1. Create a simple pairRDD of (1, 2), (3, 4), (3, 6).

Welcome to

\_\_\_\_ \_\_

/ \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_

\_\ \/ \_ \/ \_ `/ \_\_/ '\_/

/\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 1.6.0

/\_/

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0\_131)

Type in expressions to have them evaluated.

Type :help for more information.

17/06/06 06:32:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

17/06/06 06:32:32 WARN util.Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.163.130 instead (on interface eth1)

17/06/06 06:32:32 WARN util.Utils: Set SPARK\_LOCAL\_IP if you need to bind to another address

Spark context available as sc (master = local[\*], app id = local-1496723555586).

SQL context available as sqlContext.

scala> val pair\_rdd\_01 = sc.parallelize(List((1, 2), (3, 4), (3, 6)))

pair\_rdd\_01: org.apache.spark.rdd.RDD[(Int, Int)] = ParallelCollectionRDD[0] at parallelize at <console>:27

2. Transform an RDD of ("a","b","c","d","e") to PairRDD (a,0), (b,1), (c,2), (d,3), (e,4)

Welcome to

\_\_\_\_ \_\_

/ \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_

\_\ \/ \_ \/ \_ `/ \_\_/ '\_/

/\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 1.6.0

/\_/

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0\_131)

Type in expressions to have them evaluated.

Type :help for more information.

17/06/06 19:43:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

17/06/06 19:43:08 WARN util.Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.163.130 instead (on interface eth1)

17/06/06 19:43:08 WARN util.Utils: Set SPARK\_LOCAL\_IP if you need to bind to another address

Spark context available as sc (master = local[\*], app id = local-1496770991131).

SQL context available as sqlContext.

scala> val simple\_rdd\_01 = sc.parallelize(List('a','b','c','d','e'))

simple\_rdd\_01: org.apache.spark.rdd.RDD[Char] = ParallelCollectionRDD[0] at parallelize at <console>:27

scala> val pair\_rdd\_01 = simple\_rdd\_01.map(x=>(x,List('a','b','c','d','e').indexOf(x))

| )

pair\_rdd\_01: org.apache.spark.rdd.RDD[(Char, Int)] = MapPartitionsRDD[1] at map at <console>:29

scala> pair\_rdd\_01.foreach(println)

(a,0)

(b,1)

(c,2)

(d,3)

(e,4)

scala> pair\_rdd\_01.foreach(print)

(a,0)(b,1)(c,2)(d,3)(e,4)