ASSIGNMENT16.1

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

find the total elements in the list

```
scala> val listrr=sc.parallelize(List(1,2,3,4,5,6,7,8,9,10))
listrr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at
<console>:27
scala> val length=listrr.count
length: Long = 10
```

find the sum of all numbers

```
scala> val sum=listrr.reduce((x,y)=>x+y).toDouble
sum: Double = 55.0
```

calculate the average of the numbers in the list

```
scala> val avg=sum/length
avg: Double = 5.5
```

find the sum of all the even numbers in the list

```
scala> val evennum=listrr.filter{x=>(x%2)==0}
evennum: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at
<console>:29

scala> val evensum=evennum.reduce((x,y)=>x+y)
evensum: Int = 30
```

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

```
scala> val numdiv=listrr.filter{x=>(x%3)==0}
numdiv: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[6] at filter at
<console>:29

scala> val divi=numdiv.filter{x=>(x%5)==0}
divi: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[7] at filter at
<console>:31

scala> val total=divi.count
total: Long = 0
```