BDA Lab – Scala

1. Execute any four transformations and four actions

```
scala> sc.parallelize(1 to 10 by 2)
res1: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:25

scala> val lfunc = (x:Int)=> x=x
<console>:23: error: reassignment to val
       val lfunc = (x:Int)=> x=x
                             ^

scala> val lfunc = (x:Int)=> x+x
lfunc: Int => Int = $Lambda$2041/1984286422@700e3b73

scala> lfunc(5)
res2: Int = 10
```

```
scala> val dataFile = sc.textFile("C:/Users/admin/Desktop/cybersecurity.txt")
dataFile: org.apache.spark.rdd.RDD[String] = C:/Users/admin/Desktop/cybersecurity.txt MapPartitionsRDD[4] at textFile at <console>:24

scala> dataFile.map(x => x.length).collect()
res10: Array[Int] = Array(260, 502, 0)

scala> dataFile.map(x => x.toUpperCase()).collect()
res11: Array[String] = Array(WE WILL LEARN KEY TERMS AND ROLES IN CYBERSECURITY. WE WILL ALSO UNDERSTAND DIFFERENT TYPES OF ATTACKS AND THEIR IMPACT
 ON AN ORGANIZATION AND INDIVIDUALS AND WE WILL ALSO LEARN ABOUT TOOLS THAT ARE AVAILABLE TO US TO ASSIST IN ANY CYBERSECURITY INVESTIGATION., THE C
IA TRIAD WILL BE FURTHER EXPLAINED. WE WILL ALSO BEGIN TO LEARN THE SIGNIFICANCE OF INCIDENT RESPONSE AND FRAMEWORKS AROUND CYBERSECURITY. FINALLY,
WE WILL GET AN OVERVIEW OF IT GOVERNANCE BEST PRACTICES AND COMPLIANCE. WE WILL BE INTRODUCED TO KEY SECURITY TOOLS INCLUDING FIREWALLS, ANTI-VIRUS
AND CRYPTOGRAPHY. WE WILL EXPLORE PENETRATION TESTING AND DIGITAL FORENSICS. WE WILL LEARN WHERE WE CAN GET RESOURCES ON INDUSTRY AND CURRENT THREAT
S TO ASSIST IN FURTHER RESEARCH AROUND CYBERSECURITY., "")

scala> dataFile.map(x => x.toLowerCase()).collect()
res12: Array[String] = Array(we will learn key terms and roles in cybersecurity. we will also understand different types of attacks and their impact
 on an organization and individuals and we will also learn about tools that are available to us to assist in any cybersecurity investigation., the c
ia triad will be further explained. we will also begin to learn the significance of incident response and frameworks around cybersecurity. finally,
we will get an overview of it governance best practices and compliance. we will be introduced to key security tools including firewalls, anti-virus
and cryptography. we will explore penetration testing and digital forensics. we will learn where we can get resources on industry and current threat
s to assist in further research around cybersecurity., "")

scala>
```



```
scala> val rdd = sc.parallelize(Seq("Where is Mount Everest","Himalaya India"))
rdd: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[15] at parallelize at <console>:24

scala> rdd.collect
res13: Array[String] = Array(Where is Mount Everest, Himalaya India)

scala> rdd.map(x => x.split(" ")).collect
res14: Array[Array[String]] = Array(Array(Where, is, Mount, Everest), Array(Himalaya, India))

scala> rdd.flatmap(x => x.split(" ")).collect
<console>:26: error: value flatmap is not a member of org.apache.spark.rdd.RDD[String]
       rdd.flatmap(x => x.split(" ")).collect
           ^

scala> rdd.flatMap(x => x.split(" ")).collect
res16: Array[String] = Array(Where, is, Mount, Everest, Himalaya, India)

scala> rdd.flatMap(x => x.split(" ")).count()
res17: Long = 6

scala> rdd.map(x => x.split(" ")).count()
res18: Long = 2
```

```
scala> rdd.collect
res19: Array[String] = Array(Where is Mount Everest, Himalaya India)

scala> rdd.filter(x => x.contins("Himalaya")).collect
<console>:26: error: value contins is not a member of String
       rdd.filter(x => x.contins("Himalaya")).collect
                         ^

scala> rdd.filter(x => x.contains("Himalaya")).collect
res21: Array[String] = Array(Himalaya India)

scala> rdd.filter(x => x.contains("himalaya")).collect
res22: Array[String] = Array()
```

```
scala> sc.parallelize(1 to 15).filter(x => x % 2 == 0).collect
res25: Array[Int] = Array(2, 4, 6, 8, 10, 12, 14)

scala> sc.parallelize(1 to 15).filter(_ % 5 == 0).collect
res26: Array[Int] = Array(5, 10, 15)
```

```
scala> val rdd1 = sc.parallelize(List("swathi","chandana","anu","swathi"))
rdd1: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[39] at parallelize at <console>:24

scala> val rdd2 = sc.parallelize(List("anugna","anusha","anaya"))
rdd2: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[40] at parallelize at <console>:24

scala> rdd1.union(rdd2).collect
res36: Array[String] = Array(swathi, chandana, anu, swathi, anugna, anusha, anaya)

scala> rdd1.intersection(rdd2).collect
res37: Array[String] = Array()
```

Program to run wordcount on scala shell

```
scala> val data=sc.textFile("C:/Users/admin/Desktop/cybersecurity.txt")
data: org.apache.spark.rdd.RDD[String] = C:/Users/admin/Desktop/cybersecurity.txt MapPartitionsRDD[55] at textFile at <console>:25

scala> data.collect;
res41: Array[String] = Array(We will learn key terms and roles in cybersecurity. We will also understand different types of attacks and their impact
on an organization and individuals and we will also learn about tools that are available to us to assist in any cybersecurity investigation.)

scala> val splitdata = data.flatMap(line => line.split(" "));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[56] at flatMap at <console>:26

scala> splitdata.collect;
res42: Array[String] = Array(We, will, learn, key, terms, and, roles, in, cybersecurity., We, will, also, understand, different, types, of, attacks,
and, their, impact, on, an, organization, and, individuals, and, we, will, also, learn, about, tools, that, are, available, to, us, to, assist, in, a
ny, cybersecurity, investigation.)

scala> val mapdata = splitdata.map(word => (word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[57] at map at <console>:26

scala> mapdata.collect;
res43: Array[(String, Int)] = Array((We,1), (will,1), (learn,1), (key,1), (terms,1), (and,1), (roles,1), (in,1), (cybersecurity.,1), (We,1), (will,1)
, (also,1), (understand,1), (different,1), (types,1), (of,1), (attacks,1), (and,1), (their,1), (impact,1), (on,1), (an,1), (organization,1), (and,1),
 (individuals,1), (and,1), (we,1), (will,1), (also,1), (learn,1), (about,1), (tools,1), (that,1), (are,1), (available,1), (to,1), (us,1), (to,1), (as
sist,1), (in,1), (any,1), (cybersecurity,1), (investigation.,1))

scala> val reducedata = mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[58] at reduceByKey at <console>:26
```



```
scala> reducedata.collect;
res44: Array[(String, Int)] = Array((us,1), (learn,2), (are,1), (investigation.,1), (understand,1), (impact,1), (their,1), (will,3), (we,1), (individ
uals,1), (We,2), (any,1), (key,1), (about,1), (cybersecurity.,1), (different,1), (types,1), (that,1), (on,1), (attacks,1), (to,2), (available,1), (ro
les,1), (organization,1), (in,2), (cybersecurity,1), (of,1), (tools,1), (also,2), (an,1), (and,4), (assist,1), (terms,1))

scala>
```

Using RDD and FlaMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.

```
scala>  val textFile = sc.textFile("C:/Users/admin/Desktop/cybersecurity.txt")
textFile: org.apache.spark.rdd.RDD[String] = C:/Users/admin/Desktop/cybersecurity.txt MapPartitionsRDD[61] at textFile at <console>:25

scala> val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[64] at reduceByKey at <console>:26

scala> import scala.collection.immutable.ListMap
import scala.collection.immutable.ListMap

scala> val sorted=ListMap(counts.collect.sortWith(_._2 > _._2):_*)
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(and -> 4, will -> 3, learn -> 2, We -> 2, to -> 2, in -> 2, also -> 2, us -> 1, are
-> 1, investigation. -> 1, understand -> 1, impact -> 1, their -> 1, we -> 1, individuals -> 1, any -> 1, key -> 1, about -> 1, cybersecurity. -> 1,
different -> 1, types -> 1, that -> 1, on -> 1, attacks -> 1, available -> 1, roles -> 1, organization -> 1, cybersecurity -> 1, of -> 1, tools -> 1,
 an -> 1, assist -> 1, terms -> 1)

scala> println(sorted)
ListMap(and -> 4, will -> 3, learn -> 2, We -> 2, to -> 2, in -> 2, also -> 2, us -> 1, are -> 1, investigation. -> 1, understand -> 1, impact -> 1,
their -> 1, we -> 1, individuals -> 1, any -> 1, key -> 1, about -> 1, cybersecurity. -> 1, different -> 1, types -> 1, that -> 1, on -> 1, attacks -
> 1, available -> 1, roles -> 1, organization -> 1, cybersecurity -> 1, of -> 1, tools -> 1, an -> 1, assist -> 1, terms -> 1)

scala> for((k,v)<-sorted)
     | {
     |   if(v>4)
     |     {
     |      print(k+",")
     |        print(v)
     |        println()
     |      }
     | }
```