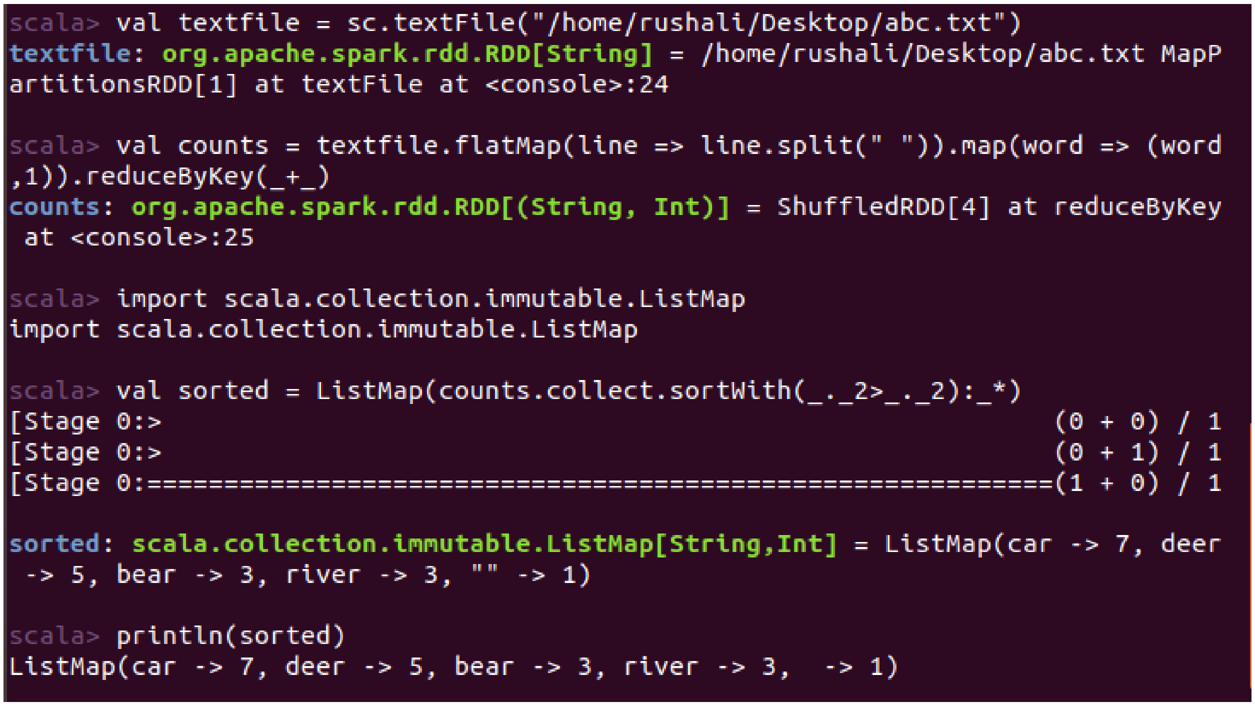
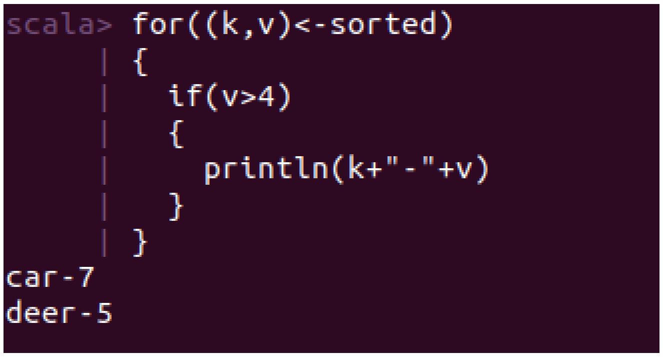
OUTPUT

```
// scala shell
scala> val textfile = sc.textFile("/home/rushali/Desktop/abc.txt")
textfile: org.apache.spark.rdd.RDD[String] = /home/rushali/Desktop/abc.txt
MapPartitionsRDD[1] at textFile at <console>:24

scala> val counts = textfile.flatMap(line => line.split(" ")).map(word =>
(word,1)).reduceByKey(+)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at
reduceByKey at <console>:25

scala> import scala.collection.immutable.ListMap
import scala.collection.immutable.ListMap

scala> val sorted = ListMap(counts.collect.sortWith(._2>.2):*)

scala> println(sorted)
ListMap(car -> 7, deer -> 5, bear -> 3, river -> 3,  -> 1)

scala> for((k,v)<-sorted)
     | {
     |   if(v>4)
     |   {
     |     println(k+"-"+v)
     |   }
     | }
car-7
deer-5
```



```
scala> val textfile = sc.textFile("/home/rushali/Desktop/abc.txt")
textfile: org.apache.spark.rdd.RDD[String] = /home/rushali/Desktop/abc.txt MapP
artitionsRDD[1] at textFile at <console>:24

scala> val counts = textfile.flatMap(line => line.split(" ")).map(word => (word
,1)).reduceByKey(_+_)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey
 at <console>:25

scala> import scala.collection.immutable.ListMap
import scala.collection.immutable.ListMap

scala> val sorted = ListMap(counts.collect.sortWith(_._2>_._2):_*)
[Stage 0:>                                                           (0 + 0) / 1
[Stage 0:>                                                           (0 + 1) / 1
[Stage 0:============================================================(1 + 0) / 1

sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(car -> 7, deer
 -> 5, bear -> 3, river -> 3, "" -> 1)

scala> println(sorted)
ListMap(car -> 7, deer -> 5, bear -> 3, river -> 3,  -> 1)
```



```
scala> for((k,v)<-sorted)
     | {
     |   if(v>4)
     |   {
     |     println(k+"-"+v)
     |   }
     | }
car-7
deer-5
```