WEEK 7 BDA LAB
USN:1BM19CS109
NAME: P PREM SAI

Demonstrate four transformations and actions using spark

1. **TRANSFORMATIONS:**
   i) cartesian product

```
Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val input1=sc.parallelize(List(1,2,3,4))
input1: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize
 at <console>:23

scala> val input2=sc.parallelize(List(4,6,7))
input2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize
 at <console>:23

scala> val cartesianprod=input1.cartesian(input2)
cartesianprod: org.apache.spark.rdd.RDD[(Int, Int)] = CartesianRDD[2] at cartes
ian at <console>:24

scala> println(cartesianprod)
CartesianRDD[2] at cartesian at <console>:24

scala> println(cartesianprod.collect())
[Stage 0:>                                                          (0 + 0) / 9
[Stage 0:>                                                          (0 + 3) / 9

[Lscala.Tuple2;@3269e790

scala> cartesianprod.collect()
res2: Array[(Int, Int)] = Array((1,4), (1,6), (1,7), (2,4), (2,6), (2,7), (3,4)
, (4,4), (3,6), (4,6), (3,7), (4,7))
```

   ii) distinct

```
scala> val testinput=sc.parallelize(List(1,2,2,3,3))
testinput: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[3]
ize at <console>:23

scala> val distinctoutput=testinput.distinct()
distinctoutput: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[6]
 at <console>:23

scala> distinctoutput.collect()
res4: Array[Int] = Array(3, 1, 2)
```

   iii) filter

```
scala> val input=sc.textFile("/home/hdoop/Desktop/a.txt")
input: org.apache.spark.rdd.RDD[String] = /home/hdoop/Desktop/a.txt MapPartitio
nsRDD[8] at textFile at <console>:23

scala> input.filter(word=>word.contains("is"))
res5: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[9] at filter at <cons
ole>:24

scala> val occurence=input.filter(word=>word.contains("is"))
occurence: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[10] at filter at
 <console>:23

scala> occurence.collect()
[Stage 5:>                                                          (0 + 0) / 2

res6: Array[String] = Array(this is a test the quick brown fox)
```

iv) flatmap

```
scala> occurence.collect()
[Stage 6:>                                                          (0 + 2) / 2

res7: Array[String] = Array(this is a test the quick brown fox)

scala> val input1=sc.parallelize(List("This is","a test","for scala"))
input1: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[12] at paralle
lize at <console>:23

scala> var split_input=input1.flatMap(word=>word.split(" "))
split_input: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[13] at flatMap
 at <console>:23

scala> spit_input.first()
<console>:23: error: not found: value spit_input
       spit_input.first()
       ^

scala> split_input.first()
res9: String = This
```

v) intersection

```
scala> split_input.first()
res9: String = This

scala> val input1=sc.parallelize(List(1,2,3,4))
input1: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[14]
e at <console>:23

scala> val input2=sc.parallelize(List(2,5,6,4))
input2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[15]
e at <console>:23

scala> var inter_section=input1.intersection(input2)
inter_section: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[21
tion at <console>:24

scala> println(inter_section.mkString(","))
<console>:24: error: value mkString is not a member of org.apache.
[Int]
       println(inter_section.mkString(","))
                             ^

scala> println(inter_section.collect().mkString(","))
[Stage 8:>                  (0 + 3) / 3][Stage 9:>
[Stage 9:>

4,2
```

2. **Actions(count,take,top,takeordered):**

```
scala> val op1=ip1.count
op1: Long = 4

scala> val op1=ip1.take(2)
op1: Array[Int] = Array(1, 3)

scala> val op1=ip1.top(2)
op1: Array[Int] = Array(6, 5)

scala> val op1=ip1.takeOrdered(3)
op1: Array[Int] = Array(1, 3, 5)

scala> val op1=ip1.takeOrdered(3)(Ordering[Int].reverse)
op1: Array[Int] = Array(6, 5, 3)
```