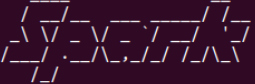


Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ echo "code code code code code spark spark spark spark spark hell  
o hello hi hi joe ken">input.txt  
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ spark-shell  
25/05/26 16:01:15 WARN Utils: Your hostname, bmscecse-HP-Elite-Tower-600-G9-Desktop-PC resolves to a loopback address:  
127.0.1.1; using 10.124.5.27 instead (on interface eno1)  
25/05/26 16:01:15 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
25/05/26 16:01:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java c  
lasses where applicable  
Spark context Web UI available at http://10.124.5.27:4040  
Spark context available as 'sc' (master = local[*], app id = local-1748255477930).  
Spark session available as 'spark'.  
Welcome to  
 version 3.5.4  
  
Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.26)  
Type in expressions to have them evaluated.  
Type :help for more information.  
  
scala> val lines=sc.textFile("input.txt")  
lines: org.apache.spark.rdd.RDD[String] = input.txt MapPartitionsRDD[1] at textFile at <console>:23  
  
scala> val words=lines.flatMap(line => line.split(" "))  
<console>:23: error: value flatMap is not a member of org.apache.spark.rdd.RDD[String]  
    val words=lines.flatMap(line => line.split(" "))  
                      ^  
  
scala> val words=lines.flatMap(line => line.split(" "))  
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:23  
  
scala> val wordParts = words.map(word => (word,1))  
wordParts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:23  
  
scala> val wordcount = wordParts.reduceByKey(_+_)  
wordcount: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23  
  
scala> val freq = wordcount.filter {case (word,count) => count > 4}  
freq: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:23  
  
scala> freq.collect().foreach(println)  
(spark,5)  
(code,5)  
  
scala>
```