LAB - 5

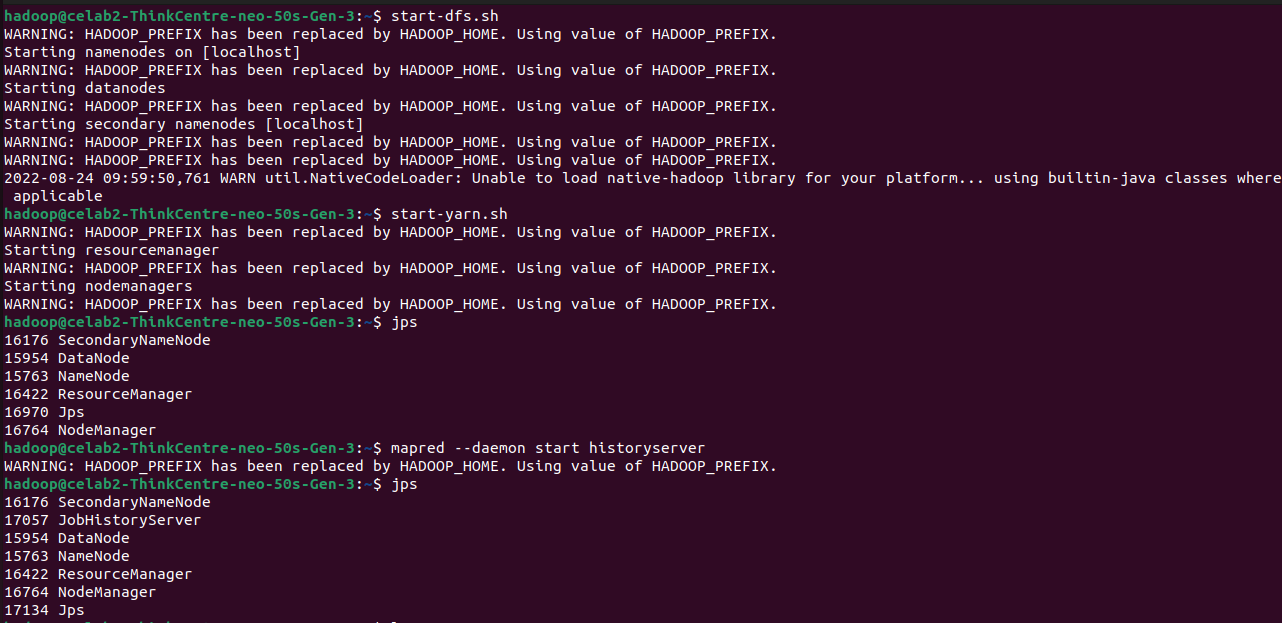
|  |  |
| --- | --- |
| Name | Keval D Gandevia |
| Roll Number | CE046 |
| ID | 19CEUEG017 |
| Subject | Big Data and Analytics |

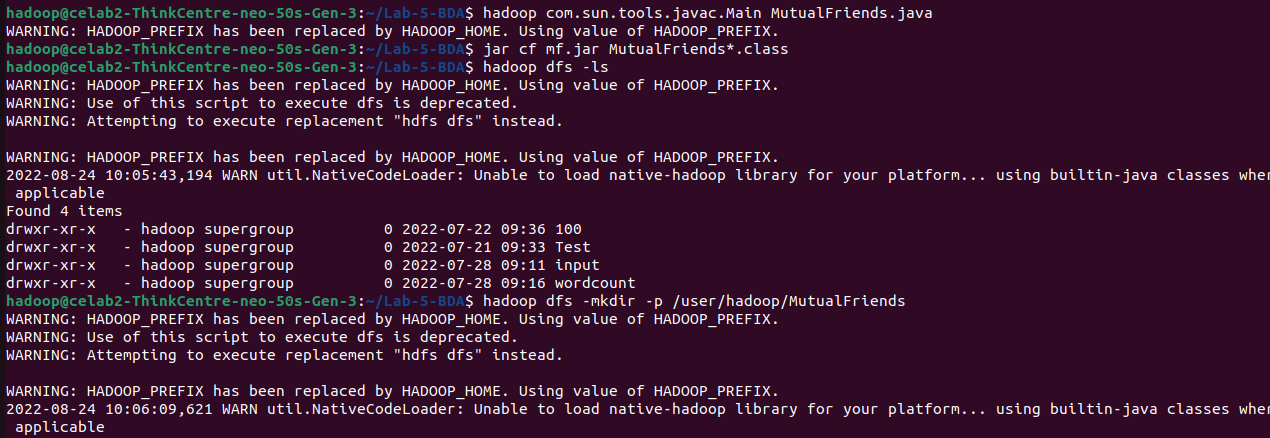
**Aim:** Finding mutual friends for every two friends using MapReduce framework.

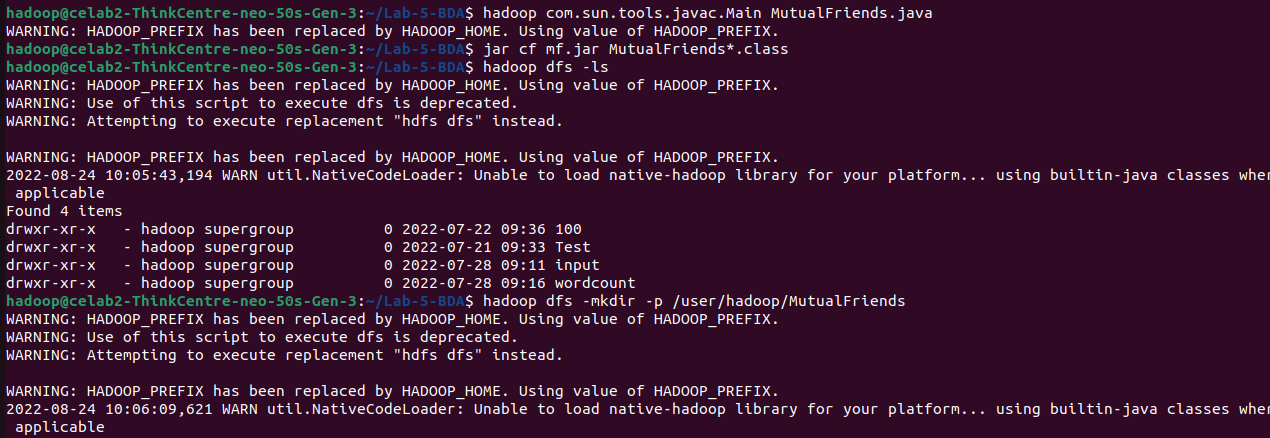
* **Java code:**

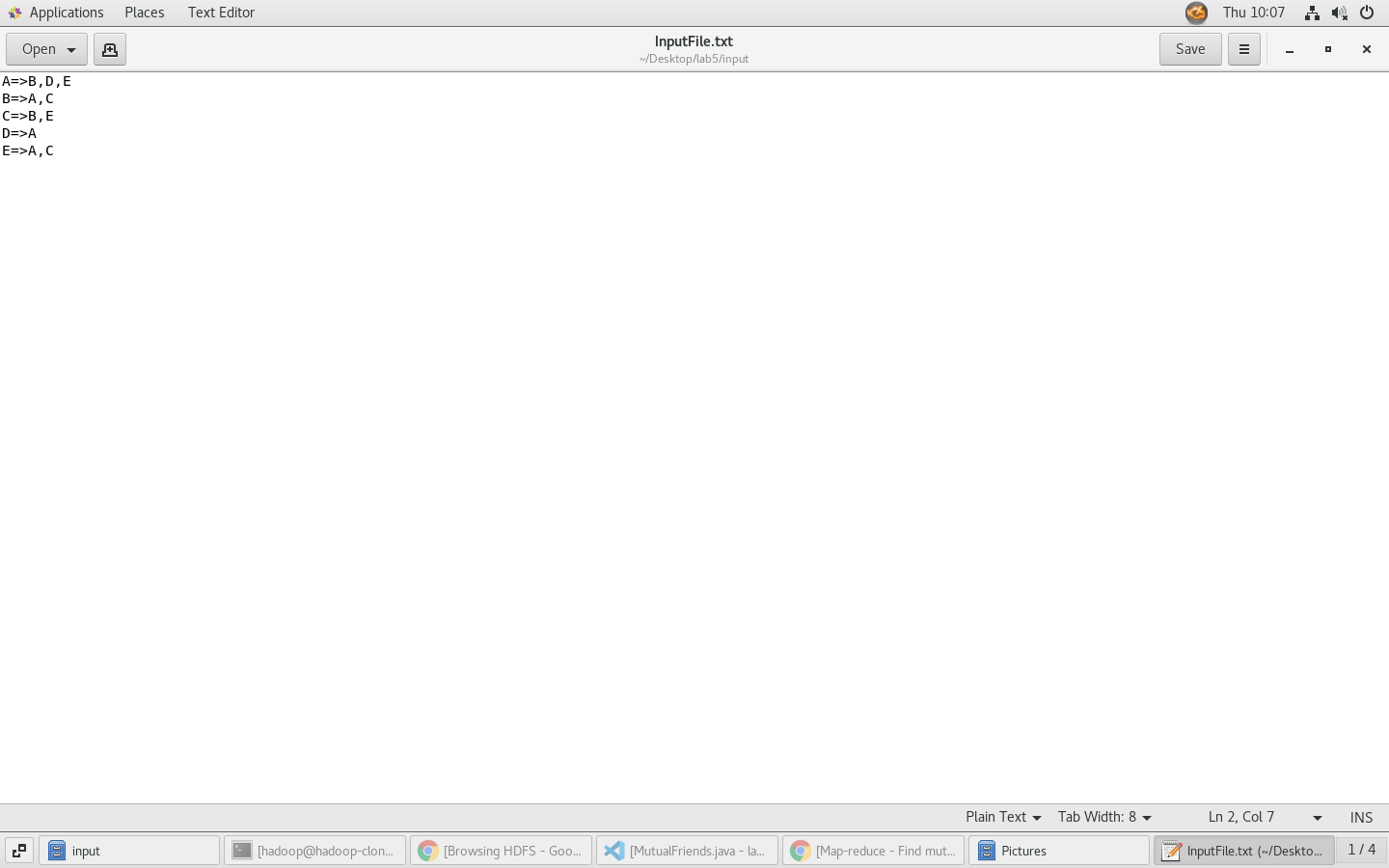
|  |
| --- |
| import java.io.IOException;  import java.util.ArrayList;  import java.util.StringTokenizer;  import org.apache.hadoop.conf.Configuration;  import org.apache.hadoop.fs.Path;  import org.apache.hadoop.io.\*;  import org.apache.hadoop.io.Text;  import org.apache.hadoop.mapreduce.Job;  import org.apache.hadoop.mapreduce.Mapper;  import org.apache.hadoop.mapreduce.Reducer;  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  public class MutualFriends  {      public static class MutualFriendMapper extends Mapper <LongWritable, Text, Text, Text>      {          public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException          {              String[] Line = value.toString().split("=>");              String commonFriend = Line[0];              Text c = *new* Text();              c.set(commonFriend);              String[] friends = Line[1].split(",");  *for*(int i = 0; i < friends.length; i++)              {  *for*(int j = i + 1; j < friends.length; j++)                  {                      Text t1 = *new* Text();                      t1.set((friends[i] + "," + friends[j]).toString());                      context.write(t1, c);                  }              }          }      }      public static class MutualFriendReducer extends Reducer <Text, Text, Text, Text>      {          public void reduce(Text key, Iterable <Text> values, Context context) throws IOException,InterruptedException          {              String resultVal = "";              Text t1 = *new* Text();  *for*(Text t*:*values)              {                  String res = t.toString();                  resultVal += (res + ",");              }              t1.set(resultVal);              context.write(key, t1);          }        }      public static void main(String[] args) throws Exception      {          Configuration   conf = *new* Configuration();          Job job = Job.getInstance(conf, "Mutual Friends");          job.setJarByClass(MutualFriends.class);          job.setMapperClass(MutualFriendMapper.class);          job.setCombinerClass(MutualFriendReducer.class);          job.setReducerClass(MutualFriendReducer.class);          job.setOutputKeyClass(Text.class);          job.setOutputValueClass(Text.class);          FileInputFormat.addInputPath(job, *new* Path(args[0]));          FileOutputFormat.setOutputPath(job, *new* Path(args[1]));          System.exit(job.waitForCompletion(true) *?* 0 *:* 1);      }  } |

* **Steps to get the output:**

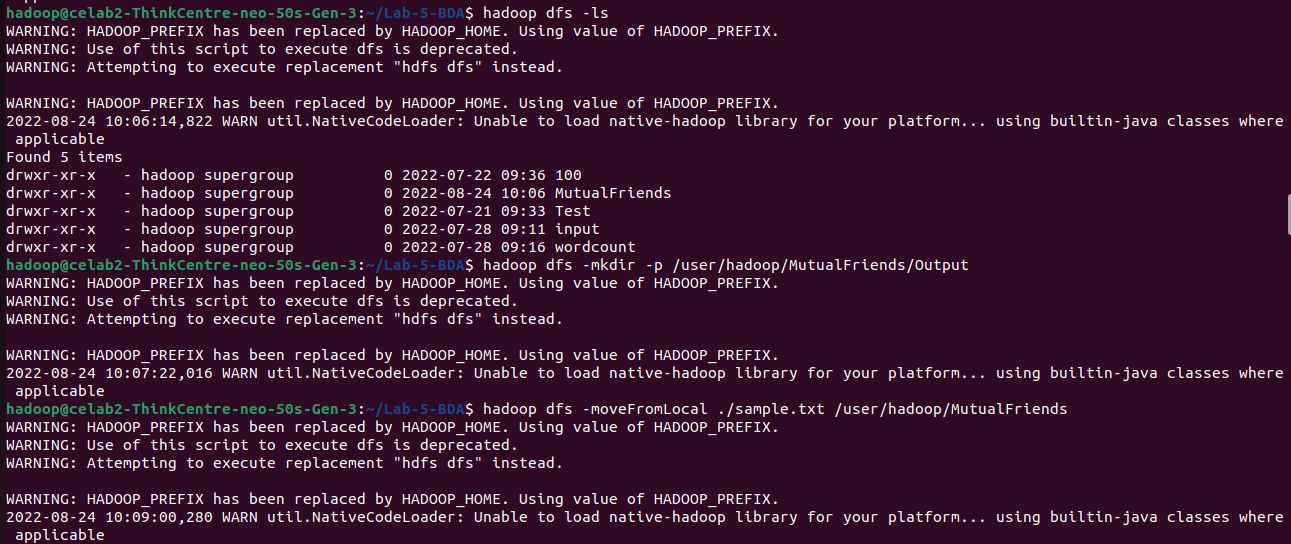
1. **Starting the Hadoop server, yarn, and history server.**
2. **Compiling the code and creating a jar file.**



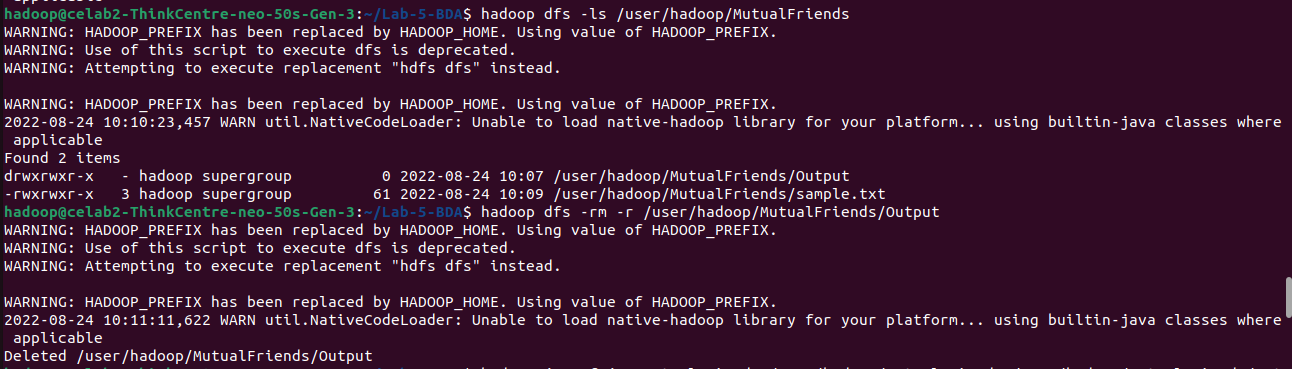
1. **Creating input directory.**
2. **Creating a text file which contains the input.**

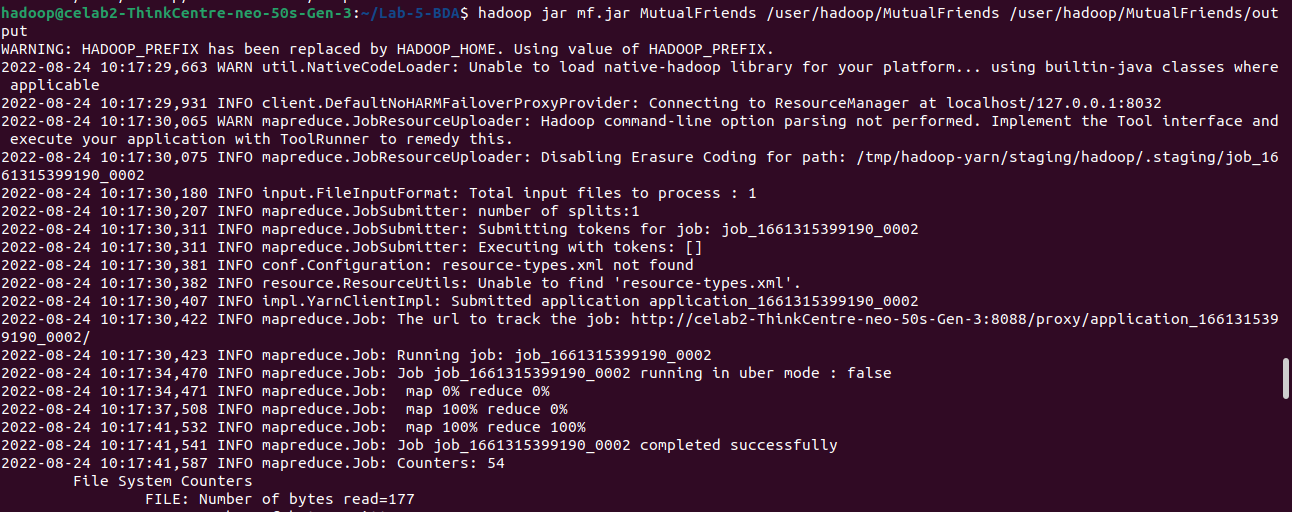
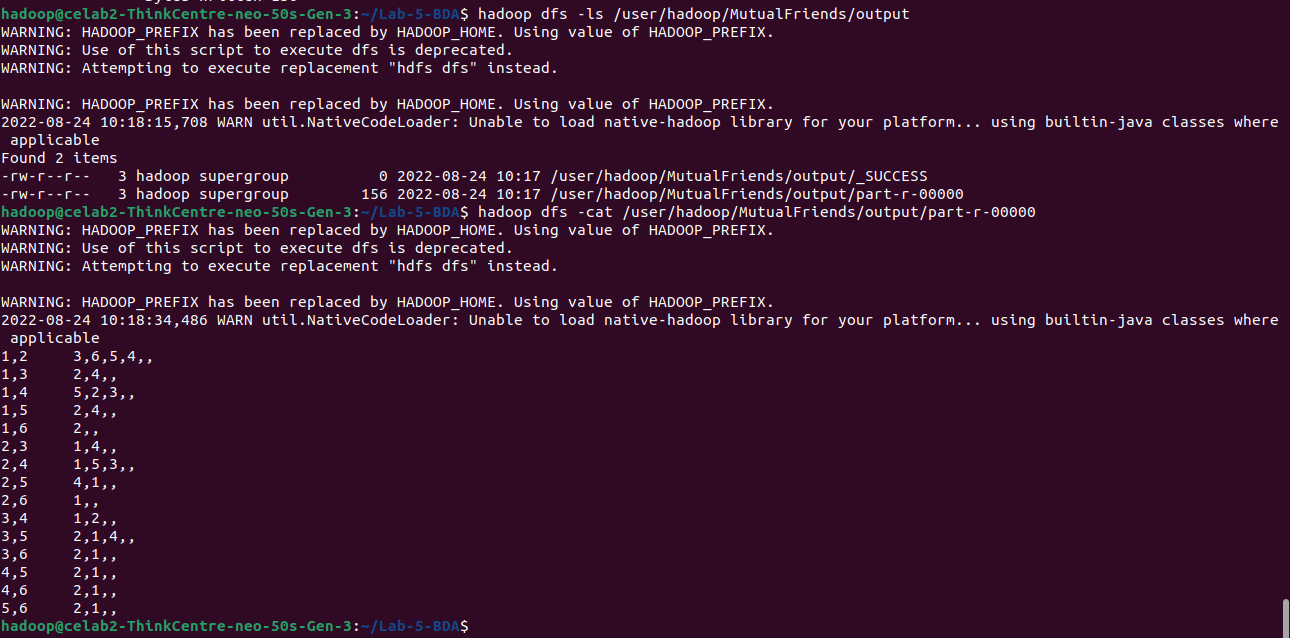


1. **Uploading file to HDFS.**



1. **Removing the output directory.**



1. **Executing the program.**
2. **Final output.**