Problem Statement

1) What is the distribution of the total number of air-travelers per year
2) What is the total air distance covered by each user per year
3) Which user has travelled the largest distance till date
4) What is the most preferred destination for all users.

Question 1 Solutions:

```
scala> val baseRDD = sc.textFile("/home/acadgild/Assignment-18/S18_Dataset_Holidays.txt")
baseRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/Assignment-18/S18_Dataset_Holidays.txt MapPartitionsRDD[4] at textFile at <con
sole>:24

scala> import org.apache.spark.storage.StorageLevel
import org.apache.spark.storage.StorageLevel

scala> baseRDD.persist(StorageLevel.MEMORY_ONLY)
res0: baseRDD.type = /home/acadgild/Assignment-18/S18_Dataset_Holidays.txt MapPartitionsRDD[4] at textFile at <console>:24

scala> val splitRDD = baseRDD.map(x => (x.split(",")(5).toInt,1))
splitRDD: org.apache.spark.rdd.RDD[(Int, Int)] = MapPartitionsRDD[5] at map at <console>:27

scala> val countSplit = splitRDD.reduceByKey((x,y) => (x + y))
countSplit: org.apache.spark.rdd.RDD[(Int, Int)] = ShuffledRDD[6] at reduceByKey at <console>:29

scala> countSplit.foreach(println)
(1994,1)
(1992,7)
(1990,8)
(1991,9)
(1993,7)

scala>
```

Question 2 Solutions:

```
scala> val splitRDD = baseRDD.map(x => ((x.split(",")(0),x.split(",")(5)),x.split(",")(4).toInt))
splitRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[7] at map at <console>:27

scala> val distRDD = splitRDD.reduceByKey((x,y) => (x + y))
distRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = ShuffledRDD[8] at reduceByKey at <console>:29
```

```
scala> distRDD.foreach(println)
((3,1992),200)
((3,1993),200)
((5,1991),200)
((6,1991),400)
((10,1993),200)
((5,1992),400)
((8,1991),200)
((8,1990),200)
((1,1993),600)
((5,1994),200)
((2,1993),200)
((2,1991),400)
((4,1990),400)
((10,1992),200)
((3,1991),200)
((1,1990),200)
((10,1990),200)
((6,1993),200)
((9,1992),400)
((8,1992),200)
((7,1990),600)
((9,1991),200)
((4,1991),200)

scala>
```

Question 3 Solutions:

```
scala> val userRDD = baseRDD.map(x=> (x.split(",")(0),x.split(",")(4).toInt))
userRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at map at <console>:27

scala> val totaldistRDD = userRDD.reduceByKey((x,y) => (x+y))
totaldistRDD: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[10] at reduceByKey at <console>:29
```

```
scala> val maxRDD = totaldistRDD.takeOrdered(1)
maxRDD: Array[(String, Int)] = Array((1,800))
```

Question 4 Solutions:

```
scala> val destRDD = baseRDD.map(x => (x.split(",")(2),1))
destRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[12] at map at <console>:27

scala> val destreduceRDD = destRDD.reduceByKey((x,y) => (x + y))
destreduceRDD: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[13] at reduceByKey at <console>:29

scala> val maxRDD = destreduceRDD.takeOrdered(1)(Ordering[Int].reverse.on(_._2))
maxRDD: Array[(String, Int)] = Array((IND,9))

scala>
```


Submitted By Shishir