Assignment 16.1

Problem Statement :-

Given a list of numbers - List[Int] (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

- find the sum of all numbers

- find the total elements in the list

- calculate the average of the numbers in the list

- find the sum of all the even numbers in the list

- find the total number of elements in the list divisible by both 5 and 3

Solution:-

List of Numbers is :-

```
scala> val ListRDD = sc.parallelize(List(1,2,3,4,5,6,7,8,9,10))
ListRDD: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala>
```

- **Sum of all numbers**

```
scala> val sum = ListRDD.sum
sum: Double = 55.0

scala>
```

- **Total elements in the list**

```
scala> val countRDD = ListRDD.count()
countRDD: Long = 10

scala>
```

- **Average of the numbers in the list**

```
scala> val avgRDD = ListRDD.sum/ListRDD.count()
avgRDD: Double = 5.5

scala>
```

- **Sum of all the even numbers in the list**

```
scala> val evenRDD = ListRDD.filter(x => x%2 == 0)
evenRDD: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[3] at filter at <console>:26

scala> evenRDD.collect()
res0: Array[Int] = Array(2, 4, 6, 8, 10)

scala> val sumEvenRDD = evenRDD.sum
sumEvenRDD: Double = 30.0

scala>
```

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

```
scala> val divRDD = ListRDD.filter(x => x%3 == 0 && x%5 == 0)
divRDD: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[5] at filter at <console>:26

scala> divRDD.collect()
res1: Array[Int] = Array()

scala>
```