Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

Big Data Processing- Coursework1 – Twitter Analysis

PART A - CONTENT ANALYSIS

TWEET MAPPER CODE:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.commons.lang.StringUtils;

public class TwitterMapper extends Mapper<Object, Text, IntWritable, IntWritable> {
    //constant variable stores a value of one
    private final IntWritable one = new IntWritable(1);
    //mapper method takes takes as input the text file
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
    // Format per tweet is id;date;hashtags;tweet;

    String dump = value.toString();
    if(StringUtils.ordinalIndexOf(dump,";",4)>-1){
        int startIndex = StringUtils.ordinalIndexOf(dump,";",3) + 1;
        String tweet = dump.substring(startIndex,dump.lastIndexOf(';'));
    //store tweet length in a variable
        IntWritable tweetSize=new IntWritable(tweet.length());
    //send tweet length and its occurrence to reducer
    context.write(tweetSize,one);

      }//END if statement

  }//END map method

}//END TwitterMapper
```

TWEET REDUCER CODE:

```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;


public class TweetReducer extends Reducer<IntWritable, IntWritable, IntWritable, IntWritable> {

    private IntWritable result = new IntWritable();
    //reducer method takes key from mapper and value from mapper as input
    public void reduce(IntWritable key, Iterable<IntWritable> occurrences, Context context)

            throws IOException, InterruptedException {

    int sum = 0;
    //each value is iterated over to sum up each occurence of X length
        for (IntWritable occurrence : occurrences) {

            sum+=occurrence.get();

        }
           //variable stores the above sum of occurences of X length
              result.set(sum);
      //the results i.e: key: length and total occurences of length X, Y, Z... are emitted
      context.write(key,result);

    }

}
```

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

TWEET COUNT CODE:

```
public class TweetCount {

  public static void runJob(String[] input, String output) throws Exception {

      Configuration conf = new Configuration();

    Job job = new Job(conf);
    //sets current class as the class which generates and enables executability of a jar file
    job.setJarByClass(TweetCount.class);
    //mapper class is set
    job.setMapperClass(TwitterMapper.class);
    //reducer class is set
    job.setReducerClass(TweetReducer.class);
    //sets output key's class type
    job.setMapOutputKeyClass(IntWritable.class);
     //sets output value's class type
    job.setMapOutputValueClass(IntWritable.class);
    //produces three tasks which output three text files with its corresponding results
    job.setNumReduceTasks(3);
    Path outputPath = new Path(output);
    FileInputFormat.setInputPaths(job, StringUtils.join(input, ","));
    FileOutputFormat.setOutputPath(job, outputPath);
    outputPath.getFileSystem(conf).delete(outputPath,true);
    job.waitForCompletion(true);
  }

  public static void main(String[] args) throws Exception {
      runJob(Arrays.copyOfRange(args, 0, args.length-1), args[args.length-1]);
  }

}
```

EXPLANATION:

The Map Reduce approach with the first part of (**A) Content Analysis** is where a counter is created to store 1 as a value which keeps track of a particular tweet length within the mapper and thus is sent (emitted) to the reducer to total up and count how many times a certain tweet length occurred within the data set (data/olympictweets). Henceforth, a map method's would store the tweet length of every single tweet in a variable and emits it directly to the reducer along with its corresponding occurrence within the data set i.e.: length=300, occurrence=200. Which means for length 300 it would occur in the data set 200x.

In addition, a reducer method would take as input the key being a length and its occurrence being a value. It would contain an iteration which iterates through each occurrence and sums up the whole occurrence total for X length:

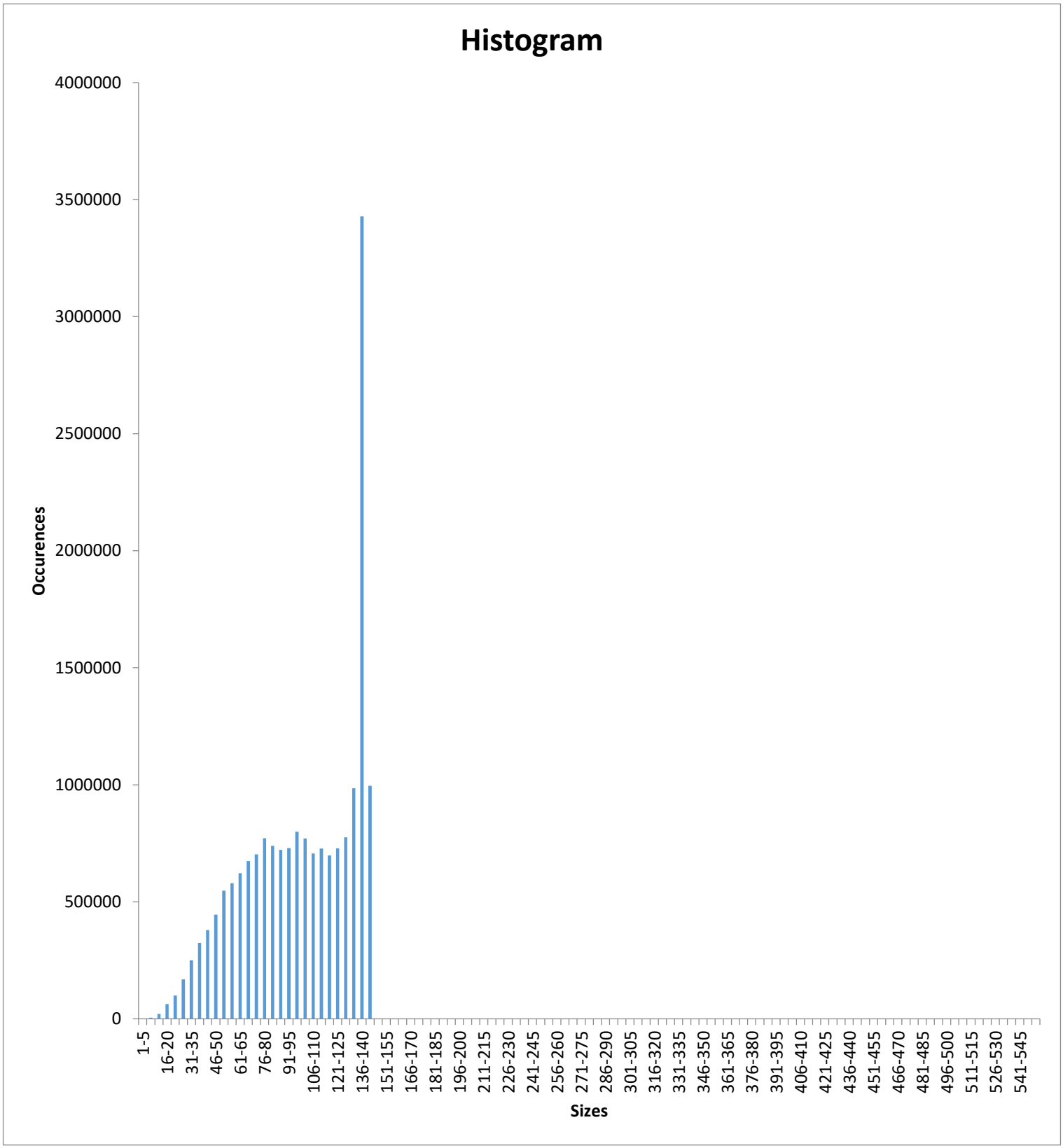
i.e.:

length=12, occurrence=1

length=12, occurrence=1

As a result, length=12, would result in two occurrences; this total of occurrences is set as an output value.

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432



Furthermore, this enabled me to produce the histogram displayed above in Excel, which clearly denotes that the range of lengths 136-140 had the most lengths and that certain ranges had no lengths reported within in the dataset at all.

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

(ii) AVERAGE LENGTH

TWEET AVERAGE MAPPER CODE:

```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.commons.lang.StringUtils;

public class TweetAverageMapper extends Mapper<Object, Text, Text, IntWritable> {
    //constant variable stores a value of one
    private final IntWritable one = new IntWritable(1);
    //create a text variable
    Text tweetStr=new Text("");
    //mapper method takes takes as input the text file
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
    // Format per tweet is id;date;hashtags;tweet;

    String dump = value.toString();
    if(StringUtils.ordinalIndexOf(dump,";",4)>-1){
        int startIndex = StringUtils.ordinalIndexOf(dump,";",3) + 1;
        String tweet = dump.substring(startIndex,dump.lastIndexOf(';'));
    //use tweettweet as a key
    String tweettweet="tweet";
    //store tweet length in a variable
    IntWritable tweetLengths=new IntWritable(tweet.length());
    //text variable stores a string which will be used as a key
    tweetStr.set(tweettweet);
    //send text variable and its contents as a key and tweet length as a value to reducer
    context.write(tweetStr,tweetLengths);

      }//END if statement

  }//END map method

}//END TweetAverageMapper
```

TWEET AVERAGE REDUCER CODE:

```
    private IntWritable result = new IntWritable();

     //reducer method takes key from mapper and value from mapper as input
    public void reduce(Text key, Iterable<IntWritable> tweetLengths, Context context)

                throws IOException, InterruptedException {
    //create a variable to store the count
    int count =0;
    //create a variable to store all the lengths added together
        int totalTweetSize = 0;
    //iterate through the sum of all tweet lengths and store that sum in a variable and count how many lengths exist altog
        for (IntWritable tweetLength : tweetLengths) {

            totalTweetSize+=tweetLength.get();
        //increment the counter variable
        count++;

        }
    //store a average of all the lengths totalled up divided by how many lengths which exist in the data set
    int avg=totalTweetSize/count;

        //variable stores the above average calculation of each length
              result.set(avg);



      //the results i.e: key: "tweet" and the average of all lengths are emitted
      context.write(key,result);

    }
```

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

TWEET AVERAGE COUNT CODE:

```
public class TweetAvgCount {

  public static void runJob(String[] input, String output) throws Exception {

      Configuration conf = new Configuration();

    Job job = new Job(conf);
    //sets current class as the class which generates and enables executability of a jar file
    job.setJarByClass(TweetAvgCount.class);
     //mapper class is set
    job.setMapperClass(TweetAverageMapper.class);
    //reducer class is set
    job.setReducerClass(TweetAverageReducer.class);
    //sets output key's class type
    job.setMapOutputKeyClass(Text.class);
     //sets output value's class type
    job.setMapOutputValueClass(IntWritable.class);
     //produces three tasks which output three text files with its corresponding results
    job.setNumReduceTasks(3);
    Path outputPath = new Path(output);
    FileInputFormat.setInputPaths(job, StringUtils.join(input, ","));
    FileOutputFormat.setOutputPath(job, outputPath);
    outputPath.getFileSystem(conf).delete(outputPath,true);
    job.waitForCompletion(true);
  }

  public static void main(String[] args) throws Exception {
      runJob(Arrays.copyOfRange(args, 0, args.length-1), args[args.length-1]);
  }

}//END TweetAvgCount
```

EXPLANATION:

The Map Reduce approach with the second part of (**A) Content Analysis** enables you to retrieve the total number of tweet lengths and the total size of all tweet lengths totalled up. A key such as a string is used to denote the tweet and the tweet length is set as a value to be emitted to the reducer. Then you can find the average of tweet lengths by adding all the tweet lengths together and diving them by how many lengths exist within the dataset. The average is then emitted as a value along with “tweet” as a key.

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

PART B- TIME ANALYSIS

TWEET DATE TIME MAPPER CODE:

```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.commons.lang.StringUtils;

public class TweetDateTimeMapper extends Mapper<Object, Text, Text, IntWritable> {
    //constant variable stores a value of one
    private final IntWritable one = new IntWritable(1);
      //create a text variable
    Text data=new Text();
     //mapper method takes takes as input the text file
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
    // Format per tweet is id;date;hashtags;tweet;

    String dump = value.toString();
    if(StringUtils.ordinalIndexOf(dump,";",4)>-1){
    //set the start index of the dataset to the date of the tweet
        int startIndex = StringUtils.ordinalIndexOf(dump,";",1) + 1;
    //store the date of the tweet in a variable
        String tweet = dump.substring(startIndex,dump.lastIndexOf(';'));
    //delimit each tweet date by removing the commas
    String args[]=tweet.split(",");
    //text variable stores the date of the tweet
    data.set(args[0]);
    //send tweet String and one to reducer
    context.write(data,one);

      }//END if statement

  }//END map method

}//END TweetDateTimeMapper
```

TWEET DATE TIME REDUCER CODE:

```
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;


public class TweetDateTimeReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

    private IntWritable result = new IntWritable();

      //reducer method takes key from mapper and value from mapper as input
    public void reduce(Text key, Iterable<IntWritable> values, Context context)

              throws IOException, InterruptedException {
    //produce a count variable which will store the overall total occurence of X date within the dataset
    int count =0;

    //iterate through each occurence of X date and total up to find the total number of times X date appeared in the dataset
        for (IntWritable value : values) {

            count+=value.get();


        }

        //set the occurence of X date as output
               result.set(count);


    //emit the date and its occurence
      context.write(key,result);

    }

}//END TweetDateTimeReducer
```

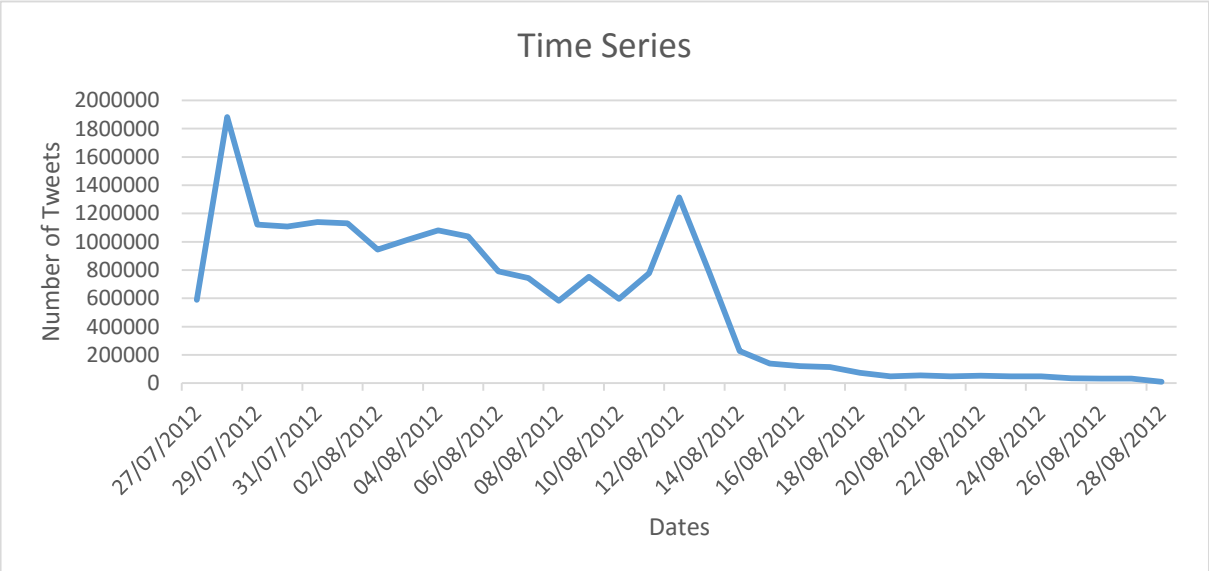
Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

TWEET DATE TIME COUNT CODE:

```java
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class TweetDateTimeCount {

  public static void runJob(String[] input, String output) throws Exception {

      Configuration conf = new Configuration();

    Job job = new Job(conf);
    //sets current class as the class which generates and enables executability of a jar file
    job.setJarByClass(TweetDateTimeCount.class);
     //mapper class is set
    job.setMapperClass(TweetDateTimeMapper.class);
    //reducer class is set
    job.setReducerClass(TweetDateTimeReducer.class);
    //sets output key's class type
    job.setMapOutputKeyClass(Text.class);
    //sets output value's class type
    job.setMapOutputValueClass(IntWritable.class);
     //produces three tasks which output three text files with its corresponding resultsl
    job.setNumReduceTasks(3);
    Path outputPath = new Path(output);
    FileInputFormat.setInputPaths(job, StringUtils.join(input, ","));
    FileOutputFormat.setOutputPath(job, outputPath);
    outputPath.getFileSystem(conf).delete(outputPath,true);
    job.waitForCompletion(true);
  }

  public static void main(String[] args) throws Exception {
      runJob(Arrays.copyOfRange(args, 0, args.length-1), args[args.length-1]);
  }

}//END TweetDateTimeCount
```

EXPLANATION:

The Map Reduce approach with the second part of (**B) Time Analysis** enables you to retrieve the date that a tweet was posted and count each date and store an occurrence of each date within the dataset and thus, compare how many tweets were made on a particular date.



As the time series diagram show above, the number of tweets fluctuate varying day to day, sometimes increasing across a number of days and sometimes decreasing. Furthermore, the day which had the greatest number of tweet at a value of just over 180, 0000 tweets occurred 27 July 2012 - 29 July 2012.

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

PART C- HASHTAG ANALYSIS

TWEET CHEER MAPPER COUNT CODE:

TweetDateTimeMapper.java | TweetDateTimeReducer.java | TweetCheerCount.java | TweetCheerMapper.java | TweetCheerReducer.java

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.commons.lang.StringUtils;

public class TweetCheerMapper extends Mapper<Object, Text, Text, IntWritable> {
    //constant variable stores a value of one
    private final IntWritable one = new IntWritable(1);
    //create a text variable
    Text data=new Text();
    //mapper method takes takes as input the text file
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {

    String dump = value.toString();
    if(StringUtils.ordinalIndexOf(dump,";",4)>-1){
        int startIndex = StringUtils.ordinalIndexOf(dump,";",3 + 1);
        String tweet = dump.substring(startIndex,dump.lastIndexOf(";"));
        //create a matrix which stores the countries from the method below
    String[][] tweetInformation = getCountryTweets();
        //iterate through the matrix and check whether the tweet within the dataset contains any key:cheer celebration and va
    for(int i=0;i<tweetInformation.length;i++){
      if(tweet.contains(tweetInformation[i][0])){
    //sets the contents within the array to lower case and stores that within the text variable which will be sent to the red
    data.set(tweetInformation[i][1].toLowerCase());
    //send country as a key and its occurrence as a value to the reducer
    context.write(data,one);
    }//END for loop
    }//END if statement


     }//END if statement
```

TWEET CHEER REDUCER CODE:

```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;


public class TweetCheerReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

    private IntWritable result = new IntWritable();
      //reducer method takes key from mapper and value from mapper as input
    public void reduce(Text key, Iterable<IntWritable> occurrences, Context context)

              throws IOException, InterruptedException {
    //use a variable to track the total number of times X country appears within the dataset
       int total = 0;
    //iterate through the occurences of X country and total up how many times X country was cheered
       for (IntWritable occurrence : occurrences) {

           total=total+occurrence.get();

       }
       //set the total occurences as output
              result.set(total);
    //emit the country and its total occurence within the dataset
     context.write(key,result);

    }

}
```

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

TWEET CHEER COUNT CODE:

```
7  import org.apache.hadoop.mapreduce.Job;
8  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
9  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
10
11 public class TweetCheerCount {
12
13   public static void runJob(String[] input, String output) throws Exception {
14
15       Configuration conf = new Configuration();
16
17     Job job = new Job(conf);
18     //sets current class as the class which generates and enables executability of a jar file
19     job.setJarByClass(TweetCheerCount.class);
20      //mapper class is set
21     job.setMapperClass(TweetCheerMapper.class);
22     //reducer class is set
23     job.setReducerClass(TweetCheerReducer.class);
24     //sets output key's class type
25     job.setMapOutputKeyClass(Text.class);
26     //sets output value's class type
27     job.setMapOutputValueClass(IntWritable.class);
28      //produces three tasks which output three text files with its corresponding results
29     job.setNumReduceTasks(3);
30     Path outputPath = new Path(output);
31     FileInputFormat.setInputPaths(job, StringUtils.join(input, ","));
32     FileOutputFormat.setOutputPath(job, outputPath);
33     outputPath.getFileSystem(conf).delete(outputPath,true);
34     job.waitForCompletion(true);
35   }
36
37   public static void main(String[] args) throws Exception {
38       runJob(Arrays.copyOfRange(args, 0, args.length-1), args[args.length-1]);
39   }
40
41 }
42
```

EXPLANATION:

The Map Reduce approach with the final part of (**C) Hash Tag Analysis** enables you to find the countries that had been cheered on via tweets that were posted, so each tweet would be compared against a matrix to see whether or not it had contained a cheer tweet of country X or country Y and if it had such content i.e: if gocountryX or teamcountryX was found, it would return country X along with the amount of times a tweet had been written with a cheer associated with it.

Student Name: Hussein Ahmed Tejan
Student Id No.: 130108432

Countries	Tweet Count
china	2
jamaica	5
kenya	3
niger	2
nigeria	2
sweden	1
bahamas	1
bulgaria	1
france	2
poland	4
canada	9
hungary	2
indonesia	2

As you can see from the table displayed above, there were not many countries that had been found that had been cheered/celebrated i.e.: goCountryX or teamCountryX. However, of the countries that were found being cheered on in tweets were the countries above with Canada having the most amount of cheers.