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:

```
        def toInteger(s: String): Option[Int] = {
          try {
            Some(s.toInt)
          } catch {
            case e: Exception => None
          }
        }
        spark-shell --master local[4]
        import org.apache.spark._
        import org.apache.spark.streaming._
        import org.apache.spark.streaming.StreamingContext._
        import org.apache.spark.SparkContext._
        import org.apache.spark.rdd.RDD
        val ssc = new StreamingContext(sc, Seconds(10))
        val lines = ssc.socketTextStream("localhost.localdomain", 9999)
        val lines1 = lines.flatMap(x => x.split(" ")).filter(x => x.matches("[0-9]+"))
        val tes = if(lines1%2==0) lines1 : lines
        tes.print()
        def findSum(line: String, sum:Int):String={
        if(sum%2!=0) line
        else sum.toString
        }
        val sum_even= findSum(lines,sum_lines)

        ssc.start()
ssc.awaitTermination()
```