* Launch Spark with the command
  + $ spark-shell
* Create an RDD
  + scala> val inputfile = sc.textFile("input.txt")
* Execute Word count Transformation
  + scala> val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_+\_);
  + flatMap(line ⇒ line.split(“ ”)): Split each line into Words.
  + (map(word ⇒ (word, 1)): Read each word as a key with a value ‘1’ using map function.
  + (reduceByKey(\_+\_)): Reduce those keys by adding values of similar keys.

Solution :

$ spark-shell

scala> val inputfile = sc.textFile("input.txt")

//hdfs://localhost:9900/user/zdina/test/text.txt

scala> val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_+\_);

scala> counts.toDebugString (description about current RDD )

scala> counts.cache() (mark an RDD to be persisted, it will be kept in memory on the nodes)

scala> counts.saveAsTextFile("output")

Scala> counts.unpersist()