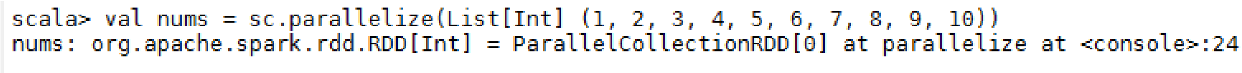
Spark Assignment 16.1

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

Started the spark shell and initialized the list.

```
val nums = sc.parallelize(List[Int] (1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
```



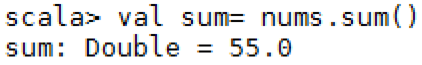
```
scala> val nums = sc.parallelize(List[Int] (1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
nums: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24
```

1) *Find the sum of all numbers:*

To find the sum of all numbers. We are going use sum() method.The code is

```
val sum= nums.sum()
```

Spark-shell output



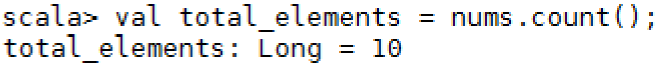
```
scala> val sum= nums.sum()
sum: Double = 55.0
```

2) *Find the total elements in the list:*

To find the number of elements . We are going use count() method.The code is

```
val total_elements = nums.count();
```

Spark-shell output



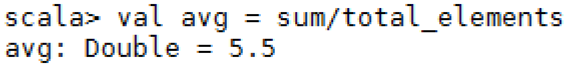
```
scala> val total_elements = nums.count();
total_elements: Long = 10
```

3) *Calculate the average of the numbers in the list:*

To calculate the average of numbers in the list . The code is

```
val avg = sum/total_elements
```

Spark-shell output



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

4) *Find the sum of all the even numbers in the list:*

To calculate the sum of all even numbers in the list, first we have to find the even numbers from the the list . We are going to use filter function first.

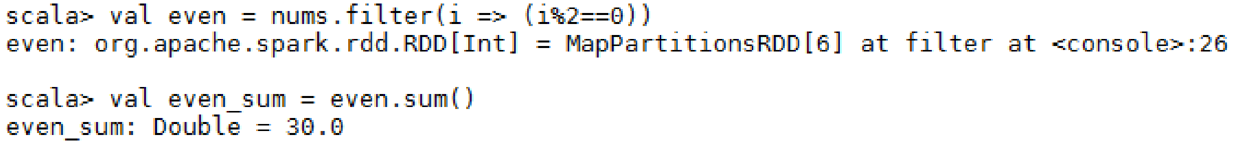
After filtering going to use sum() function, to find the sum. The code is

val even = nums.filter(i => (i%2==0))

To find sum:

val even_sum = even.sum()

Spark-shell output



```
scala> val even = nums.filter(i => (i%2==0))
even: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[6] at filter at <console>:26

scala> val even_sum = even.sum()
even_sum: Double = 30.0
```

5) *Find the total number of elements in the list divisible by both 5 and 3:*

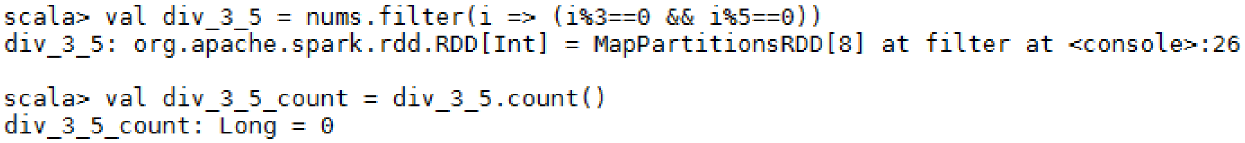
To calculate the number of elements divisible by 3 & 5, first we have to find numbers divisible by 3 & 5. We are going to use filter function for this. After filtering we are going to use count() function. The code is

val div_3_5 = nums.filter(i => (i%3==0 && i%5==0))

To find the count:

val div_3_5_count = div_3_5.count()

Spark-shell output



```
scala> val div_3_5 = nums.filter(i => (i%3==0 && i%5==0))
div_3_5: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[8] at filter at <console>:26

scala> val div_3_5_count = div_3_5.count()
div_3_5_count: Long = 0
```