Problem Statement

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

1. find the sum of all numbers

Solution-

We are creating a RDD in which we are defining a List of integers below with name **ListRDD.** This will be used in further operations also

```
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>
```

Below we are calculating the sum of all elements present in List using sum operation

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

scala>
```

2. Find the total elements in the list.

Solution- Below we are using count() operation on previously created RDD ListRDD to fin the total elements in the list.

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

scala>
```

3. Calculate the average of the numbers in the list

Solution-

Below we are using sum() and count() both on previously created RDD ListRDD to calculate average of the numbers in the list-

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

scala>
```

4. Find the sum of all the even numbers in the list-

Solution-

Here we are performing multiple steps-

a. First we are creating a RDD to filter the elements from previously created RDD ListRDD which are even numbers using below code-
Val evenRDD = ListRDD.filter(x => x%2 ==)
evenRDD.collect() shows the result of same

b. Now we are adding the elements present in evenRDD using sum() to get the sum of all even numbers-

```
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>
```

5. Find the total number of elements in the list divisible by both 5 and 3

Solution- As per the question here we are creating the RDD divRDD by filtering the elements from ListRDD which are divisible by 5 and 3.

Then collect() shows that there are no elements in the List which are 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>
```