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

1. find the sum of all numbers

   val list = sc.parallelize(List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
   val sum = list.sum.toInt
   println(sum)

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

scala> val sum = list.sum.toInt
sum: Int = 55

scala> println(sum)
55
```

2. find the total elements in the list

   val list = sc.parallelize(List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))

   val length = list.count

   println(length)

```
scala> val list = sc.parallelize(List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
list: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[13] at parallelize at <console>:24

scala> val length = list.count
length: Long = 10

scala> println(length)
10

scala>
```

3. calculate the average of the numbers in the list

val list = sc.parallelize(List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
val sum = list.sum
val length = list.count
val result = sum/length

```
scala> val list = sc.parallelize(List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
list: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[15] at parallelize at <console>:24

scala> val sum = list.sum
sum: Double = 55.0

scala> val length = list.count
length: Long = 10

scala> val result = sum/length
result: Double = 5.5

scala>
```

4. find the sum of all the even numbers in the list

val list = sc.parallelize(Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))

val evenNumbers = list.filter(x=>(x%2==0))

val sum = evenNumbers.sum

println(sum)

```
scala> val list = sc.parallelize(Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
list: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[38] at parallelize at <console>:24

scala> val evenNumbers = list.filter(x=>(x%2==0))
evenNumbers: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[39] at filter at <console>:26

scala> val sum = evenNumbers.sum
sum: Double = 30.0

scala> println(sum)
30.0
```

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

val list = sc.parallelize(Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))

val numberDivisibility = list.filter(x=>(x%3==0&&x%5==0)).count

println(numberDivisibility)

```
scala> val list = sc.parallelize(Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
list: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[49] at parallelize at <console>:24

scala> val numberDivisibility = list.filter(x=>(x%3==0&&x%5==0)).count
numberDivisibility: Long = 0

scala> println(numberDivisibility)
0

scala>
```