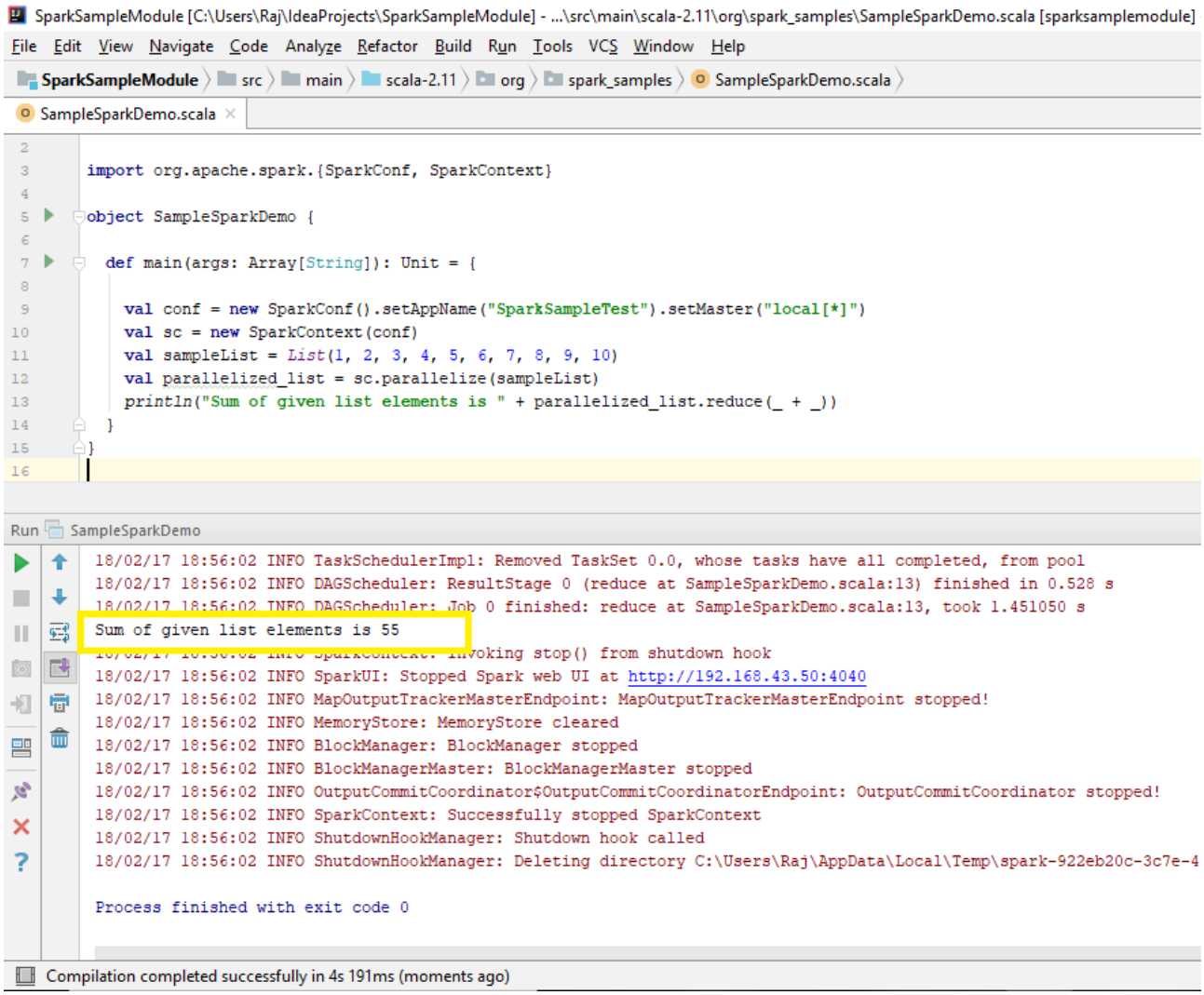
# TASK 1

1) Sum of all numbers

```
import org.apace.spark.{SparkConf,SparkContext}
object SampleSparkDemo {
        def main(args: Array[String]): Unit = {
        val conf = new SparkConf().setAppName("SparkSampleTest").setMaster("local[*]")
        val sc = new SparkContext(conf)
        val sampleList = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
        val parallelized_list = sc.parallelize(sampleList)
        println("Sum of given list elements is " + parallelized_list.reduce(_ + _))
    }
}
```

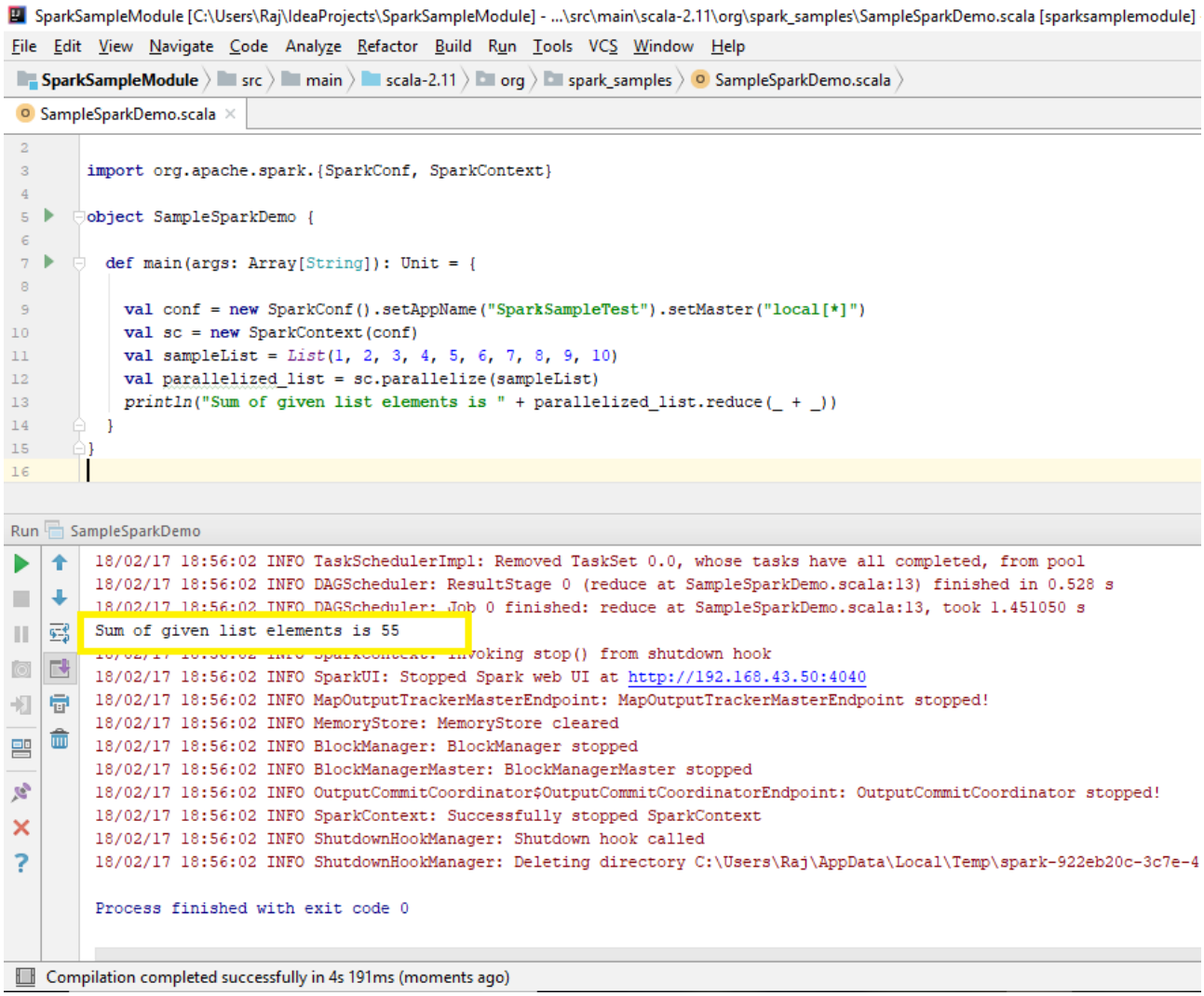



**Output:**

Total number of elements in the list is 55

2) Total elements in the list

```
import org.apace.spark.{SparkConf,SparkContext}
object SampleSparkDemo {
        def main(args: Array[String]): Unit = {
        val conf = new SparkConf().setAppName("SparkSampleTest").setMaster("local[*]")
        val sc = new SparkContext(conf)
        val sampleList = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
        val parallelized_list = sc.parallelize(sampleList)
        println("Total number of elements in the list is " + parallelized_list.count())
    }
}
```

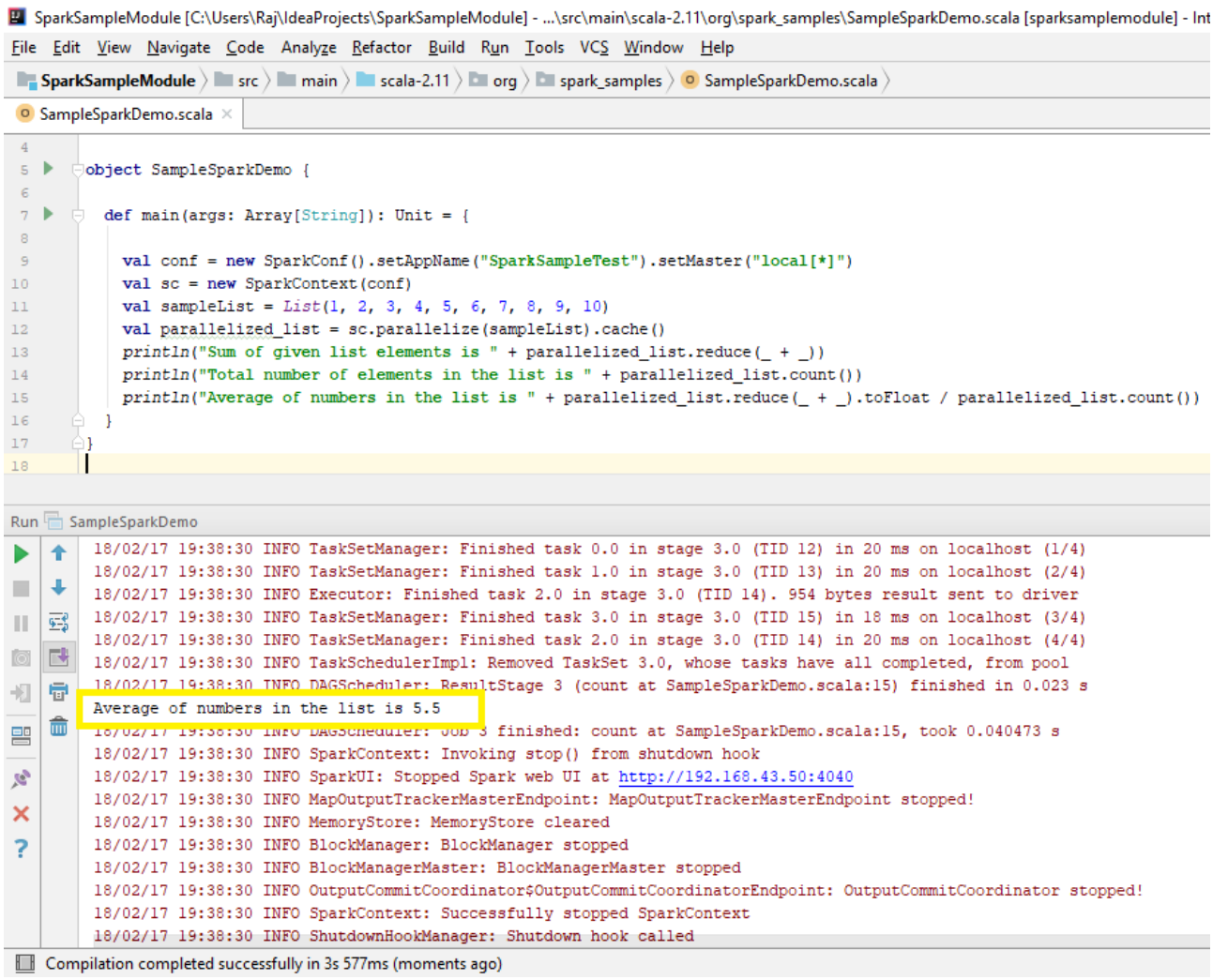



**Output:**

Total number of elements in the list is 10

3) Average of the numbers in the list

```
import org.apace.spark.{SparkConf,SparkContext}
object SampleSparkDemo {
        def main(args: Array[String]): Unit = {
        val conf = new SparkConf().setAppName("SparkSampleTest").setMaster("local[*]")
        val sc = new SparkContext(conf)
        val sampleList = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
        val parallelized_list = sc.parallelize(sampleList)
        println("Average of numbers in the list is " + parallelized_list.reduce(_ + _).toFloat /
parallelized_list.count())
    }
}
```

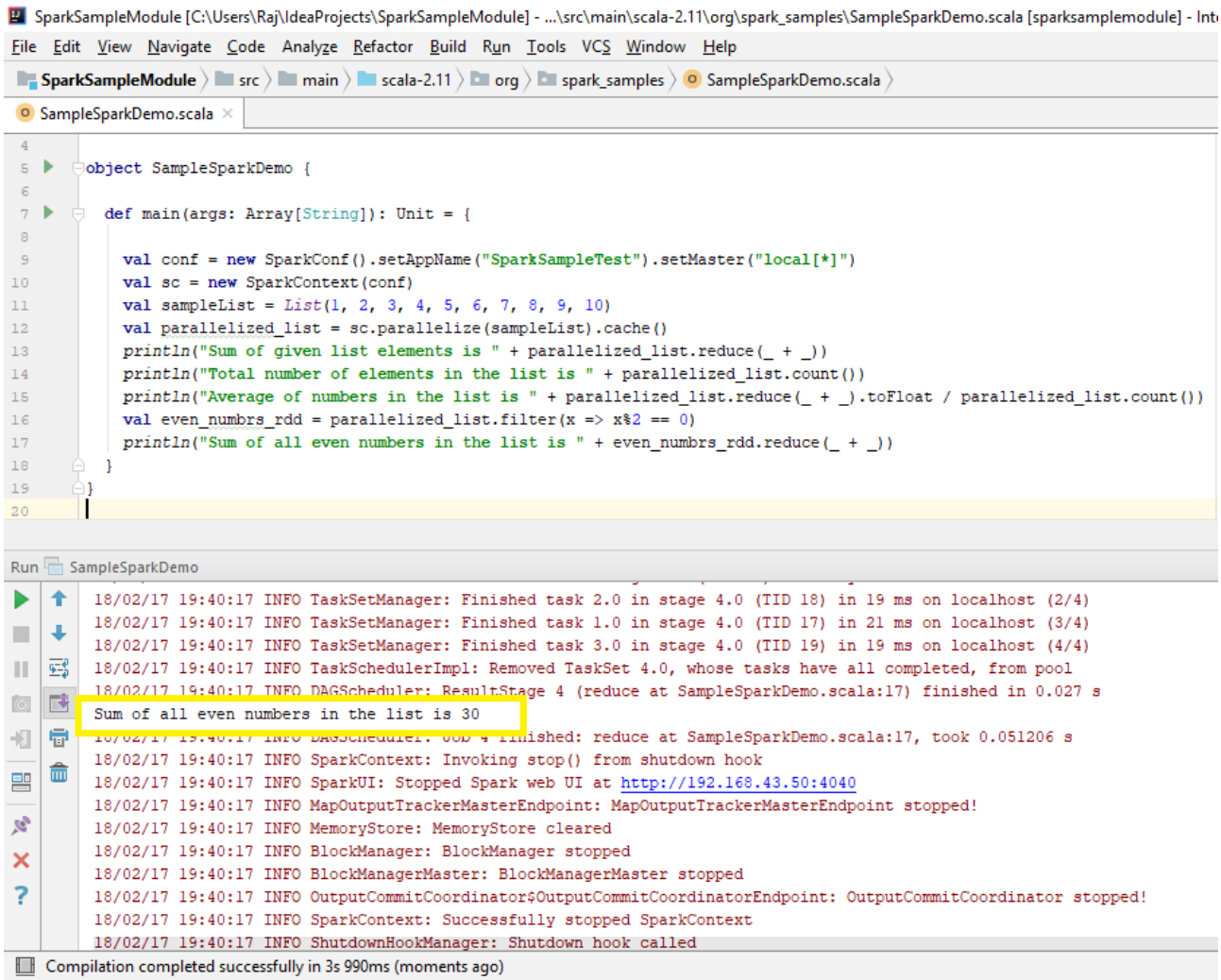



**Output:**

Average of numbers in the list is 5.5

4) Sum of all the even numbers in the list

```
import org.apace.spark.{SparkConf,SparkContext}
object SampleSparkDemo {
        def main(args: Array[String]): Unit = {
        val conf = new SparkConf().setAppName("SparkSampleTest").setMaster("local[*]")
        val sc = new SparkContext(conf)
        val sampleList = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
        val parallelized_list = sc.parallelize(sampleList)
        val even_numbrs_rdd = parallelized_list.filter(x => x%2 == 0)
       println("Sum of all even numbers in the list is " + even_numbrs_rdd.reduce(_ + _))
   }
}
```

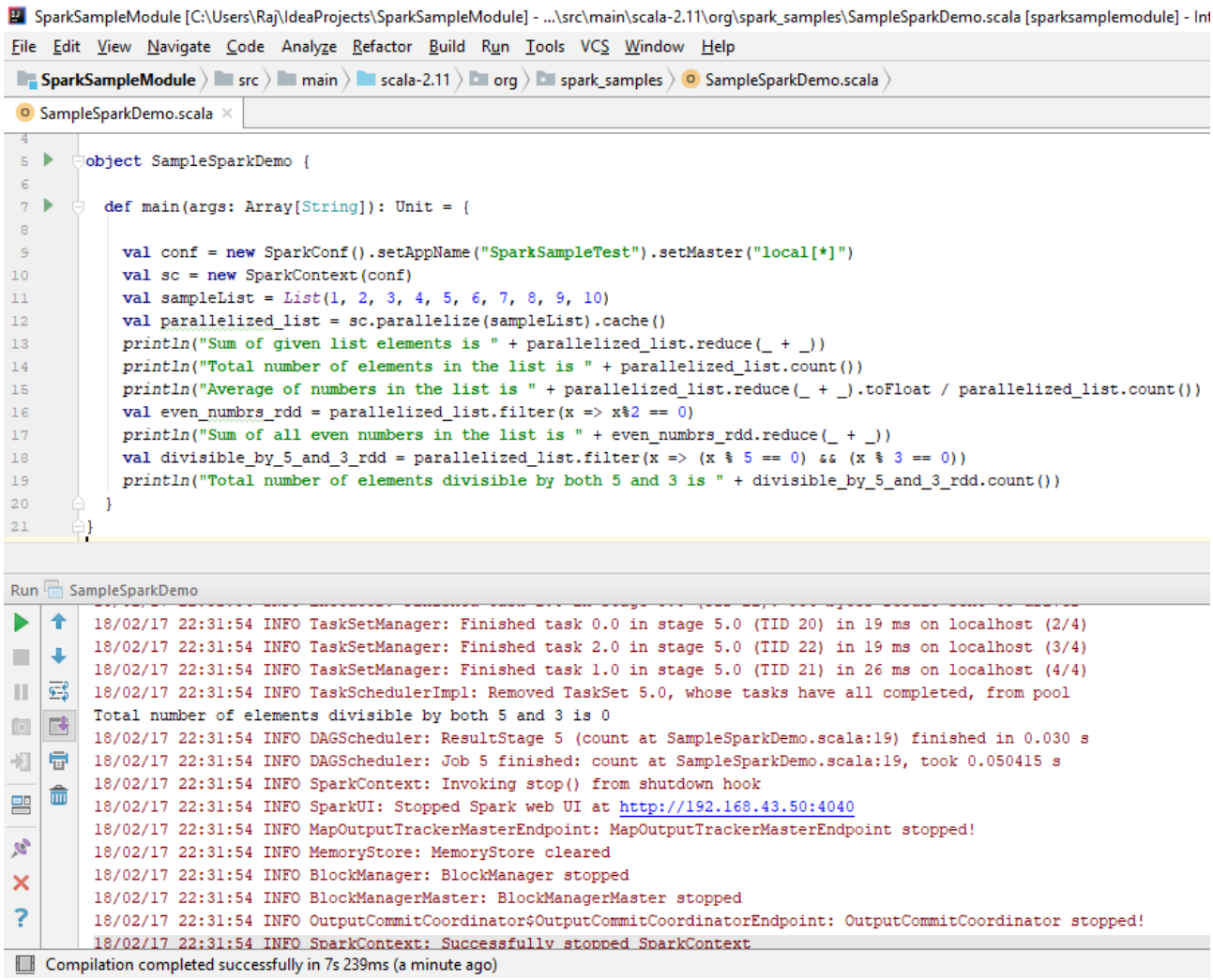



**Output:**

Sum of all even numbers in the list is 30

5) Total number of elements in the list divisible by both 5 and 3

```
import org.apace.spark.{SparkConf,SparkContext}
object SampleSparkDemo {
        def main(args: Array[String]): Unit = {
        val conf = new SparkConf().setAppName("SparkSampleTest").setMaster("local[*]")
        val sc = new SparkContext(conf)
        val sampleList = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
        val parallelized_list = sc.parallelize(sampleList)
       val divisible_by_5_and_3_rdd = parallelized_list.filter(x => (x % 5 == 0) && (x % 3 == 0))
       println("Total number of elements divisible by both 5 and 3 is " +
       divisible_by_5_and_3_rdd.count())
   }
}
```




**Output:**

Total number of elements divisible by both 5 and 3 is 0