SPARK 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

```
//Declare the List in form of RDD with the help of Spark Context Object

val rdd1 = sc.parallelize(List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))

//Reduce function takes elements from RDD and adds them up.

val sum = rdd1.reduce(_+_)

//1-display the sum of all number

println(sum)
```

Output_1: The sum of all numbers in the list

```
scala> val rdd1 = sc.parallelize(List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
rdd1: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:27

scala> val sum = rdd1.reduce(_+_)
sum: Int = 55

scala> println(sum)
55
```

```
//calculating the total elements in the list

val cnt = rdd1.count()

//2-display the total elements in the list

println(cnt)
```

Output _2: the total elements in the list

```
scala> val cnt = rdd1.count()
cnt: Long = 10

scala> println(cnt)
10

scala>
```

//defining function to calculate the average

```
def avgfunc(sum : Int , count :Int): Int = {sum/count}
```

//aggregate function to calculate sum and count two separate function in one rdd operation respectively

```
val b = rdd1.aggregate((0,0))((x,y)=> (x._1 +y, x._2+1),(x,y)=> (x._1 + y._1 ,x._2 + y._2))
```

// calling the user defined function for average calculation

```
val c = avgfunc(b._1 ,b._2)
```

//3-display the average of elements

```
println(c)
```

Output _3: the average of the numbers in the list

```
scala> def avgfunc(sum : Int , count :Int): Int = {sum/count}
avgfunc: (sum: Int, count: Int)Int

scala> val b = rdd1.aggregate((0,0))((x,y)=> (x._1 +y, x._2+1),(x,y)=> (x._1 + y._1 ,x._2 + y._2))
b: (Int, Int) = (55,10)

scala> val c = avgfunc(b._1 ,b._2)
c: Int = 5

scala> println(c)
5
```

```
//filtering only the even values

val rdd3  = rdd1.filter(even=>(even%2==0))

//printing only the even values

rdd3.foreach(println)

//Summing up the even elements by reduce function

val rdd4 = rdd3.reduce(_+_)

//4-display the sum of all even elements in the list

println(rdd4)
```

Output_4: the sum of all the even numbers in the list

```
scala> val rdd3  = rdd1.filter(even=>(even%2==0))
rdd3: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at <console>:29

scala> val rdd4 = rdd3.reduce(_+_)
rdd4: Int = 30

scala> println(rdd4)
30
```

```
//filtering only those values which are divisible by 5 & 3

val rdd5 = rdd1.filter(x=>(x%5==0)).filter(x=>(x%3==0))

//total number of elements in list which are divisible by 5 & 3

val rdd6 = rdd5.count()
```

//5-display the total count of elements which are divisible by 5 & 3

println(rdd6)

Output_5: the total number of elements in the list divisible by both 5 and 3

```
scala> val rdd5 = rdd1.filter(x=>(x%5==0)).filter(x=>(x%3==0))
rdd5: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[3] at filter at <console>:29

scala> val rdd6 = rdd5.count()
rdd6: Long = 0

scala> println(rdd6)
0
```