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.

Soultion-

Below is the code used to read a stream of strings and filtering the rows for even summed numbers-

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

//main method-

object EvenLines {
  def main(args: Array[String]) {
    if (args.length < 2) {
      System.err.println("Usage: EvenLines <localhost> <9999>")
      System.exit(1)
    }
    StreamingExamples.setStreamingLogLevels()

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

    var strList = "";
    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; }}
```

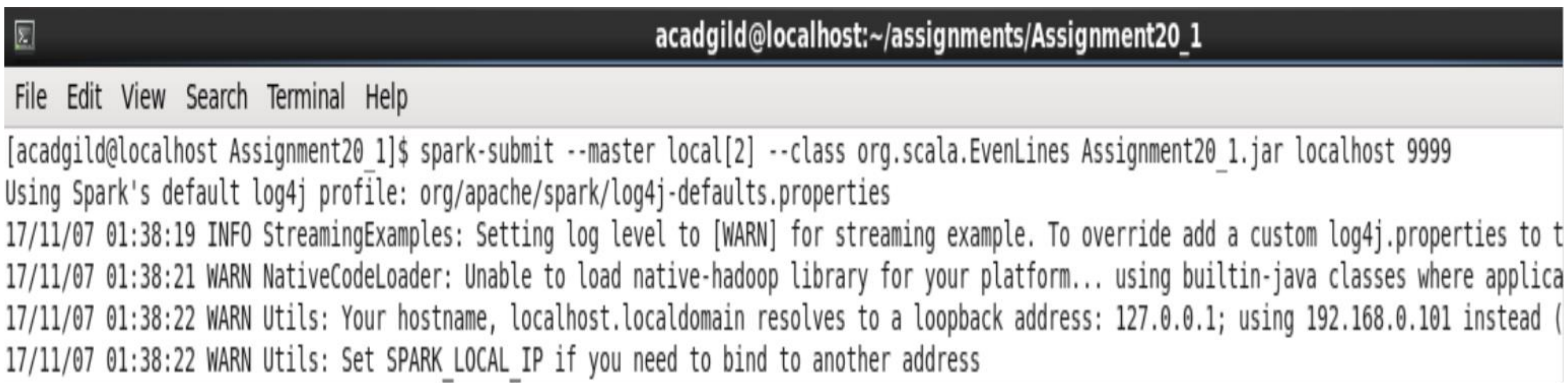
Code to define level of logs-

```
package org.scala
import org.apache.log4j.{Level, Logger}
import org.apache.spark.internal.Logging


/** Utility functions for Spark Streaming examples. */
object StreamingExamples extends Logging {

  /** Set reasonable logging levels for streaming if the user has not configured log4j. */
  def setStreamingLogLevels() {
    val log4jInitialized = Logger.getRootLogger.getAllAppenders.hasMoreElements
    if (!log4jInitialized) {
      // We first log something to initialize Spark's default logging, then we override the
      // logging level.
      logInfo("Setting log level to [WARN] for streaming example." +
        " To override add a custom log4j.properties to the classpath.")
      Logger.getRootLogger.setLevel(Level.WARN)     }}}
```

Now we will run the jar in spark-shell to run the application-

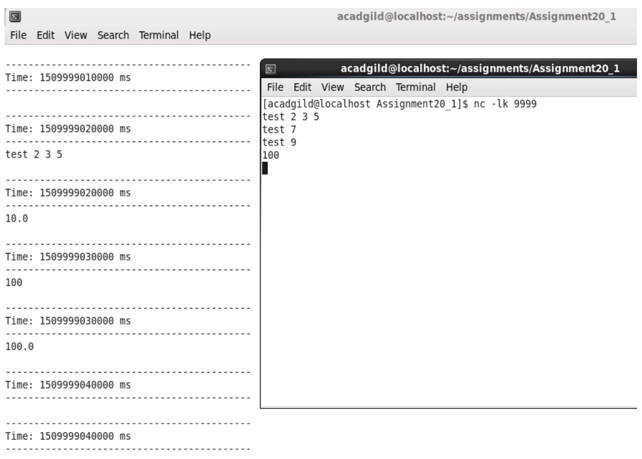

acadgild@localhost:~/assignments/Assignment20_1
File Edit View Search Terminal Help

```
[acadgild@localhost Assignment20_1]$ spark-submit --master local[2] --class org.scala.EvenLines Assignment20_1.jar localhost 9999
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
17/11/07 01:38:19 INFO StreamingExamples: Setting log level to [WARN] for streaming example. To override add a custom log4j.properties to t
17/11/07 01:38:21 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applica
17/11/07 01:38:22 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.0.101 instead (
17/11/07 01:38:22 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
```

Now we will input some stream in netcat server as shown below-

- [acadgild@localhost Assignment20_1]$ nc -lk 9999
- Sample input –
- test 2 3 5
- test 7
- test 9
- 100

Below shows the screenshot for output-


acadgild@localhost:~/assignments/Assignment20_1
File Edit View Search Terminal Help
acadgild@localhost:~/assignments/Assignment20_1
File Edit View Search Terminal Help

```
-------------------------------------------
Time: 1509999010000 ms
-------------------------------------------

-------------------------------------------
Time: 1509999020000 ms
-------------------------------------------
test 2 3 5

-------------------------------------------
Time: 1509999020000 ms
-------------------------------------------
10.0

-------------------------------------------
Time: 1509999030000 ms
-------------------------------------------
100

-------------------------------------------
Time: 1509999030000 ms
-------------------------------------------
100.0

-------------------------------------------
Time: 1509999040000 ms
-------------------------------------------

-------------------------------------------
Time: 1509999040000 ms
-------------------------------------------
```

```
[acadgild@localhost Assignment20_1]$ nc -lk 9999
test 2 3 5
test 7
test 9
100
```

odd sum lines test 7 and test 9 are ignored and even sum lines and their sum is printed.