Program No:	
Roll No :	**1545**
Title of Program :	**Spark**
Objective :	**Map reduce**

Output:

```
scala> val data = Seq("good morning", "good afternoon", "good evening", "good night")
data: Seq[String] = List(good morning, good afternoon, good evening, good night)
```

```
scala> var myRdd = sc.parallelize(data)
myRdd: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[0] at parallelize at <console>:29

scala> myRdd.collect.foreach(println);
[Stage 0:>                                                          (0 + 0) / 2]25/09/22 12:03:10 W
 Initial job has not accepted any resources; check your cluster UI to ensure that workers are regis
resources
good morning
good afternoon
good evening
good night
```

```
scala> myRdd.take(2).foreach(println);
good morning
good afternoon

scala> myRdd.toDF().show;
+--------------+
|            _1|
+--------------+
|  good morning|
|good afternoon|
|  good evening|
|    good night|
+--------------+
```

```
scala> var wordsRdd = myRdd.flatMap(w => w.split(" "))
wordsRdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[3] at flatMap at <console>:31

scala> wordsRdd.collect.foreach(println);
good
morning
good
afternoon
good
evening
good
night
```

```
scala> val mapRdd = wordsRdd.map(word => (word, 1))
mapRdd: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[4] at map at <console>:33

scala> mapRdd.collect.foreach(println);
(good,1)
(morning,1)
(good,1)
(afternoon,1)
(good,1)
(evening,1)
(good,1)
(night,1)
```

```
scala> val reduceRdd = mapRdd.reduceByKey(_+_)
reduceRdd: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[5] at reduceByKey at <console>:35

scala> reduceRdd.collect.foreach(println);
(evening,1)
(afternoon,1)
(night,1)
(morning,1)
(good,4)
```

```
scala> val txtRdd = sc.textFile("/user/cloudera/inp.csv")
txtRdd: org.apache.spark.rdd.RDD[String] = /user/cloudera/inp.csv MapPartitionsRDD[9] at textFile at <console>:27

scala> txtRdd.collect.foreach(println);
car bear car river
river car river
car river bear bear
river bear car
```

```
scala> val reducerRdd = mapperRdd.reduceByKey(_+_)
reducerRdd: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[12] at reduceByKey at <console>:33

scala> reducerRdd.collect.foreach(println);
(bear,4)
(car,5)
(river,5)
```

```
scala> val wordRdd = txtRdd.flatMap(word => word.split(" "))
wordRdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[10] at flatMap at <console>:29

scala> wordRdd.collect.foreach(println);
car
bear
car
river
river
car
river
car
river
bear
bear
river
bear
car
```

```
scala> val mapperRdd = wordRdd.map(word => (word, 1))
mapperRdd: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[11] at map at <console>:31

scala> mapperRdd.collect.foreach(println);
(car,1)
(bear,1)
(car,1)
(river,1)
(river,1)
(car,1)
(river,1)
(car,1)
(river,1)
(bear,1)
(bear,1)
(river,1)
(bear,1)
(car,1)
```