package org.itc.com  
import org.apache.spark.SparkConf  
import org.apache.spark.SparkContext  
  
object Main {  
 def main(args: Array[String]): Unit = {  
 val sparkconf = new SparkConf()  
 sparkconf.set("spark.app.name","firstdemo")  
  
 val sc = new SparkContext(sparkconf)// (master= "yarn",appName= "aaa")  
  
 val rdd1 = sc.textFile(args(0))  
 val rdd2 = rdd1.flatMap(x => x.split(" "))  
 val rdd3 = rdd2.map(x => (x.toLowerCase(),1))  
  
 //val rdd4 = rdd3.reduceByKey((x,y)=> x+y).sortByKey() //sort by key  
 // Tuple(abc,2) for tuple index starts from 1 (1 - abc, 2- 2)  
 // sortBy(\_.\_1) sort by first column , sortBy(\_.\_2) sort by second column  
 val rdd4 = rdd3.reduceByKey((x,y)=> x+y).sortBy(\_.\_2) // sort by value  
  
  
 // val rdd5 = rdd4.sortByKey()  
 // outOfMemory Exception  
 // val result = rdd4.collect()  
  
 rdd4.collect().foreach(*println*)  
 rdd4.saveAsTextFile(args(1))  
  
 //spark-submit --class org.itc.com.Main --master yarn sparkhadoopconnection.jar UKUSMarHDFS/bharathi/data.txt  
 //spark-submit --class org.itc.com.Main --master yarn sparkhadoopconnection.jar UKUSMarHDFS/bharathi/data.txt UKUSMarHDFS/bharathi/output (new directory)  
 }  
}

spark-submit --class org.itc.com.Main --master yarn sparkhadoopconnection.jar UKUSMarHDFS/bharathi/data.txt UKUSMarHDFS/bharathi/output

UKUSMarHDFS/bharathi/output - output is created automatically (if you specify already existing directory, it will throw error file already exist)

\*\*\*\*\*Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://ip-172-31-3-80.eu-west-2.compute.internal:8020/user/ec2-user/UKUSMarHDFS/bharathi/output already exists \*\*\*\*\*\*\*\*\*\*



rdd4.coalesce(1).saveAsTextFile(args(1))

spark-submit --class org.itc.com.Main --master yarn sparkhadoopconnection.jar UKUSMarHDFS/bharathi/data.txt UKUSMarHDFS/bharathi/output2

// now the output is stored in single file by using coalesce(1)

