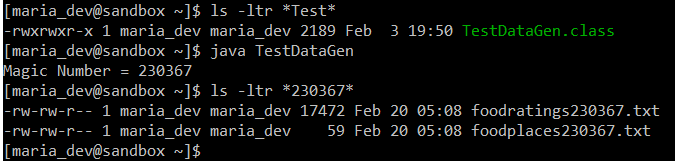
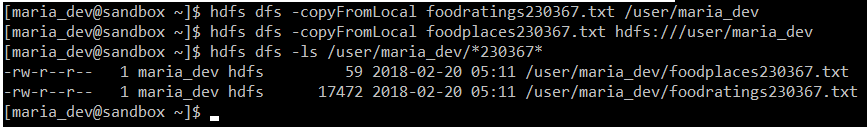
**CS595 - Assignment 6**

* Use the TestDataGen program from previous assignments to generate a new data file.

Copy the file to HDFS.



Magic Number = 230367



**Command Executed:**

java TestDataGen

hdfs dfs -copyFromLocal foodratings230367.txt /user/maria\_dev

hdfs dfs -copyFromLocal foodplaces230367.txt hdfs:///user/maria\_dev

1. Read in the text file into an RDD named ex1RDD.

This RDD should now have records each consisting of a single string having 6 comma separated parts something like the following:

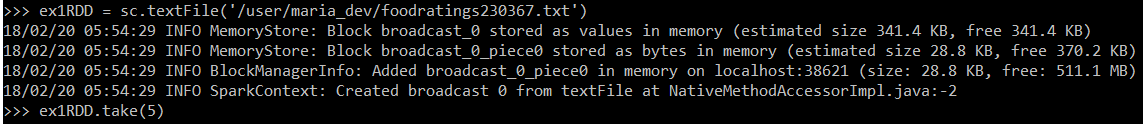
u'Joe,44,33,41,1,5'

u'Mel,13,33,30,50,6'

u'Mel,12,40,30,42,1'

u'Sam,15,28,28,39,2'

List the first five records of the RDD using the “take(5)” action and copy them and the “magic number to your assignment submission for this exercise.





**Command Executed:**

Magic Number = 230367

ex1RDD = sc.textFile('/user/maria\_dev/foodratings230367.txt')

ex1RDD.take(5)

1. Create another RDD called ex2RDD where each record of this new RDD has 6 fields, each a string, by splitting apart each record on “,” boundaries from the ex1RDD.

The records of the new RDD should look something like:

u'Joe', u'44', u'33', u'41', u'1‘, u’5’

u‘Mel', u'13', u'33', u'30', u'50, u’6’‘

u‘Mel', u'12', u'40', u'30', u'42‘, u’1’

u'Sam', u'15', u'28', u'28', u'39‘, u’3’

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.



****

**Command Executed:**

Magic Number = 230367

ex2RDD = ex1RDD.map(lambda line: line.split(","))

ex2RDD.take(5)

1. Create another RDD called ex3RDD from ex2RDD where each record of this new RDD has its third column converted from a string to an integer.

The records of the new RDD should look something like:

u'Joe', u'44', 33, u'41', u'1‘, u’1’

u‘Mel', u'13', 33, u'30', u'50‘, u’2’

u‘Mel', u'12', 40, u'30', u'42‘, u’3’

u'Sam', u'15', 28, u'28', u'39‘, u’4’

Hint: Use a lambda function something like the following:

lambda line : [line[0], line[1], int(line[2]), line[3], line[4], line[5]]

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.





**Command Executed:**

Magic Number = 230367

ex3RDD = ex2RDD.map(lambda line: [line[0], line[1], int(line[2]), line[3], line[4], line[5]])

ex3RDD.take(5)

1. Create another RDD called ex4RDD from ex3RDD where each record of this new RDD is allowed to have a value of < 25 for its third field.

The records of the new RDD should look something like:

u'Joe', u'44', 21, u'41', u'1‘, u’6’

u‘Mel', u'13', 3, u'30', u'50‘, u’1’

u‘Mel', u'12', 4, u'30', u'42‘, u’4’

u'Sam', u'15', 8, u'28', u'39‘, u’5’

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

****

****

**Command Executed:**

Magic Number = 230367

ex4RDD = ex3RDD.filter(lambda line: int(line[2]) < 25)

ex4RDD.take(5)

1. Create another RDD called ex5RDD from ex4RDD where each record is a key value pair where the key is the first field of the record and the value is the entire record.

The records of the new RDD should look something like:

(u’Joe’, (u'Joe', u'44', 21, u'41', u'1‘, u’1’))

(u’Mel’, (u‘Mel', u'13', 3, u'30', u'50‘, u’6’))

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

****

****

**Command Executed:**

Magic Number = 230367

ex5RDD = ex4RDD.map(lambda line: [line[0], line])

ex5RDD.take(5)

1. Create another RDD called ex6RDD from ex5RDD where the records are organized in ascending order by key

The records of the new RDD should look something like:

(u’Joe’, (u'Joe', u'44', 21, u'41', u'1‘, u’4’))

(u’Mel’ , (u‘Mel', u'13', 3, u'30', u'50‘, u’3’))

(u’Sam’ , (u‘Sam', u'23', 3, u'40', u'20‘, u’7’))

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

****

****

****

**Command Executed:**

Magic Number = 230367

ex6RDD = ex5RDD.sortByKey()

ex6RDD.take(5)