Word count

```
scala> val data = sc.textFile("file:///home/chaitanya/Desktop/Sparkdata.txt")
data: org.apache.spark.rdd.RDD[String] = file:///home/chaitanya/Desktop/Sparkdata.txt MapPartitionsRDD[5] at textFile at
 <console>:23

scala> data.collect
res14: Array[String] = Array(hello world, hello scala, hello spark, "")

scala> val splitdata = data.flatMap(line => line.split(" "))
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[6] at flatMap at <console>:23

scala> splitdata.collect
res15: Array[String] = Array(hello, world, hello, scala, hello, spark, "")

scala> val mapdata=splitdata.map(word => (word,1))
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[7] at map at <console>:23

scala> mapdata.collect
res16: Array[(String, Int)] = Array((hello,1), (world,1), (hello,1), (scala,1), (hello,1), (spark,1), ("",1))

scala> val reducedata=mapdata.reduceByKey(_+_)
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[8] at reduceByKey at <console>:23

scala> reducedata.collect
res17: Array[(String, Int)] = Array((scala,1), ("",1), (hello,3), (world,1), (spark,1))

scala> val filterwords=reducedata.filter{ case (word,count) => count>4}
filterwords: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at filter at <console>:23

scala> filterwords.collect.foreach(println)

scala>
```

import scala.collection.immutable.ListMap

```
scala> val sorted = ListMap(reducedata.collect.sortWith(_._2> _._2):_*)
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(hello -> 3, scala -> 1, "" -> 1, world -> 1, spark -> 1
)

scala> for((k,v)<-sorted){
     | if(v>4){
     | print(k+",")
     | print(v)
     | println()
     | }
     | }

scala>
```

```
val listRdd = spark.sparkContext.parallelize(List(1,2,3,4,5,3,2))
println("output min using binary : "+listRdd.reduce(_ min _))
 println("output max using binary : "+listRdd.reduce(_ max _))
 println("output sum using binary : "+listRdd.reduce(_ + _))
```

Alternatively, you can also write the above operations as below. val listRdd = spark.sparkContext.parallelize(List(1,2,3,4,5,3,2))

```
 println("output min : "+listRdd.reduce( (a,b) => a min b))
println("output max : "+listRdd.reduce( (a,b) => a max b))
 println("output sum : "+listRdd.reduce( (a,b) => a + b))
```

output min : 1 output max : 5 output sum : 2

Write a Scala program to print numbers from 1 to 100 using for loop.

```
object ExampleForLoop1 {
def main(args: Array[String]) {
var counter: Int=0;
for(counter &amp;lt;- 1 to 100)
print(counter + &amp;quot; &amp;quot;);
// to print new line
println();
}
}
```