000 ms.1 from job set of time 1520192630000 ms
18/03/05 01:13:50 INFO TaskSetManager: Finished task 0.0 in stage 5.0 (TID 3) in 75 ms on localhost (1/1)
18/03/05 01:13:50 INFO TaskSchedulerImpl: Removed TaskSet 5.0, whose tasks have all completed, from pool
```



```
18/03/05 01:13:50 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 6 ms
18/03/05 01:13:50 INFO Executor: Finished task 0.0 in stage 7.0 (TID 5). 1156 bytes result sent to driver
18/03/05 01:13:50 INFO DAGScheduler: ResultStage 7 (print at EvenLines.scala:39) finished in 0.116 s
18/03/05 01:13:50 INFO DAGScheduler: Job 4 finished: print at EvenLines.scala:39, took 0.511472 s
-------------------------------------------
Time: 1520192630000 ms
-------------------------------------------
10.0

18/03/05 01:13:50 INFO JobScheduler: Finished job streaming job 1520192630000 ms.1 from job set of time 1520192630000 ms
18/03/05 01:13:50 INFO JobScheduler: Total delay: 0.930 s for time 1520192630000 ms (execution: 0.829 s)
18/03/05 01:13:50 INFO MapPartitionsRDD: Removing RDD 8 from persistence list
18/03/05 01:13:50 INFO TaskSetManager: Finished task 0.0 in stage 7.0 (TID 5) in 125 ms on localhost (1/1)
18/03/05 01:13:50 INFO TaskSchedulerImpl: Removed TaskSet 7.0, whose tasks have all completed, from pool
18/03/05 01:13:50 INFO BlockManager: Removing RDD 8
18/03/05 01:13:50 INFO BlockRDD: Removing RDD 7 from persistence list
18/03/05 01:13:50 INFO SocketInputDStream: Removing blocks of RDD BlockRDD[7] at socketTextStream at EvenLines.scala:25 of ti
me 1520192630000 ms
18/03/05 01:13:50 INFO MapPartitionsRDD: Removing RDD 12 from persistence list
18/03/05 01:13:50 INFO BlockManager: Removing RDD 7
18/03/05 01:13:50 INFO BlockManager: Removing RDD 12
18/03/05 01:13:50 INFO ShuffledRDD: Removing RDD 11 from persistence list
18/03/05 01:13:50 INFO BlockManager: Removing RDD 11
18/03/05 01:13:50 INFO MapPartitionsRDD: Removing RDD 10 from persistence list
18/03/05 01:13:50 INFO BlockManager: Removing RDD 10
```


Applications Places System
Mon Mar 5, 1:19 AM
Acadgild
acadgild@localhost:~/IdeaProjects/sampleProject/target/scala-2.10
File Edit View Search Terminal Help

```
18/03/05 01:14:20 INFO MemoryStore: Block broadcast_10_piece0 stored as bytes in memory (estimated size 1122.0 B, free 104.3
KB)
18/03/05 01:14:20 INFO BlockManagerInfo: Added broadcast_10_piece0 in memory on localhost:41378 (size: 1122.0 B, free: 517.4
MB)
18/03/05 01:14:20 INFO SparkContext: Created broadcast 10 from broadcast at DAGScheduler.scala:1006
18/03/05 01:14:20 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 13 (MapPartitionsRDD[32] at filter at EvenLi
nes.scala:31)
18/03/05 01:14:20 INFO TaskSchedulerImpl: Adding task set 13.0 with 1 tasks
18/03/05 01:14:20 INFO TaskSetManager: Starting task 0.0 in stage 13.0 (TID 10, localhost, partition 0,NODE_LOCAL, 2085 bytes
)
18/03/05 01:14:20 INFO Executor: Running task 0.0 in stage 13.0 (TID 10)
18/03/05 01:14:20 INFO BlockManager: Found block input-0-1520192654800 locally
18/03/05 01:14:20 INFO Executor: Finished task 0.0 in stage 13.0 (TID 10). 928 bytes result sent to driver
18/03/05 01:14:20 INFO TaskSetManager: Finished task 0.0 in stage 13.0 (TID 10) in 22 ms on localhost (1/1)
18/03/05 01:14:20 INFO TaskSchedulerImpl: Removed TaskSet 13.0, whose tasks have all completed, from pool
18/03/05 01:14:20 INFO DAGScheduler: ResultStage 13 (print at EvenLines.scala:36) finished in 0.010 s
18/03/05 01:14:20 INFO DAGScheduler: Job 8 finished: print at EvenLines.scala:36, took 0.067723 s
-------------------------------------------
Time: 1520192660000 ms
-------------------------------------------
test 10 20

18/03/05 01:14:20 INFO JobScheduler: Finished job streaming job 1520192660000 ms.0 from job set of time 1520192660000 ms
18/03/05 01:14:20 INFO JobScheduler: Starting job streaming job 1520192660000 ms.1 from job set of time 1520192660000 ms
18/03/05 01:14:20 INFO SparkContext: Starting job: print at EvenLines.scala:39
18/03/05 01:14:20 INFO DAGScheduler: Registering RDD 34 (reduce at EvenLines.scala:39)
18/03/05 01:14:20 INFO DAGScheduler: Got job 9 (print at EvenLines.scala:39) with 1 output partitions
18/03/05 01:14:20 INFO DAGScheduler: Final stage: ResultStage 15 (print at EvenLines.scala:39)
18/03/05 01:14:20 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 14)
18/03/05 01:14:20 INFO DAGScheduler: Missing parents: List(ShuffleMapStage 14)
```


Applications Places System
Mon Mar 5, 1:19 AM
Acadgild
acadgild@localhost:~/IdeaProjects/sampleProject/target/scala-2.10
File Edit View Search Terminal Help

```
18/03/05 01:14:20 INFO TaskSchedulerImpl: Removed TaskSet 15.0, whose tasks have all completed, from pool
18/03/05 01:14:20 INFO DAGScheduler: ResultStage 15 (print at EvenLines.scala:39) finished in 0.023 s
18/03/05 01:14:20 INFO DAGScheduler: Job 9 finished: print at EvenLines.scala:39, took 0.178395 s
-------------------------------------------
Time: 1520192660000 ms
-------------------------------------------
30.0

18/03/05 01:14:20 INFO JobScheduler: Finished job streaming job 1520192660000 ms.1 from job set of time 1520192660000 ms
18/03/05 01:14:20 INFO JobScheduler: Total delay: 0.387 s for time 1520192660000 ms (execution: 0.296 s)
18/03/05 01:14:20 INFO MapPartitionsRDD: Removing RDD 26 from persistence list
18/03/05 01:14:20 INFO BlockManager: Removing RDD 26
18/03/05 01:14:20 INFO BlockRDD: Removing RDD 25 from persistence list
18/03/05 01:14:20 INFO BlockManager: Removing RDD 25
18/03/05 01:14:20 INFO SocketInputDStream: Removing blocks of RDD BlockRDD[25] at socketTextStream at EvenLines.scala:25 of t
ime 1520192660000 ms
18/03/05 01:14:20 INFO MapPartitionsRDD: Removing RDD 30 from persistence list
18/03/05 01:14:20 INFO BlockManager: Removing RDD 30
18/03/05 01:14:20 INFO ShuffledRDD: Removing RDD 29 from persistence list
18/03/05 01:14:20 INFO BlockManager: Removing RDD 29
18/03/05 01:14:20 INFO MapPartitionsRDD: Removing RDD 28 from persistence list
18/03/05 01:14:20 INFO BlockManager: Removing RDD 28
18/03/05 01:14:20 INFO MapPartitionsRDD: Removing RDD 27 from persistence list
18/03/05 01:14:20 INFO BlockManager: Removing RDD 27
18/03/05 01:14:20 INFO ReceivedBlockTracker: Deleting batches ArrayBuffer(1520192640000 ms)
18/03/05 01:14:20 INFO InputInfoTracker: remove old batch metadata: 1520192640000 ms
18/03/05 01:14:29 INFO MemoryStore: Block input-0-1520192669600 stored as bytes in memory (estimated size 9.0 B, free 114.3 K
B)
18/03/05 01:14:29 INFO BlockManagerInfo: Added input-0-1520192669600 in memory on localhost:41378 (size: 9.0 B, free: 517.4 M
B)
18/03/05 01:14:29 INFO BlockGenerator: Pushed block input-0-1520192669600
18/03/05 01:14:30 INFO JobScheduler: Starting job streaming job 1520192670000 ms.0 from job set of time 1520192670000 ms
18/03/05 01:14:30 INFO JobScheduler: Added jobs for time 1520192670000 ms
18/03/05 01:14:30 INFO SparkContext: Starting job: print at EvenLines.scala:36
```

As we can see in the screenshots above, our program has picked up all the lines entered and filtered out those resulting in sum of odd number and has printed the rest.