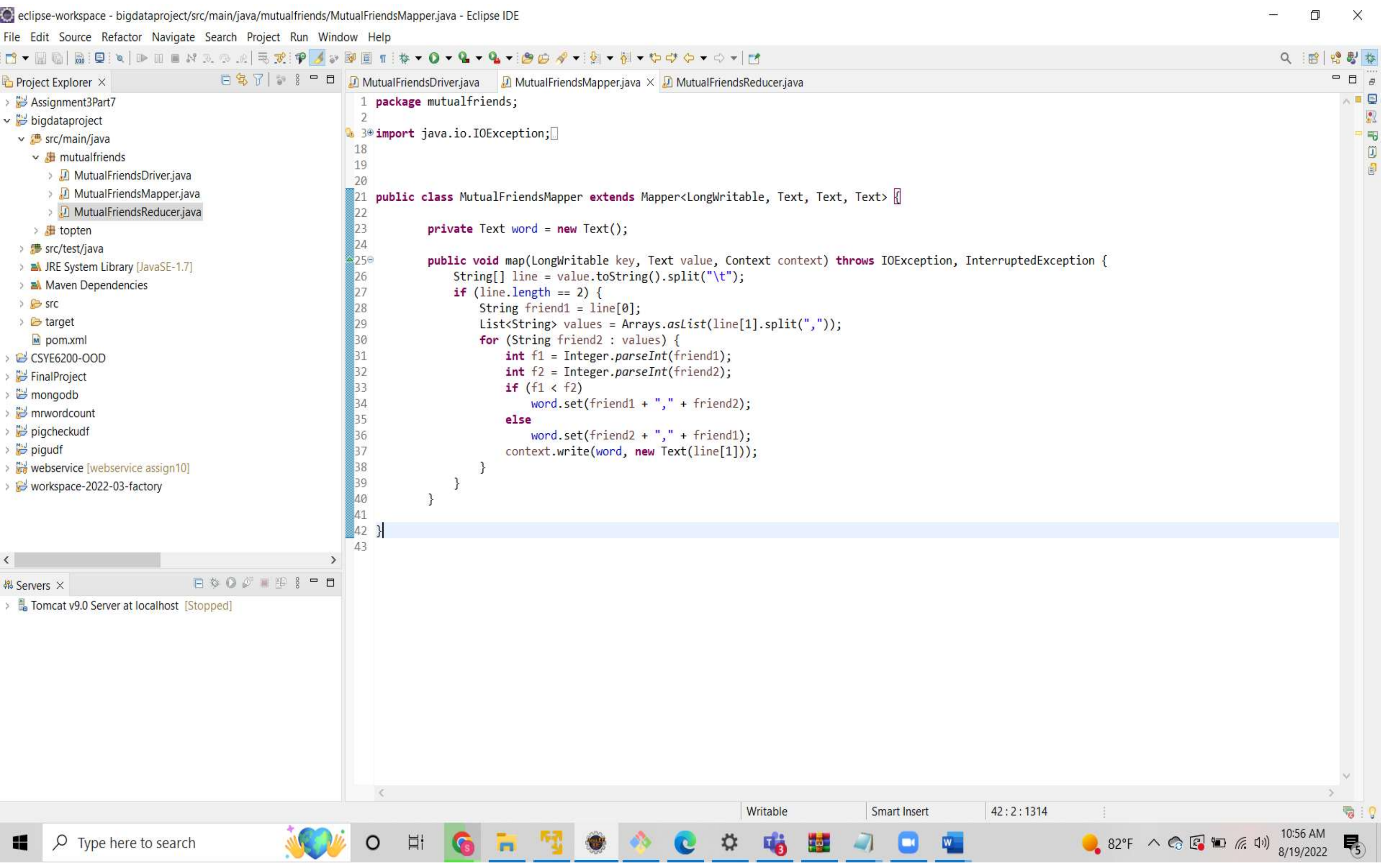
Big Data Final Project

Mutual Friends Map Reduce

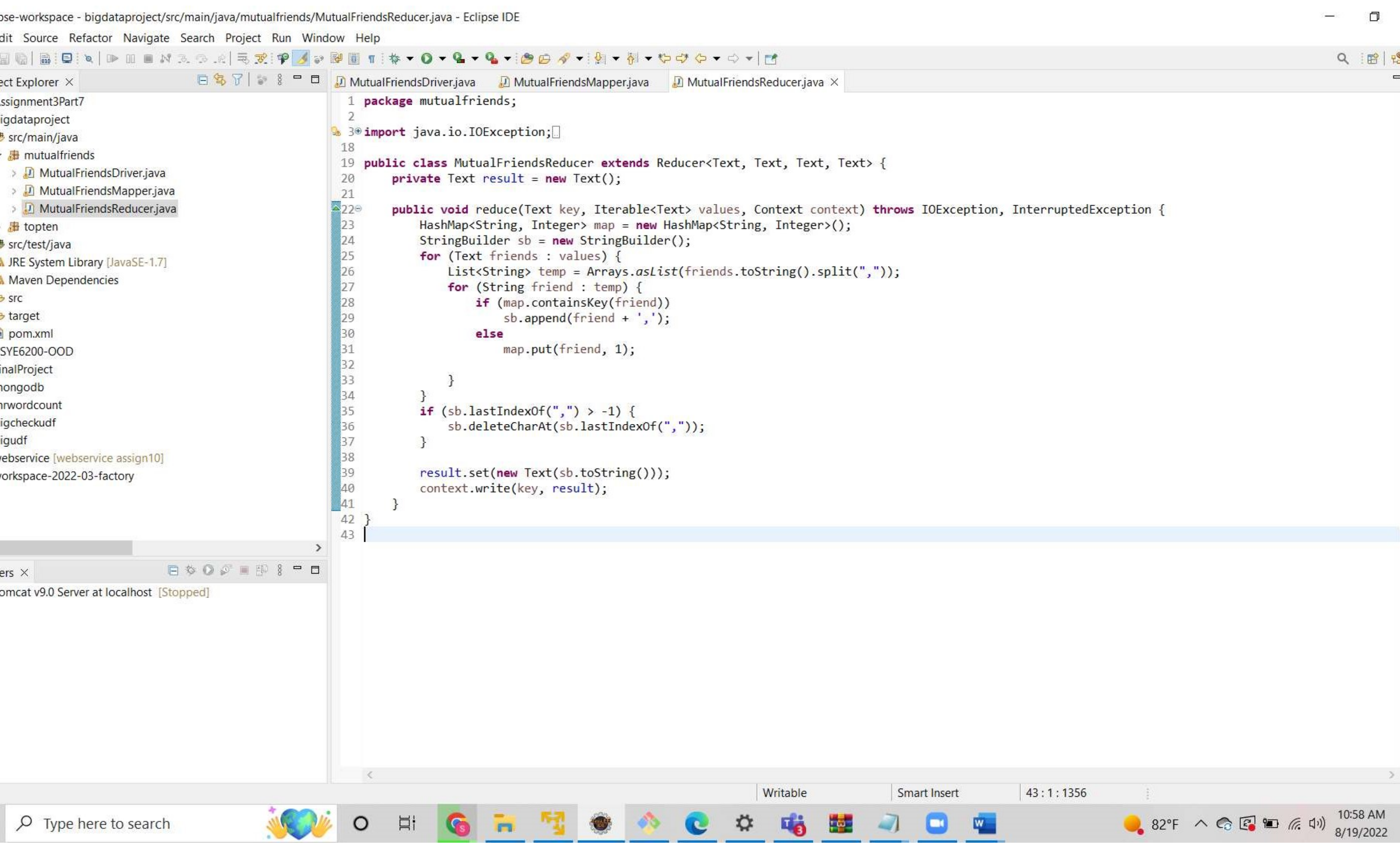
Implementing a simple "Mutual/Common friend list of two friends using MapReduce.

Input: The input contains the adjacency list and has multiple lines in the following format: Here, is a unique integer ID corresponding to a unique user and is a comma-separated list of unique IDs (ID) corresponding to the friends of the user. The friendships are mutual (i.e., edges are undirected): if A is friend with B then B is also friend with A. The data provided is consistent with that rule as there is an explicit entry for each side of each edge. So when you make the pair, always consider (A, B) or (B, A) for user A and B but not both.

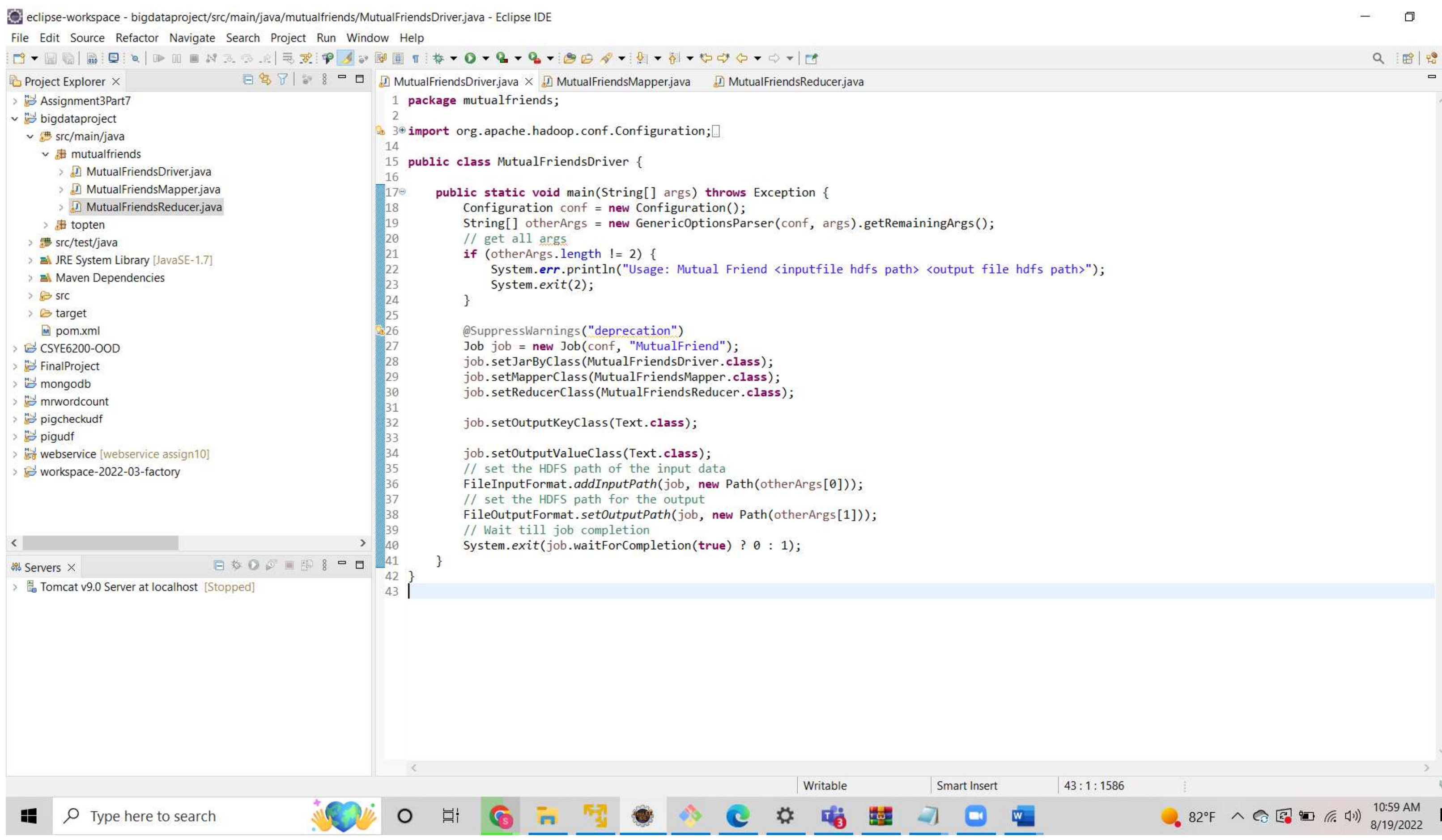
Output: The output should contain one line per user in the following format: <User_A>, <User_B><Mutual/Common Friend List> where <User_A> & <User_B> are unique IDs corresponding to a user A and B (A and B are friend). < Mutual/Common Friend List > is a comma-separated list of unique IDs corresponding to mutual friend list of User A and B.



```java
package mutualfriends;

import java.io.IOException;


public class MutualFriendsMapper extends Mapper<LongWritable, Text, Text, Text> {

	private Text word = new Text();

	public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
		String[] line = value.toString().split("\t");
		if (line.length == 2) {
			String friend1 = line[0];
			List<String> values = Arrays.asList(line[1].split(","));
			for (String friend2 : values) {
				int f1 = Integer.parseInt(friend1);
				int f2 = Integer.parseInt(friend2);
				if (f1 < f2)
					word.set(friend1 + "," + friend2);
				else
					word.set(friend2 + "," + friend1);
				context.write(word, new Text(line[1]));
			}
		}
	}

}
```

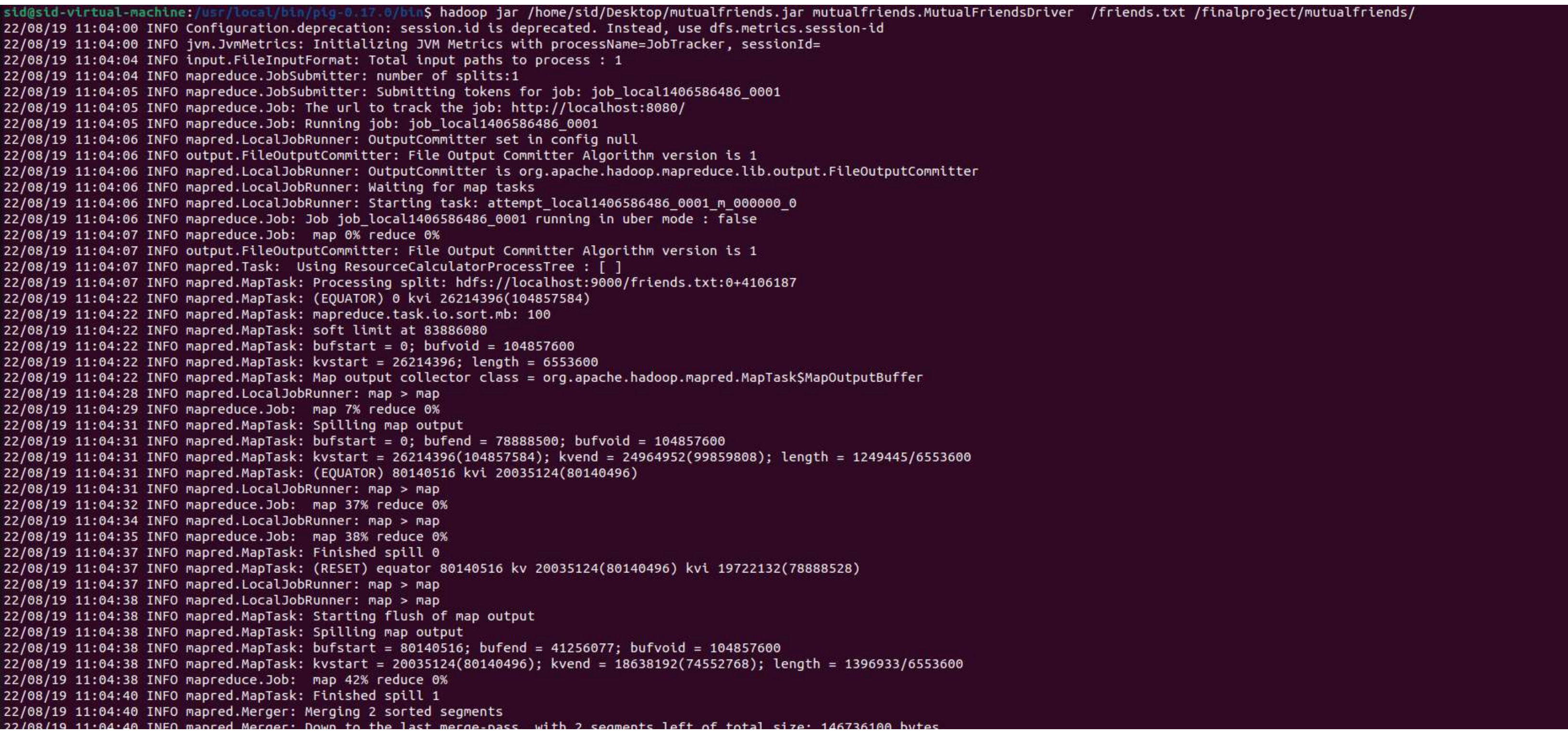

se-workspace - bigdataproject/src/main/java/mutualfriends/MutualFriendsReducer.java - Eclipse IDE
MutualFriendsDriver.java
MutualFriendsMapper.java
MutualFriendsReducer.java
Writable
Smart Insert
43 : 1 : 1356
10:58 AM
8/19/2022

```java
package mutualfriends;

import java.io.IOException;

public class MutualFriendsReducer extends Reducer<Text, Text, Text, Text> {
    private Text result = new Text();

    public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
        HashMap<String, Integer> map = new HashMap<String, Integer>();
        StringBuilder sb = new StringBuilder();
        for (Text friends : values) {
            List<String> temp = Arrays.asList(friends.toString().split(","));
            for (String friend : temp) {
                if (map.containsKey(friend))
                    sb.append(friend + ',');
                else
                    map.put(friend, 1);

            }
        }
        if (sb.lastIndexOf(",") > -1) {
            sb.deleteCharAt(sb.lastIndexOf(","));
        }

        result.set(new Text(sb.toString()));
        context.write(key, result);
    }
}
```

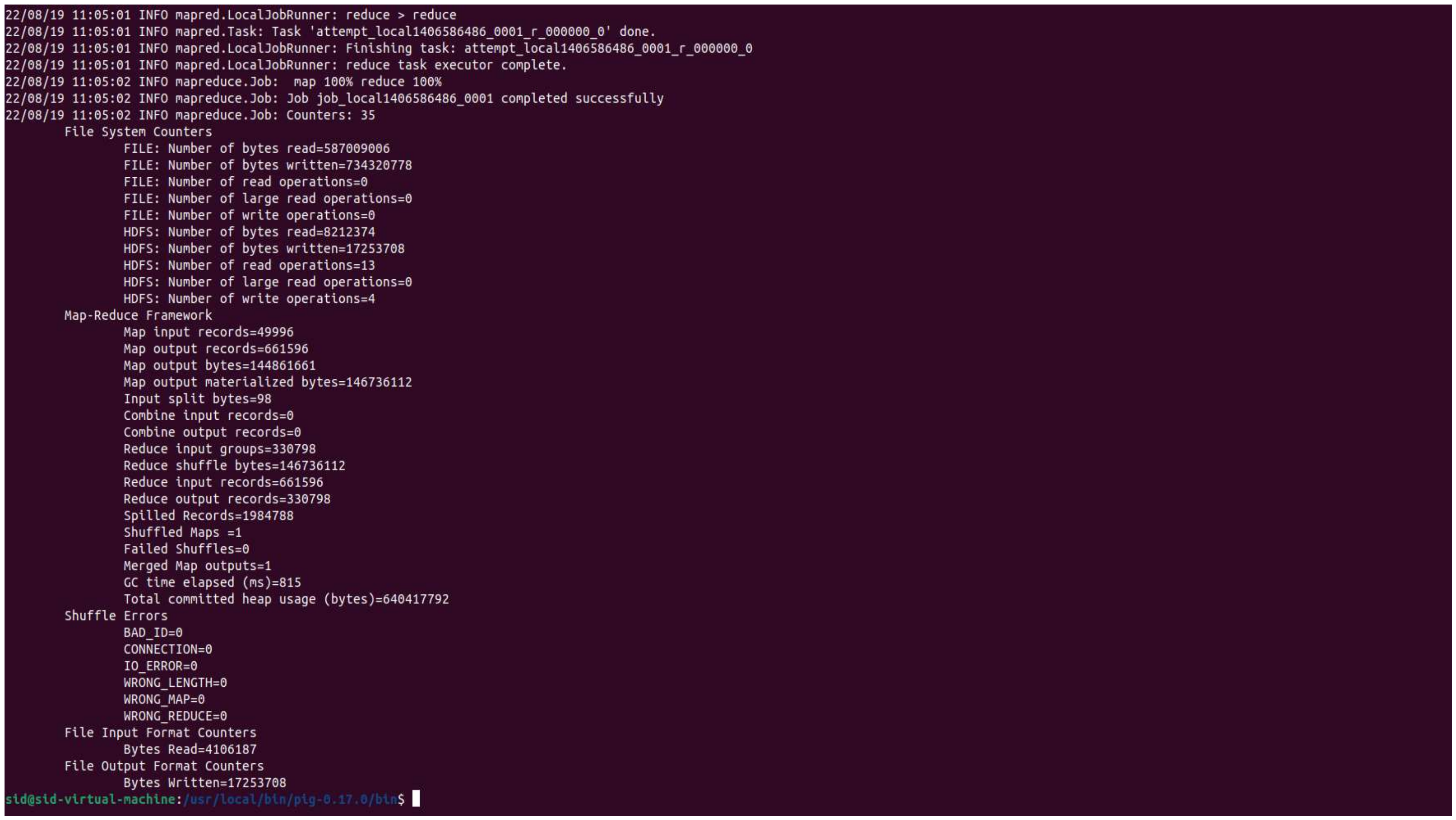

eclipse-workspace - bigdataproject/src/main/java/mutualfriends/MutualFriendsDriver.java - Eclipse IDE
MutualFriendsDriver.java
MutualFriendsMapper.java
MutualFriendsReducer.java
Writable
Smart Insert
43 : 1 : 1586
10:59 AM
8/19/2022

```java
package mutualfriends;

import org.apache.hadoop.conf.Configuration;

public class MutualFriendsDriver {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        // get all args
        if (otherArgs.length != 2) {
            System.err.println("Usage: Mutual Friend <inputfile hdfs path> <output file hdfs path>");
            System.exit(2);
        }

        @SuppressWarnings("deprecation")
        Job job = new Job(conf, "MutualFriend");
        job.setJarByClass(MutualFriendsDriver.class);
        job.setMapperClass(MutualFriendsMapper.class);
        job.setReducerClass(MutualFriendsReducer.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(Text.class);
        // set the HDFS path of the input data
        FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
        // set the HDFS path for the output
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
        // Wait till job completion
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Command to run jar file : hadoop jar /home/sid/Desktop/mutualfriends.jar mutualfriends.MutualFriendsDriver /friends.txt /finalproject/mutualfriends/

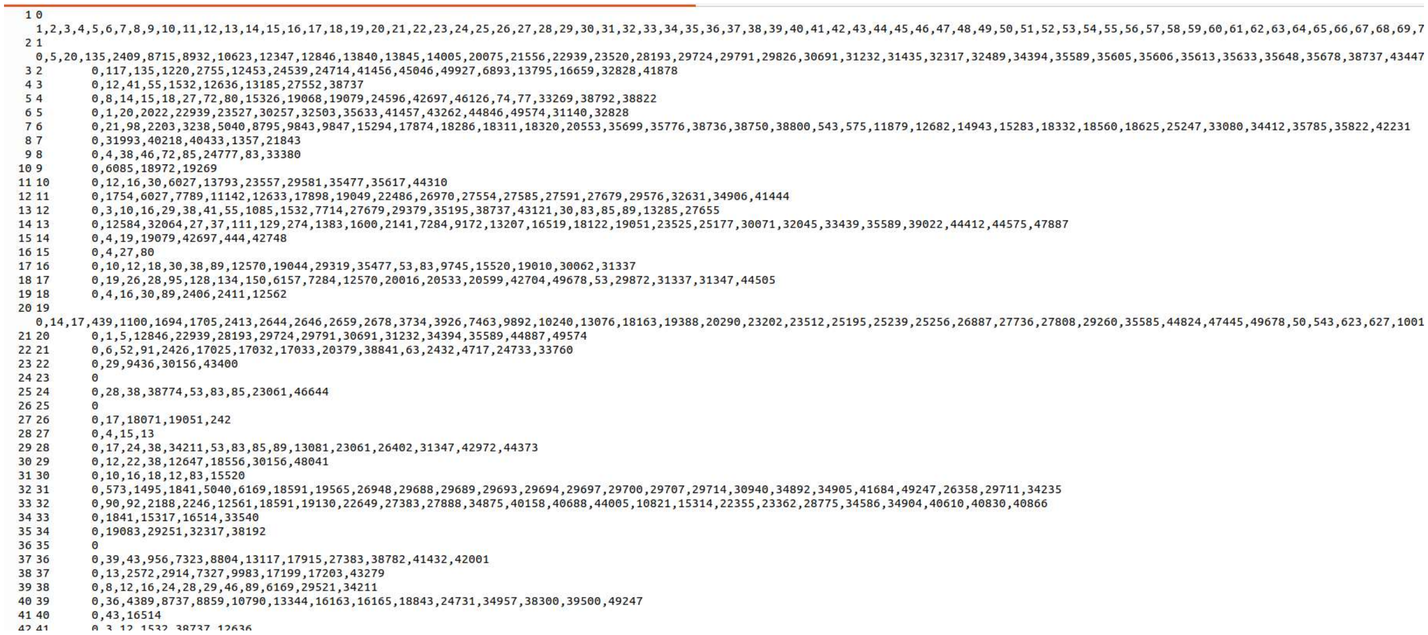


```
sid@sid-virtual-machine:/usr/local/bin/pig-0.17.0/bin$ hadoop jar /home/sid/Desktop/mutualfriends.jar mutualfriends.MutualFriendsDriver  /friends.txt /finalproject/mutualfriends/
22/08/19 11:04:00 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
22/08/19 11:04:00 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
22/08/19 11:04:04 INFO input.FileInputFormat: Total input paths to process : 1
22/08/19 11:04:04 INFO mapreduce.JobSubmitter: number of splits:1
22/08/19 11:04:05 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1406586486_0001
22/08/19 11:04:05 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/08/19 11:04:05 INFO mapreduce.Job: Running job: job_local1406586486_0001
22/08/19 11:04:06 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/08/19 11:04:06 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
22/08/19 11:04:06 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/08/19 11:04:06 INFO mapred.LocalJobRunner: Waiting for map tasks
22/08/19 11:04:06 INFO mapred.LocalJobRunner: Starting task: attempt_local1406586486_0001_m_000000_0
22/08/19 11:04:06 INFO mapreduce.Job: Job job_local1406586486_0001 running in uber mode : false
22/08/19 11:04:07 INFO mapreduce.Job:  map 0% reduce 0%
22/08/19 11:04:07 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
22/08/19 11:04:07 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/08/19 11:04:07 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/friends.txt:0+4106187
22/08/19 11:04:22 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/08/19 11:04:22 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/08/19 11:04:22 INFO mapred.MapTask: soft limit at 83886080
22/08/19 11:04:22 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/08/19 11:04:22 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/08/19 11:04:22 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/08/19 11:04:28 INFO mapred.LocalJobRunner: map > map
22/08/19 11:04:29 INFO mapreduce.Job:  map 7% reduce 0%
22/08/19 11:04:31 INFO mapred.MapTask: Spilling map output
22/08/19 11:04:31 INFO mapred.MapTask: bufstart = 0; bufend = 78888500; bufvoid = 104857600
22/08/19 11:04:31 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 24964952(99859808); length = 1249445/6553600
22/08/19 11:04:31 INFO mapred.MapTask: (EQUATOR) 80140516 kvi 20035124(80140496)
22/08/19 11:04:31 INFO mapred.LocalJobRunner: map > map
22/08/19 11:04:32 INFO mapreduce.Job:  map 37% reduce 0%
22/08/19 11:04:34 INFO mapred.LocalJobRunner: map > map
22/08/19 11:04:35 INFO mapreduce.Job:  map 38% reduce 0%
22/08/19 11:04:37 INFO mapred.MapTask: Finished spill 0
22/08/19 11:04:37 INFO mapred.MapTask: (RESET) equator 80140516 kv 20035124(80140496) kvi 19722132(78888528)
22/08/19 11:04:37 INFO mapred.LocalJobRunner: map > map
22/08/19 11:04:38 INFO mapred.LocalJobRunner: map > map
22/08/19 11:04:38 INFO mapred.MapTask: Starting flush of map output
22/08/19 11:04:38 INFO mapred.MapTask: Spilling map output
22/08/19 11:04:38 INFO mapred.MapTask: bufstart = 80140516; bufend = 41256077; bufvoid = 104857600
22/08/19 11:04:38 INFO mapred.MapTask: kvstart = 20035124(80140496); kvend = 18638192(74552768); length = 1396933/6553600
22/08/19 11:04:38 INFO mapreduce.Job:  map 42% reduce 0%
22/08/19 11:04:40 INFO mapred.MapTask: Finished spill 1
22/08/19 11:04:40 INFO mapred.Merger: Merging 2 sorted segments
22/08/19 11:04:40 INFO mapred.Merger: Down to the last merge-pass, with 2 segments left of total size: 146736100 bytes
```



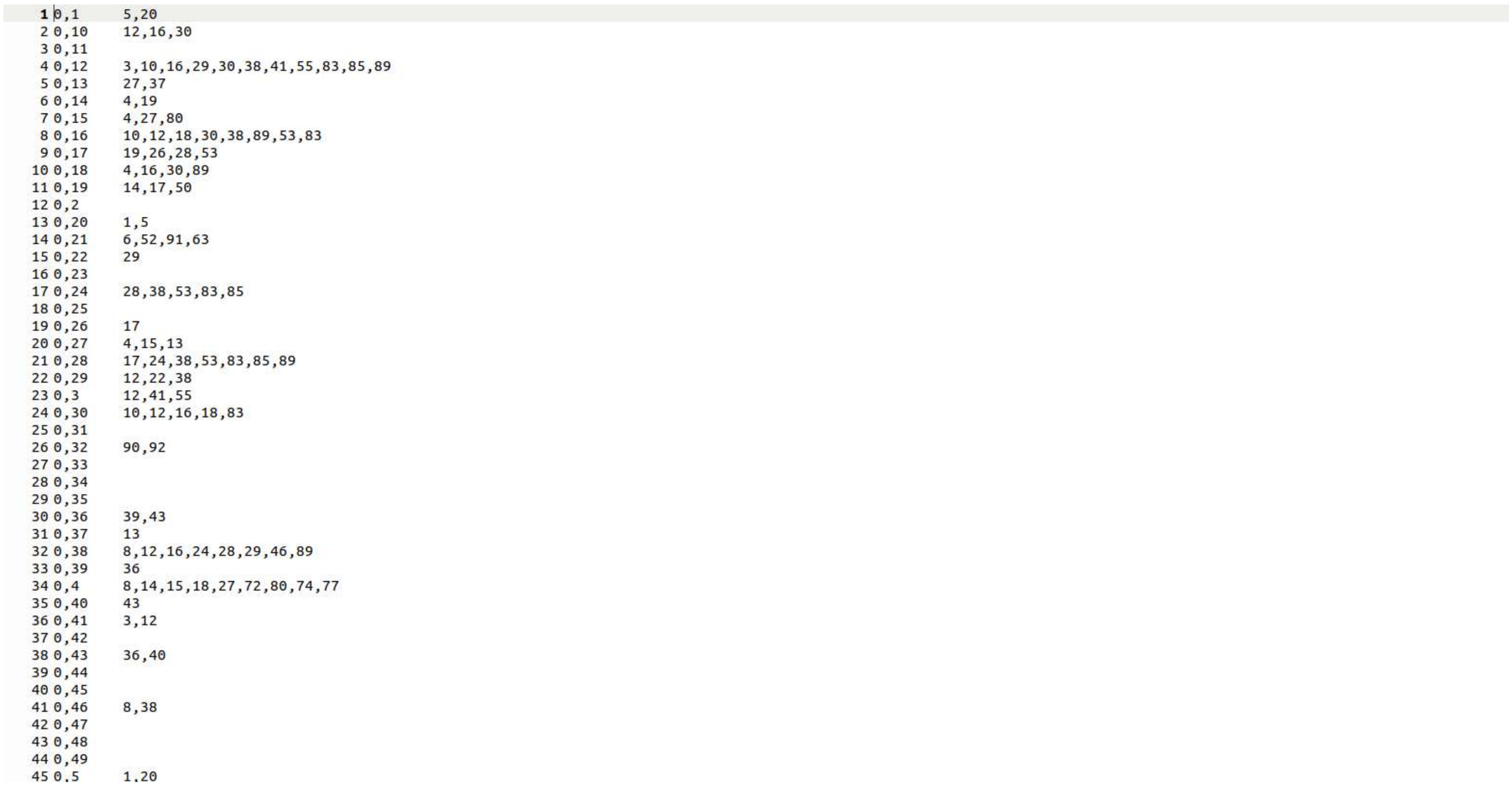
```
22/08/19 11:05:01 INFO mapred.LocalJobRunner: reduce > reduce
22/08/19 11:05:01 INFO mapred.Task: Task 'attempt_local1406586486_0001_r_000000_0' done.
22/08/19 11:05:01 INFO mapred.LocalJobRunner: Finishing task: attempt_local1406586486_0001_r_000000_0
22/08/19 11:05:01 INFO mapred.LocalJobRunner: reduce task executor complete.
22/08/19 11:05:02 INFO mapreduce.Job:  map 100% reduce 100%
22/08/19 11:05:02 INFO mapreduce.Job: Job job_local1406586486_0001 completed successfully
22/08/19 11:05:02 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=587009006
                FILE: Number of bytes written=734320778
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=8212374
                HDFS: Number of bytes written=17253708
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=49996
                Map output records=661596
                Map output bytes=144861661
                Map output materialized bytes=146736112
                Input split bytes=98
                Combine input records=0
                Combine output records=0
                Reduce input groups=330798
                Reduce shuffle bytes=146736112
                Reduce input records=661596
                Reduce output records=330798
                Spilled Records=1984788
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=815
                Total committed heap usage (bytes)=640417792
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=4106187
        File Output Format Counters
                Bytes Written=17253708
sid@sid-virtual-machine:/usr/local/bin/pig-0.17.0/bin$
```

Input:



```
0
 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,7
1
 0,5,20,135,2409,8715,8932,10623,12347,12846,13840,13845,14005,20075,21556,22939,23520,28193,29724,29791,29826,30691,31232,31435,32317,32489,34394,35589,35605,35606,35613,35633,35648,35678,38737,43447
2       0,117,135,1220,2755,12453,24539,24714,41456,45046,49927,6893,13795,16659,32828,41878
3       0,12,41,55,1532,12636,13185,27552,38737
4       0,8,14,15,18,27,72,80,15326,19068,19079,24596,42697,46126,74,77,33269,38792,38822
5       0,1,20,2022,22939,23527,30257,32503,35633,41457,43262,44846,49574,31140,32828
6       0,21,98,2203,3238,5040,8795,9843,9847,15294,17874,18286,18311,18320,20553,35699,35776,38736,38750,38800,543,575,11879,12682,14943,15283,18332,18560,18625,25247,33080,34412,35785,35822,42231
7       0,31993,40218,40433,1357,21843
8       0,4,38,46,72,85,24777,83,33380
9       0,6085,18972,19269
10      0,12,16,30,6027,13793,23557,29581,35477,35617,44310
11      0,1754,6027,7789,11142,12633,17898,19049,22486,26970,27554,27585,27591,27679,29576,32631,34906,41444
12      0,3,10,16,29,38,41,55,1085,1532,7714,27679,29379,35195,38737,43121,30,83,85,89,13285,27655
13      0,12584,32064,27,37,111,129,274,1383,1600,2141,7284,9172,13207,16519,18122,19051,23525,25177,30071,32045,33439,35589,39022,44412,44575,47887
14      0,4,19,19079,42697,444,42748
15      0,4,27,80
16      0,10,12,18,30,38,89,12570,19044,29319,35477,53,83,9745,15520,19010,30062,31337
17      0,19,26,28,95,128,134,150,6157,7284,12570,20016,20533,20599,42704,49678,53,29872,31337,31347,44505
18      0,4,16,30,89,2406,2411,12562
19
 0,14,17,439,1100,1694,1705,2413,2644,2646,2659,2678,3734,3926,7463,9892,10240,13076,18163,19388,20290,23202,23512,25195,25239,25256,26887,27736,27808,29260,35585,44824,47445,49678,50,543,623,627,1001
20      0,1,5,12846,22939,28193,29724,29791,30691,31232,34394,35589,44887,49574
21      0,6,52,91,2426,17025,17032,17033,20379,38841,63,2432,4717,24733,33760
22      0,29,9436,30156,43400
23      0
24      0,28,38,38774,53,83,85,23061,46644
25      0
26      0,17,18071,19051,242
27      0,4,15,13
28      0,17,24,38,34211,53,83,85,89,13081,23061,26402,31347,42972,44373
29      0,12,22,38,12647,18556,30156,48041
30      0,10,16,18,12,83,15520
31      0,573,1495,1841,5040,6169,18591,19565,26948,29688,29689,29693,29694,29697,29700,29707,29714,30940,34892,34905,41684,49247,26358,29711,34235
32      0,90,92,2188,2246,12561,18591,19130,22649,27383,27888,34875,40158,40688,44005,10821,15314,22355,23362,28775,34586,34904,40610,40830,40866
33      0,1841,15317,16514,33540
34      0,19083,29251,32317,38192
35      0
36      0,39,43,956,7323,8804,13117,17915,27383,38782,41432,42001
37      0,13,2572,2914,7327,9983,17199,17203,43279
38      0,8,12,16,24,28,29,46,89,6169,29521,34211
39      0,36,4389,8737,8859,10790,13344,16163,16165,18843,24731,34957,38300,39500,49247
40      0,43,16514
41      0,3,12,1532,38737,12636
```

Output :

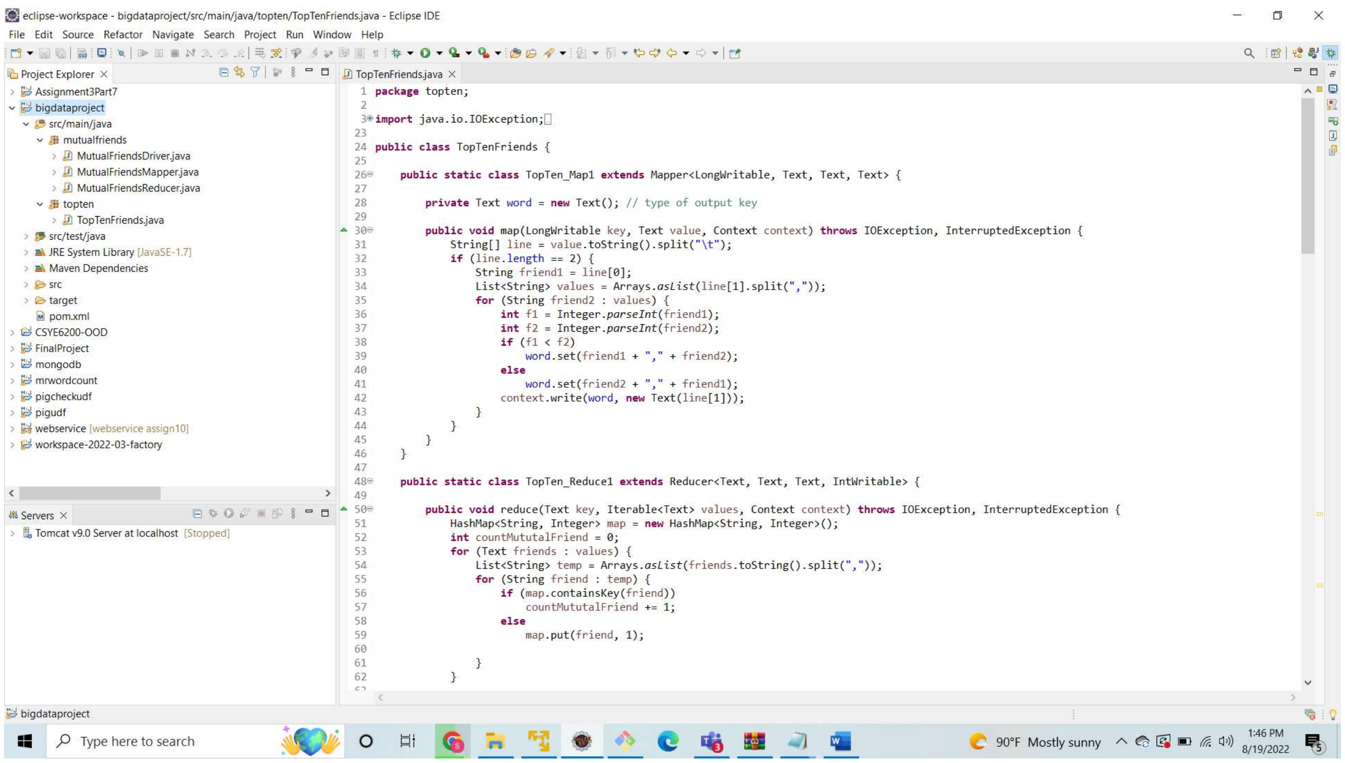


```
0,1     5,20
0,10    12,16,30
0,11
0,12    3,10,16,29,30,38,41,55,83,85,89
0,13    27,37
0,14    4,19
0,15    4,27,80
0,16    10,12,18,30,38,89,53,83
0,17    19,26,28,53
0,18    4,16,30,89
0,19    14,17,50
0,2
0,20    1,5
0,21    6,52,91,63
0,22    29
0,23
0,24    28,38,53,83,85
0,25
0,26    17
0,27    4,15,13
0,28    17,24,38,53,83,85,89
0,29    12,22,38
0,3     12,41,55
0,30    10,12,16,18,83
0,31
0,32    90,92
0,33
0,34
0,35
0,36    39,43
0,37    13
0,38    8,12,16,24,28,29,46,89
0,39    36
0,4     8,14,15,18,27,72,80,74,77
0,40    43
0,41    3,12
0,42
0,43    36,40
0,44
0,45
0,46    8,38
0,47
0,48
0,49
0,5     1,20
```

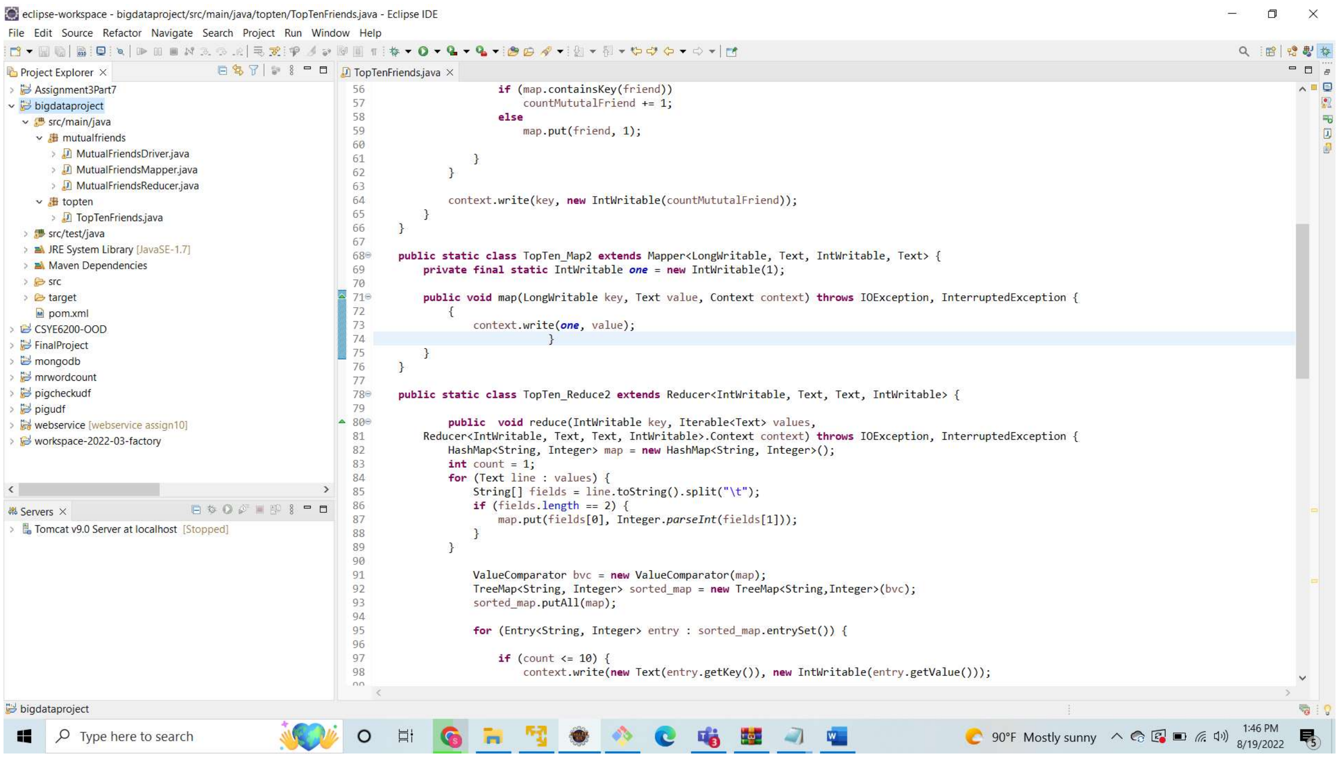
Top Ten Mutual Friends

Friend pairs whose common friend number are within the top-10 in all the pairs printing them in decreasing order .

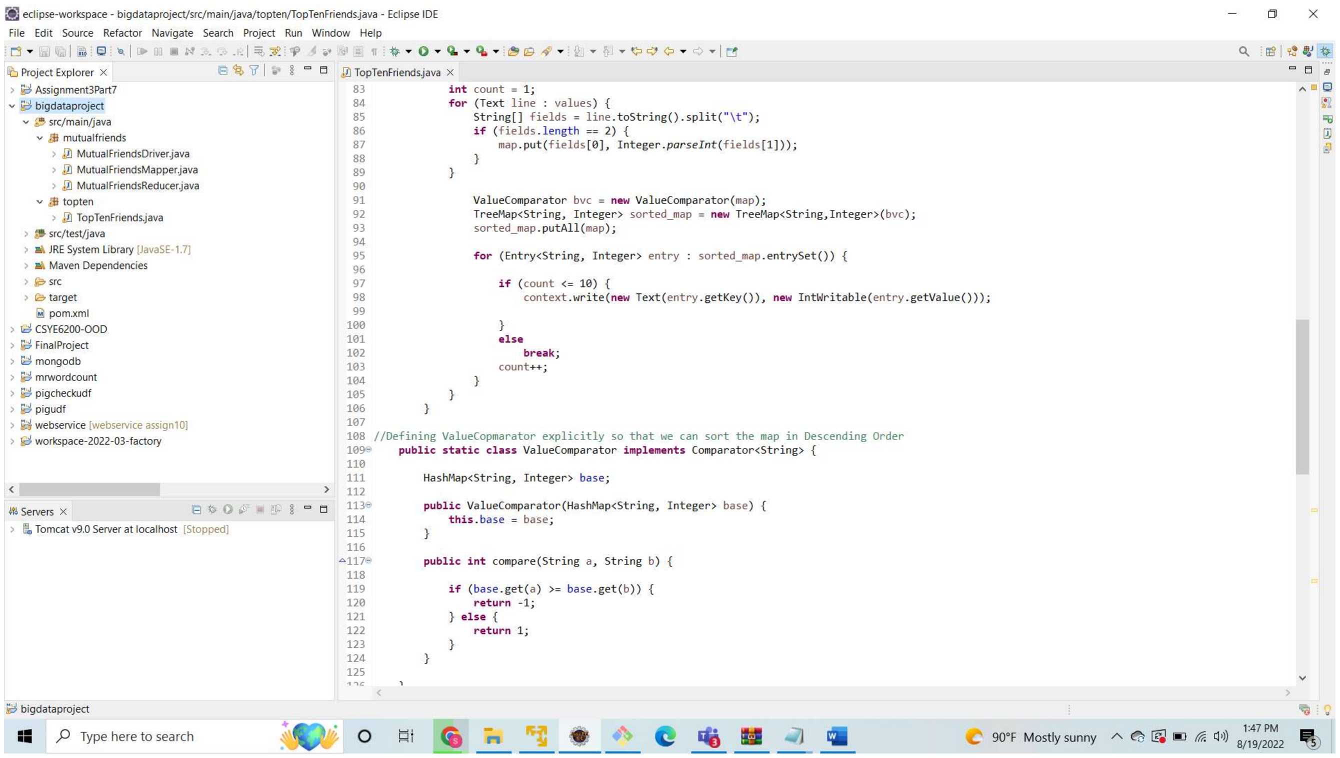
Using 2 Mapper and 2 Reducer class and comparator to sort data



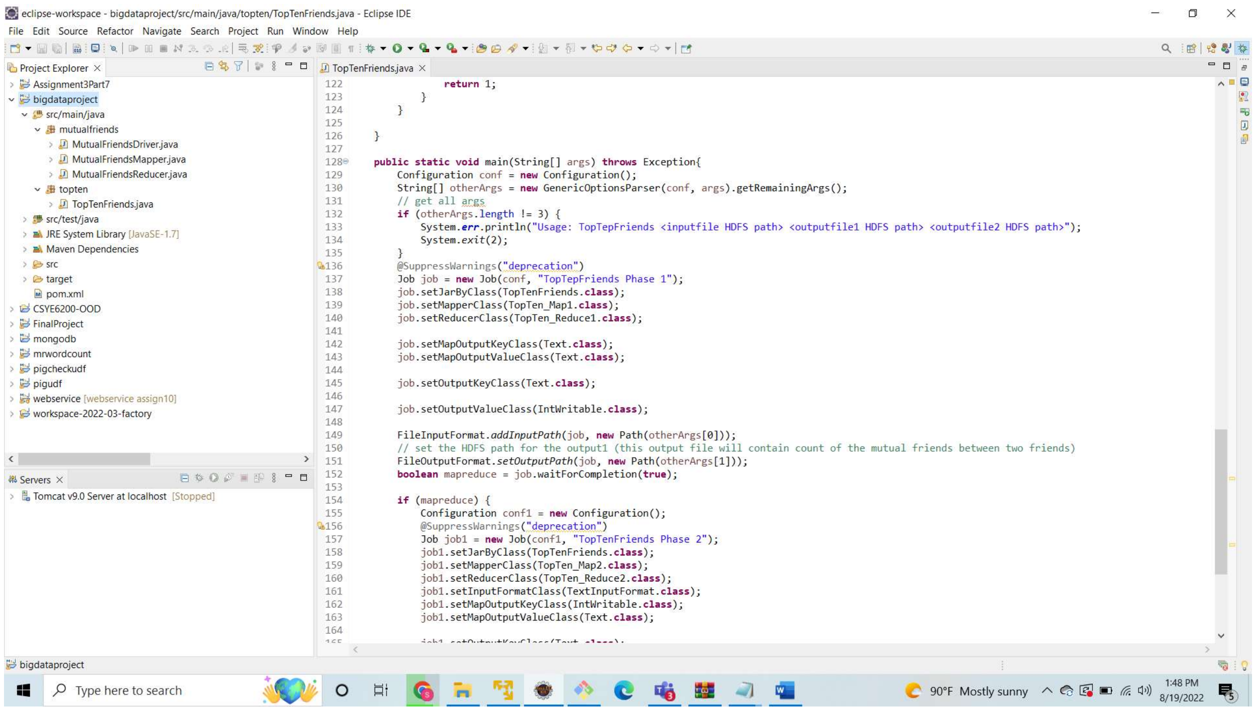
```java
package topten;

import java.io.IOException;

public class TopTenFriends {

    public static class TopTen_Map1 extends Mapper<LongWritable, Text, Text, Text> {

        private Text word = new Text(); // type of output key

        public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
            String[] line = value.toString().split("\t");
            if (line.length == 2) {
                String friend1 = line[0];
                List<String> values = Arrays.asList(line[1].split(","));
                for (String friend2 : values) {
                    int f1 = Integer.parseInt(friend1);
                    int f2 = Integer.parseInt(friend2);
                    if (f1 < f2)
                        word.set(friend1 + "," + friend2);
                    else
                        word.set(friend2 + "," + friend1);
                    context.write(word, new Text(line[1]));
                }
            }
        }
    }

    public static class TopTen_Reduce1 extends Reducer<Text, Text, Text, IntWritable> {

        public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            HashMap<String, Integer> map = new HashMap<String, Integer>();
            int countMututalFriend = 0;
            for (Text friends : values) {
                List<String> temp = Arrays.asList(friends.toString().split(","));
                for (String friend : temp) {
                    if (map.containsKey(friend))
                        countMututalFriend += 1;
                    else
                        map.put(friend, 1);

                }
            }
```



```java
                    if (map.containsKey(friend))
                        countMututalFriend += 1;
                    else
                        map.put(friend, 1);

                }
            }

            context.write(key, new IntWritable(countMututalFriend));
        }
    }

    public static class TopTen_Map2 extends Mapper<LongWritable, Text, IntWritable, Text> {
        private final static IntWritable one = new IntWritable(1);

        public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
            {
                context.write(one, value);
                            }
        }
    }

    public static class TopTen_Reduce2 extends Reducer<IntWritable, Text, Text, IntWritable> {

            public  void reduce(IntWritable key, Iterable<Text> values,
        Reducer<IntWritable, Text, Text, IntWritable>.Context context) throws IOException, InterruptedException {
            HashMap<String, Integer> map = new HashMap<String, Integer>();
            int count = 1;
            for (Text line : values) {
                String[] fields = line.toString().split("\t");
                if (fields.length == 2) {
                    map.put(fields[0], Integer.parseInt(fields[1]));
                }
            }

                ValueComparator bvc = new ValueComparator(map);
                TreeMap<String, Integer> sorted_map = new TreeMap<String,Integer>(bvc);
                sorted_map.putAll(map);

                for (Entry<String, Integer> entry : sorted_map.entrySet()) {

                    if (count <= 10) {
                        context.write(new Text(entry.getKey()), new IntWritable(entry.getValue()));
```


eclipse-workspace - bigdataproject/src/main/java/topten/TopTenFriends.java - Eclipse IDE

```java
            int count = 1;
            for (Text line : values) {
                String[] fields = line.toString().split("\t");
                if (fields.length == 2) {
                    map.put(fields[0], Integer.parseInt(fields[1]));
                }
            }

                ValueComparator bvc = new ValueComparator(map);
                TreeMap<String, Integer> sorted_map = new TreeMap<String,Integer>(bvc);
                sorted_map.putAll(map);

                for (Entry<String, Integer> entry : sorted_map.entrySet()) {

                    if (count <= 10) {
                        context.write(new Text(entry.getKey()), new IntWritable(entry.getValue()));

                    }
                    else
                        break;
                    count++;
                }
            }
        }

//Defining ValueCopmarator explicitly so that we can sort the map in Descending Order
    public static class ValueComparator implements Comparator<String> {

        HashMap<String, Integer> base;

        public ValueComparator(HashMap<String, Integer> base) {
            this.base = base;
        }

        public int compare(String a, String b) {

            if (base.get(a) >= base.get(b)) {
                return -1;
            } else {
                return 1;
            }
        }

```

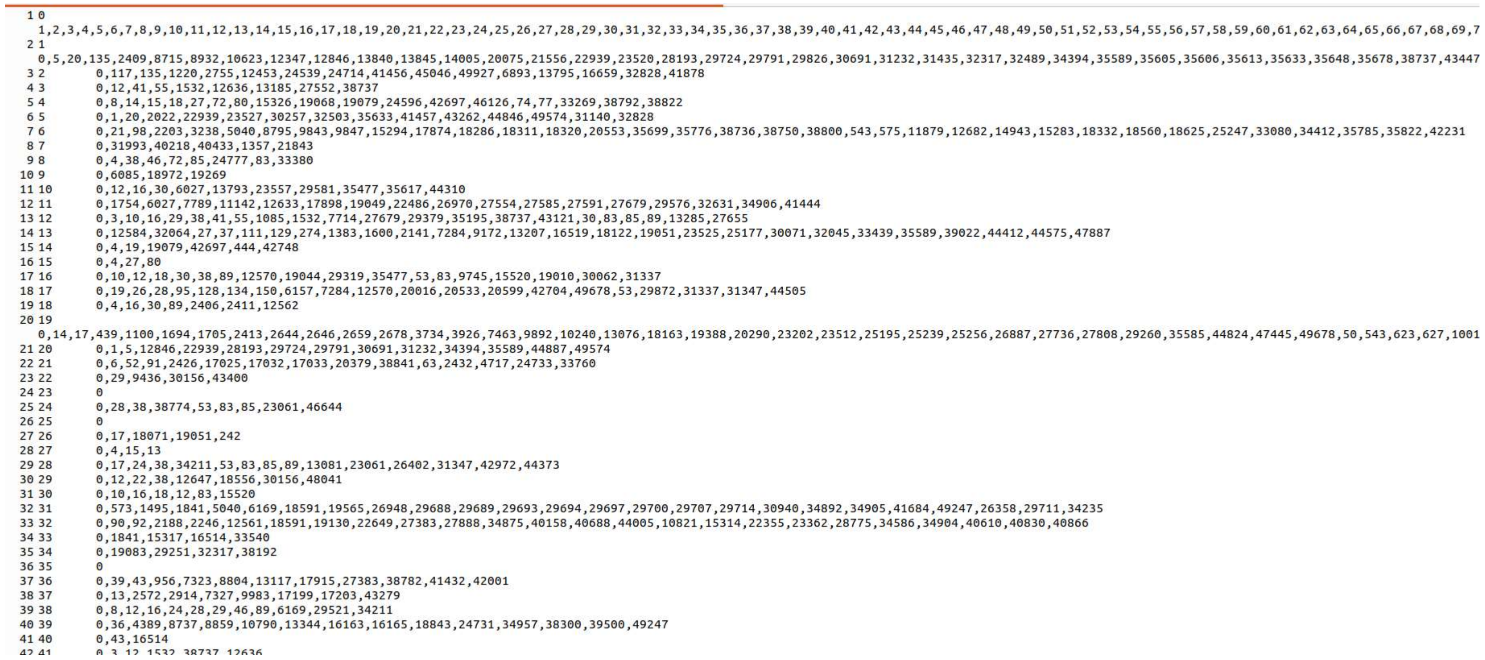

eclipse-workspace - bigdataproject/src/main/java/topten/TopTenFriends.java - Eclipse IDE

```java
                return 1;
            }
        }

    }

    public static void main(String[] args) throws Exception{
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        // get all args
        if (otherArgs.length != 3) {
            System.err.println("Usage: TopTepFriends <inputfile HDFS path> <outputfile1 HDFS path> <outputfile2 HDFS path>");
            System.exit(2);
        }
        @SuppressWarnings("deprecation")
        Job job = new Job(conf, "TopTepFriends Phase 1");
        job.setJarByClass(TopTenFriends.class);
        job.setMapperClass(TopTen_Map1.class);
        job.setReducerClass(TopTen_Reduce1.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
        // set the HDFS path for the output1 (this output file will contain count of the mutual friends between two friends)
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
        boolean mapreduce = job.waitForCompletion(true);

        if (mapreduce) {
            Configuration conf1 = new Configuration();
            @SuppressWarnings("deprecation")
            Job job1 = new Job(conf1, "TopTenFriends Phase 2");
            job1.setJarByClass(TopTenFriends.class);
            job1.setMapperClass(TopTen_Map2.class);
            job1.setReducerClass(TopTen_Reduce2.class);
            job1.setInputFormatClass(TextInputFormat.class);
            job1.setMapOutputKeyClass(IntWritable.class);
            job1.setMapOutputValueClass(Text.class);

```

Command to run jar file : Output of first mapper reducer is taken by second one as the input

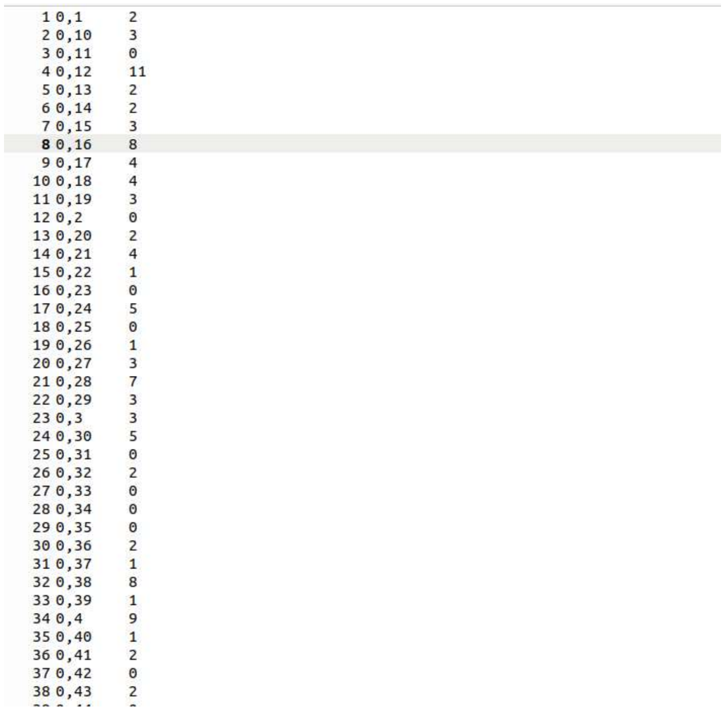
```
hadoop jar /home/sid/Desktop/toptenfriend.jar topten.TopTenFriends /friends.txt /finalproject/topten /finalproject/top2
```

Input Text File:



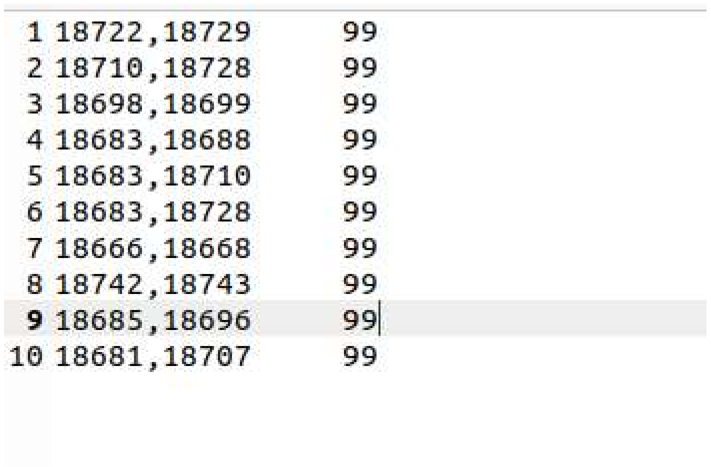
```
0
 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,7
1
 0,5,20,135,2409,8715,8932,10623,12347,12846,13840,13845,14005,20075,21556,22939,23520,28193,29724,29791,29826,30691,31232,31435,32317,32489,34394,35589,35605,35606,35613,35633,35648,35678,38737,43447
2       0,117,135,1220,2755,12453,24539,24714,41456,45046,49927,6893,13795,16659,32828,41878
3       0,12,41,55,1532,12636,13185,27552,38737
4       0,8,14,15,18,27,72,80,15326,19068,19079,24596,42697,46126,74,77,33269,38792,38822
5       0,1,20,2022,22939,23527,30257,32503,35633,41457,43262,44846,49574,31140,32828
6       0,21,98,2203,3238,5040,8795,9843,9847,15294,17874,18286,18311,18320,20553,35699,35776,38736,38750,38800,543,575,11879,12682,14943,15283,18332,18560,18625,25247,33080,34412,35785,35822,42231
7       0,31993,40218,40433,1357,21843
8       0,4,38,46,72,85,24777,83,33380
9       0,6085,18972,19269
10      0,12,16,30,6027,13793,23557,29581,35477,35617,44310
11      0,1754,6027,7789,11142,12633,17898,19049,22486,26970,27554,27585,27591,27679,29576,32631,34906,41444
12      0,3,10,16,29,38,41,55,1085,1532,7714,27679,29379,35195,38737,43121,30,83,85,89,13285,27655
13      0,12584,32064,27,37,111,129,274,1383,1600,2141,7284,9172,13207,16519,18122,19051,23525,25177,30071,32045,33439,35589,39022,44412,44575,47887
14      0,4,19,19079,42697,444,42748
15      0,4,27,80
16      0,10,12,18,30,38,89,12570,19044,29319,35477,53,83,9745,15520,19010,30062,31337
17      0,19,26,28,95,128,134,150,6157,7284,12570,20016,20533,20599,42704,49678,53,29872,31337,31347,44505
18      0,4,16,30,89,2406,2411,12562
19
 0,14,17,439,1100,1694,1705,2413,2644,2646,2659,2678,3734,3926,7463,9892,10240,13076,18163,19388,20290,23202,23512,25195,25239,25256,26887,27736,27808,29260,35585,44824,47445,49678,50,543,623,627,1001
20      0,1,5,12846,22939,28193,29724,29791,30691,31232,34394,35589,44887,49574
21      0,6,52,91,2426,17025,17032,17033,20379,38841,63,2432,4717,24733,33760
22      0,29,9436,30156,43400
23      0
24      0,28,38,38774,53,83,85,23061,46644
25      0
26      0,17,18071,19051,242
27      0,4,15,13
28      0,17,24,38,34211,53,83,85,89,13081,23061,26402,31347,42972,44373
29      0,12,22,38,12647,18556,30156,48041
30      0,10,16,18,12,83,15520
31      0,573,1495,1841,5040,6169,18591,19565,26948,29688,29689,29693,29694,29697,29700,29707,29714,30940,34892,34905,41684,49247,26358,29711,34235
32      0,90,92,2188,2246,12561,18591,19130,22649,27383,27888,34875,40158,40688,44005,10821,15314,22355,23362,28775,34586,34904,40610,40830,40866
33      0,1841,15317,16514,33540
34      0,19083,29251,32317,38192
35      0
36      0,39,43,956,7323,8804,13117,17915,27383,38782,41432,42001
37      0,13,2572,2914,7327,9983,17199,17203,43279
38      0,8,12,16,24,28,29,46,89,6169,29521,34211
39      0,36,4389,8737,8859,10790,13344,16163,16165,18843,24731,34957,38300,39500,49247
40      0,43,16514
41      0,3,12,1532,38737,12636
```

Output from 1st Mapper reducer – Friends pair with total number of common friends



```
0,1     2
0,10    3
0,11    0
0,12    11
0,13    2
0,14    2
0,15    3
0,16    8
0,17    4
0,18    4
0,19    3
0,2     0
0,20    2
0,21    4
0,22    1
0,23    0
0,24    5
0,25    0
0,26    1
0,27    3
0,28    7
0,29    3
0,3     3
0,30    5
0,31    0
0,32    2
0,33    0
0,34    0
0,35    0
0,36    2
0,37    1
0,38    8
0,39    1
0,4     9
0,40    1
0,41    2
0,42    0
0,43    2
```

Output from 2nd Mapper reducer – Top 10 Friends pair with highest number of common friends

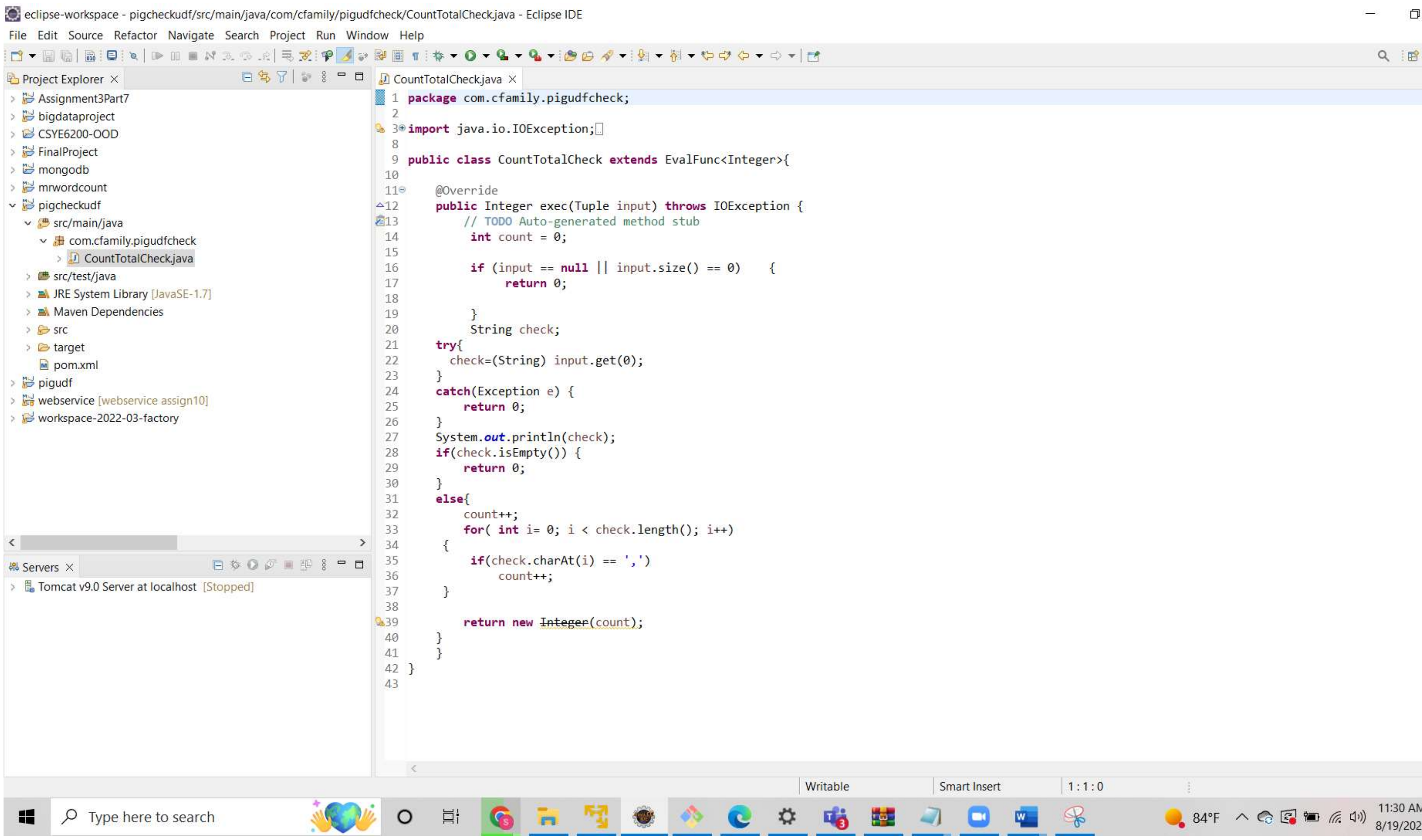


```
18722,18729      99
18710,18728      99
18698,18699      99
18683,18688      99
18683,18710      99
18683,18728      99
18666,18668      99
18742,18743      99
18685,18696      99
18681,18707      99
```

Find Count of Friends each person has Using Pig and arrange them in Desc Order using UDF

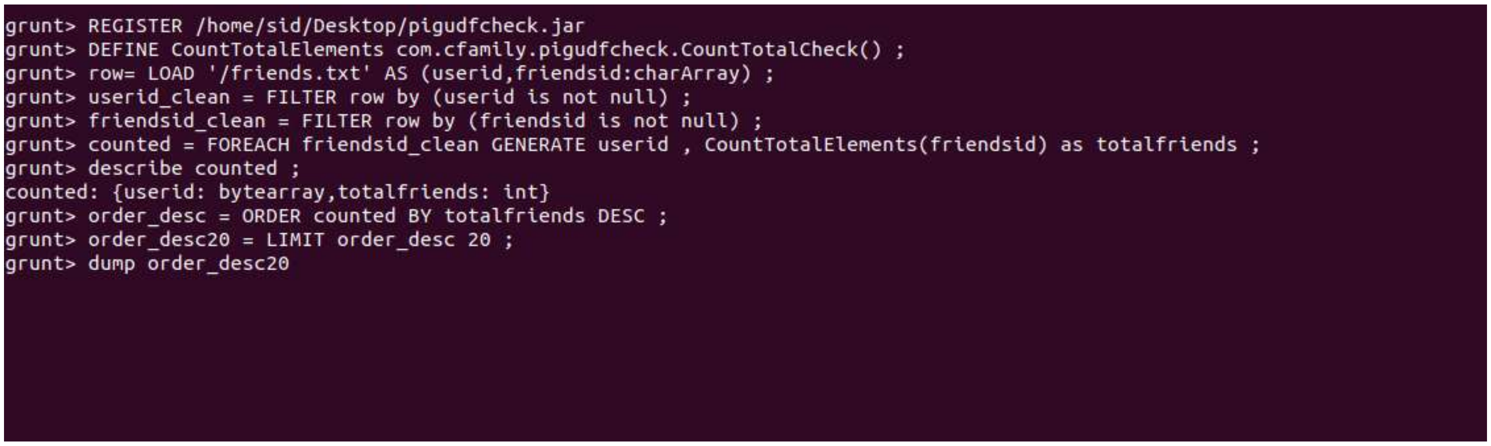
In case we need to find which person is most social we can use the algorithms above to calculate the mutual friends and top ten friends to suggest them this will help as they are more social. Also arranging them in ascending order helps us to find the users which do not have low friends so would let us help to suggest them more friends as they can help in increasing the time on the site

UDF Requirements

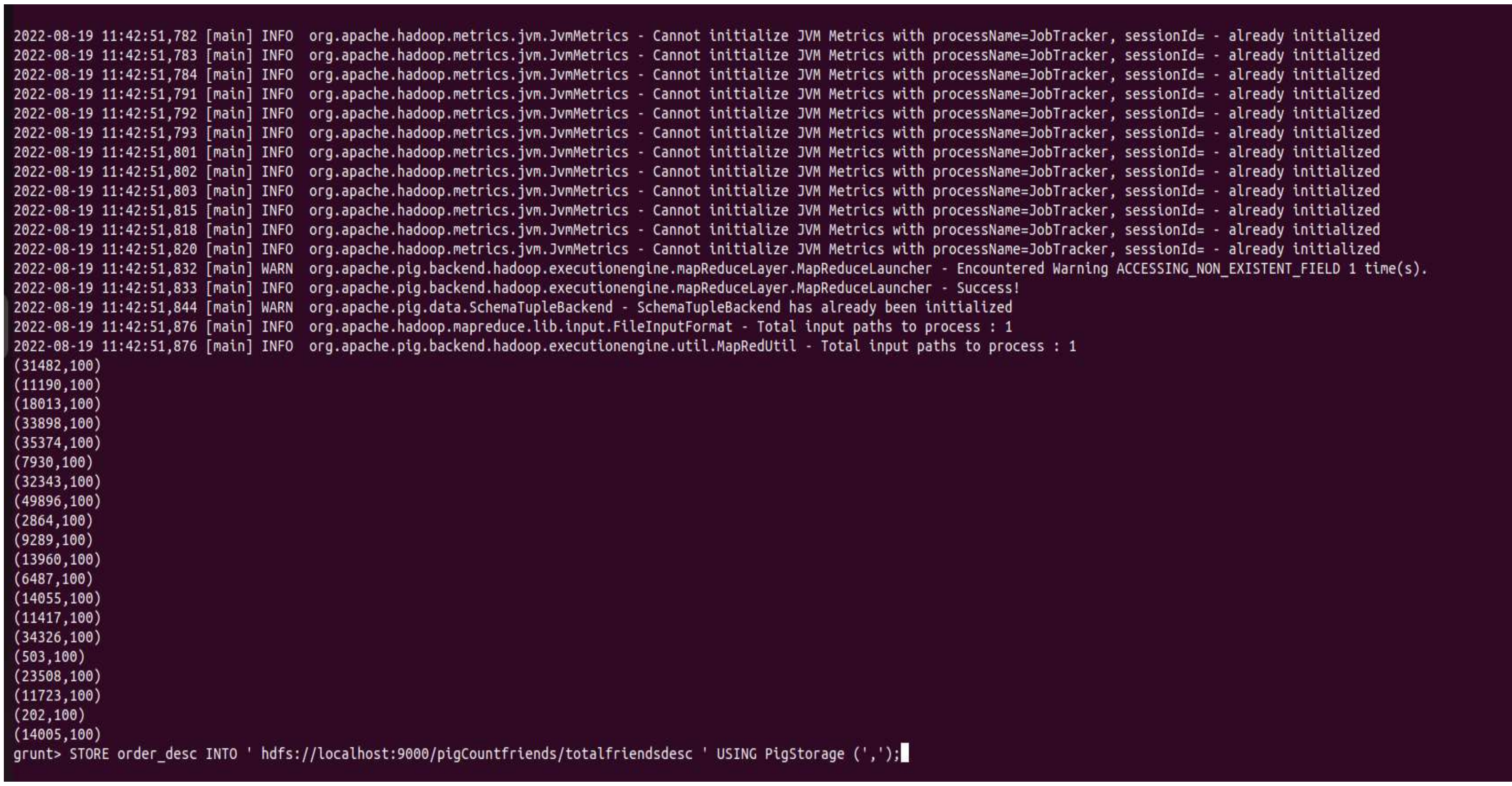


```java
package com.cfamily.pigudfcheck;

import java.io.IOException;

public class CountTotalCheck extends EvalFunc<Integer>{

    @Override
    public Integer exec(Tuple input) throws IOException {
        // TODO Auto-generated method stub
         int count = 0;

         if (input == null || input.size() == 0)    {
              return 0;

         }
         String check;
    try{
      check=(String) input.get(0);
    }
    catch(Exception e) {
        return 0;
    }
    System.out.println(check);
    if(check.isEmpty()) {
        return 0;
    }
    else{
        count++;
        for( int i= 0; i < check.length(); i++)
     {
         if(check.charAt(i) == ',')
             count++;
     }

        return new Integer(count);
    }
    }
}
```

As our data for friends id is , (comma) separated and thereby we cannot use functions like Count on it , we need to create UDF(user defined functions) to count friends . Create a jar file of the project , register and define to pig and then we can use the function.

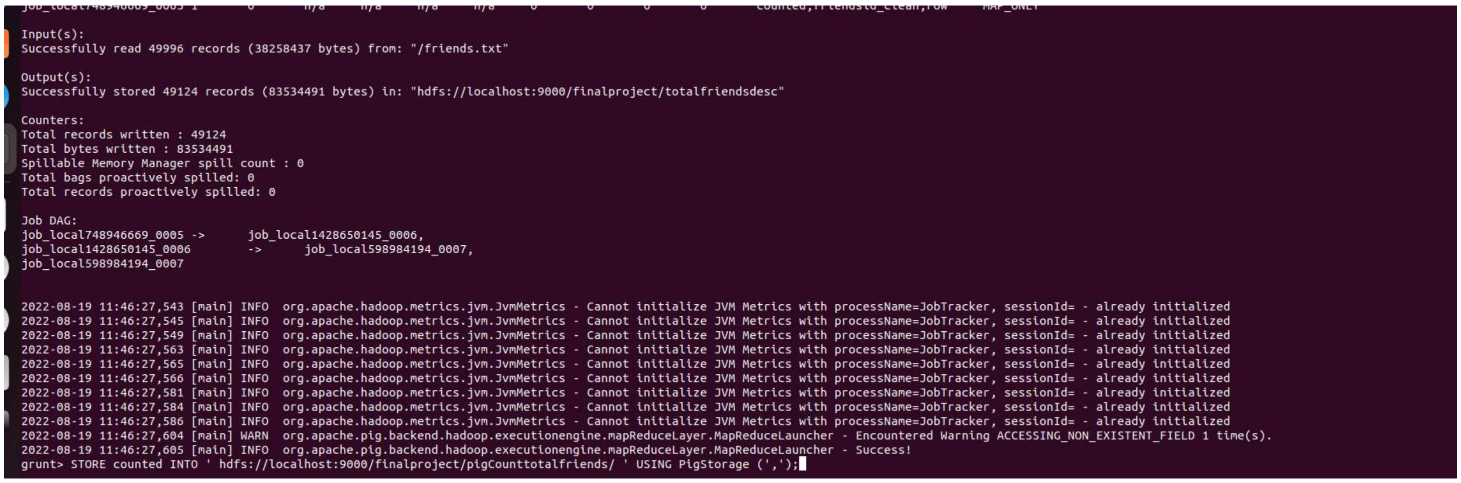


```
grunt> REGISTER /home/sid/Desktop/pigudfcheck.jar
grunt> DEFINE CountTotalElements com.cfamily.pigudfcheck.CountTotalCheck() ;
grunt> row= LOAD '/friends.txt' AS (userid,friendsid:charArray) ;
grunt> userid_clean = FILTER row by (userid is not null) ;
grunt> friendsid_clean = FILTER row by (friendsid is not null) ;
grunt> counted = FOREACH friendsid_clean GENERATE userid , CountTotalElements(friendsid) as totalfriends ;
grunt> describe counted ;
counted: {userid: bytearray,totalfriends: int}
grunt> order_desc = ORDER counted BY totalfriends DESC ;
grunt> order_desc20 = LIMIT order_desc 20 ;
grunt> dump order_desc20
```



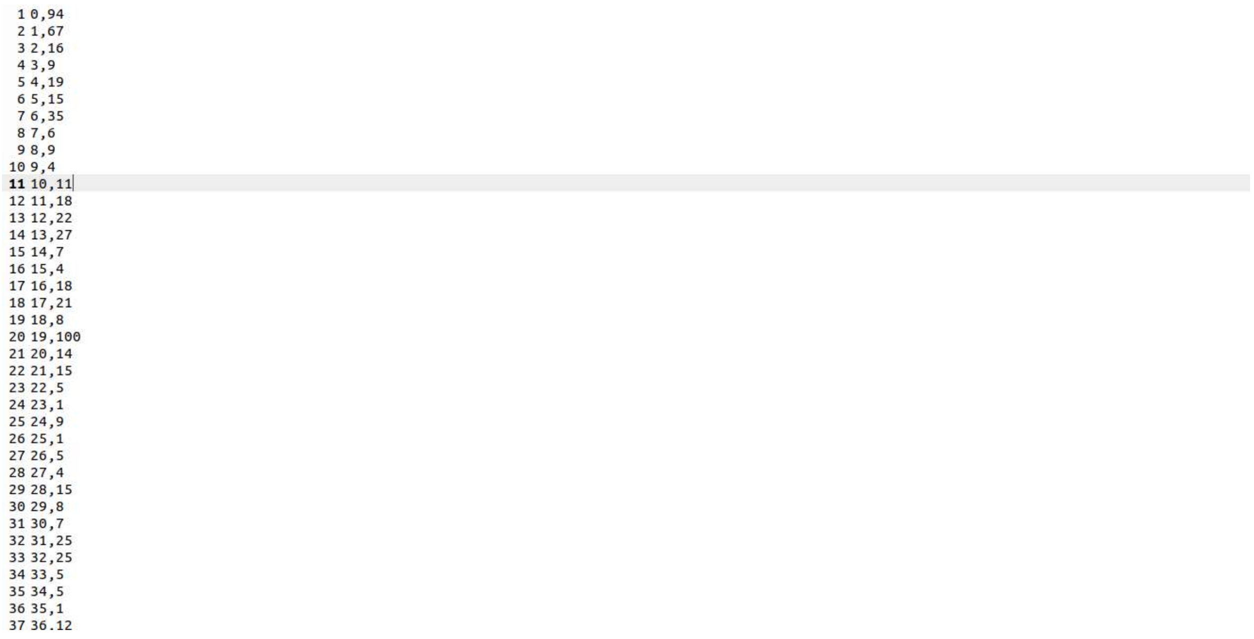
```
2022-08-19 11:42:51,782 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,783 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,784 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,791 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,792 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,793 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,801 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,802 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,803 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,815 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,818 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,820 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:42:51,832 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTENT_FIELD 1 time(s).
2022-08-19 11:42:51,833 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-08-19 11:42:51,844 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2022-08-19 11:42:51,876 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2022-08-19 11:42:51,876 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(31482,100)
(11190,100)
(18013,100)
(33898,100)
(35374,100)
(7930,100)
(32343,100)
(49896,100)
(2864,100)
(9289,100)
(13960,100)
(6487,100)
(14055,100)
(11417,100)
(34326,100)
(503,100)
(23508,100)
(11723,100)
(202,100)
(14005,100)
grunt> STORE order_desc INTO ' hdfs://localhost:9000/pigCountfriends/totalfriendsdesc ' USING PigStorage (',');
```

Saving the output of total and Desc friends in HDFS



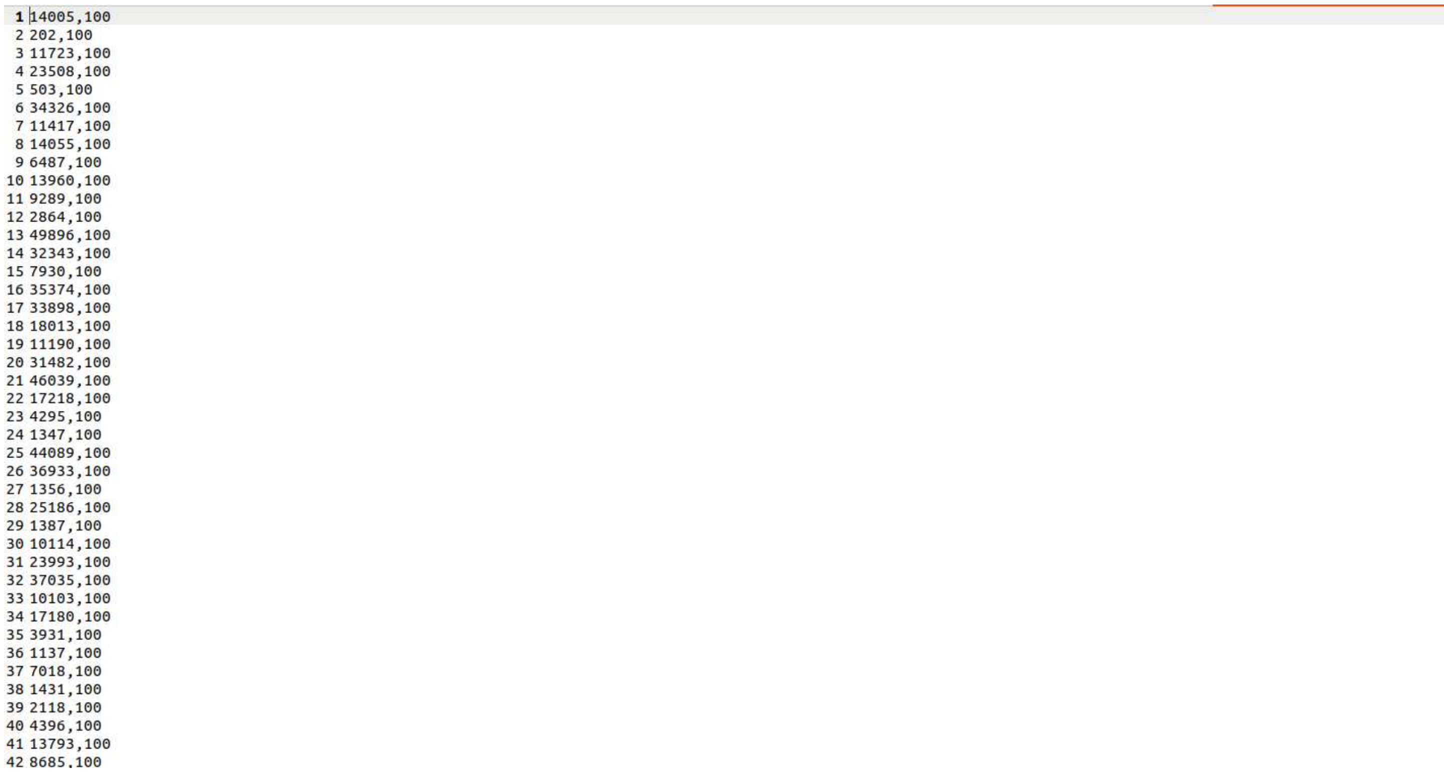
```
Input(s):
Successfully read 49996 records (38258437 bytes) from: "/friends.txt"

Output(s):
Successfully stored 49124 records (83534491 bytes) in: "hdfs://localhost:9000/finalproject/totalfriendsdesc"

Counters:
Total records written : 49124
Total bytes written : 83534491
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local748946669_0005 ->      job_local1428650145_0006,
job_local1428650145_0006        ->      job_local598984194_0007,
job_local598984194_0007


2022-08-19 11:46:27,543 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,545 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,549 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,563 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,565 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,566 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,581 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,584 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,586 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2022-08-19 11:46:27,604 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTENT_FIELD 1 time(s).
2022-08-19 11:46:27,605 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt> STORE counted INTO ' hdfs://localhost:9000/finalproject/pigCounttotalfriends/ ' USING PigStorage (',');
```

Output Total Friends per User ID (left of comma is user id , right is total friends)



```
0,94
1,67
2,16
3,9
4,19
5,15
6,35
7,6
8,9
9,4
10,11
11,18
12,22
13,27
14,7
15,4
16,18
17,21
18,8
19,100
20,14
21,15
22,5
23,1
24,9
25,1
26,5
27,4
28,15
29,8
30,7
31,25
32,25
33,5
34,5
35,1
36.12
```

Outpur Total Friends per User ID in Descending Order of Friends(left of comma is user id , right is total friends)



```
14005,100
202,100
11723,100
23508,100
503,100
34326,100
11417,100
14055,100
6487,100
13960,100
9289,100
2864,100
49896,100
32343,100
7930,100
35374,100
33898,100
18013,100
11190,100
31482,100
46039,100
17218,100
4295,100
1347,100
44089,100
36933,100
1356,100
25186,100
1387,100
10114,100
23993,100
37035,100
10103,100
17180,100
3931,100
1137,100
7018,100
1431,100
2118,100
4396,100
13793,100
8685.100
```