Name: Aditya Bapat
UID: 2019140007
BE IT Batch B

ISE

Dataset:
https://www.kaggle.com/datasets/sudalairajkumar/novel-corona-virus-2019-dataset

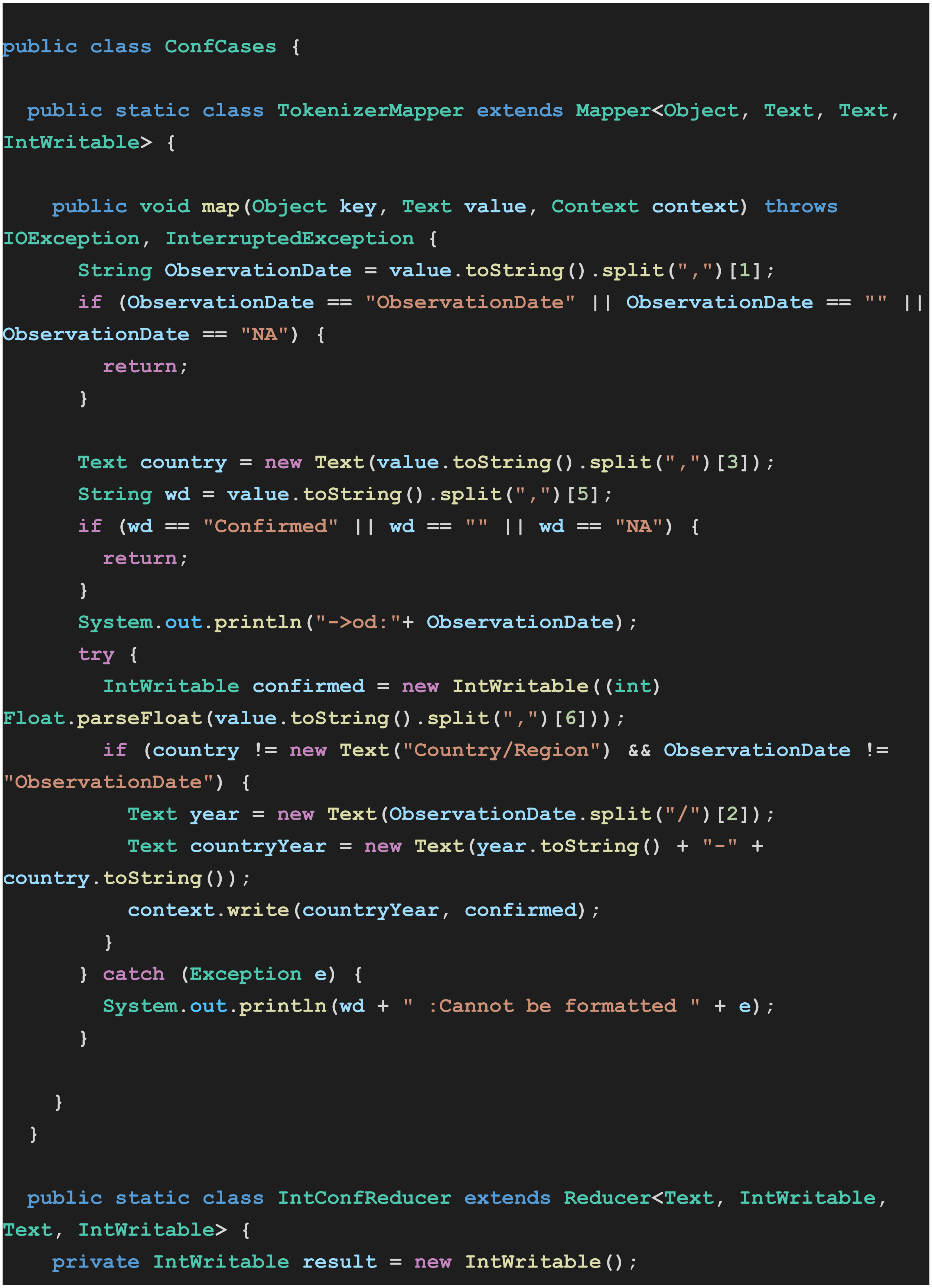
The 2019 Novel Coronavirus (2019-nCoV) is a virus (more specifically, a coronavirus) identified as the cause of an outbreak of respiratory illness first detected in Wuhan, China. Early on, many of the patients in the outbreak in Wuhan, China reportedly had some link to a large seafood and animal market, suggesting animal-to-person spread. However, a growing number of patients reportedly have not had exposure to animal markets, indicating person-to-person spread is occurring. At this time, it’s unclear how easily or sustainably this virus is spreading between people.

This dataset has daily level information on the number of affected cases, deaths and recovery from 2019 novel coronavirus. Please note that this is a time series data and so the number of cases on any given day is the cumulative number.

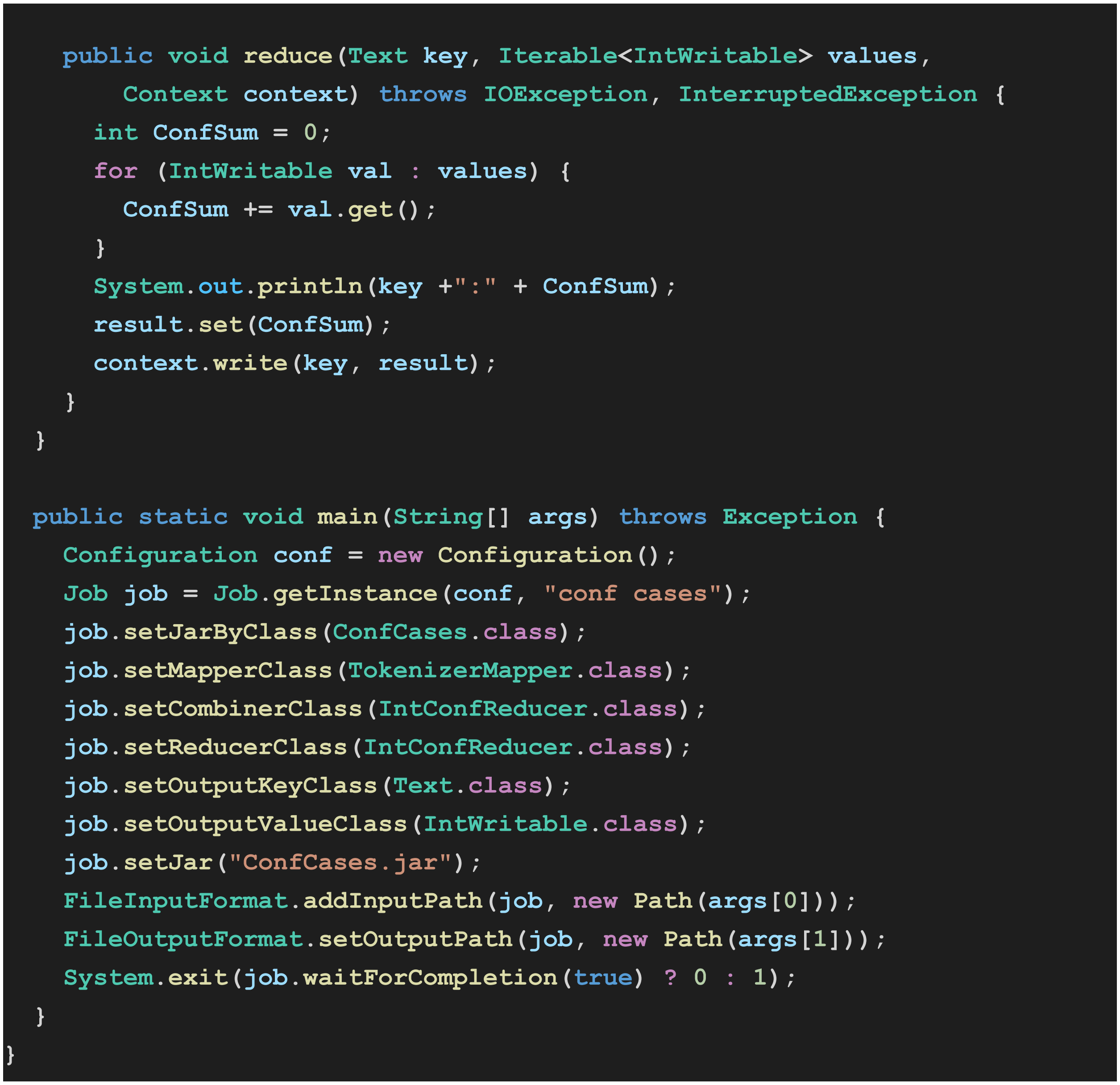
The data is available from 22 Jan, 2020.

Query 1: Country wise confirmed cases over the years:

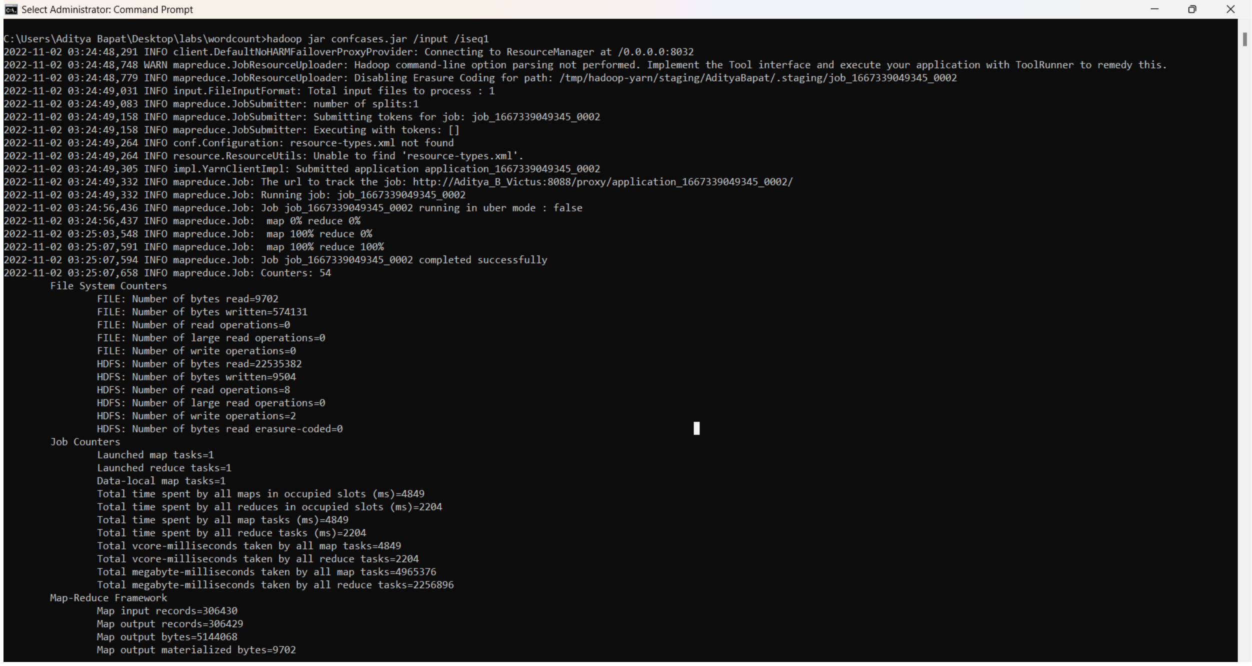
```
import java.io.IOException;
// import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
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;
```



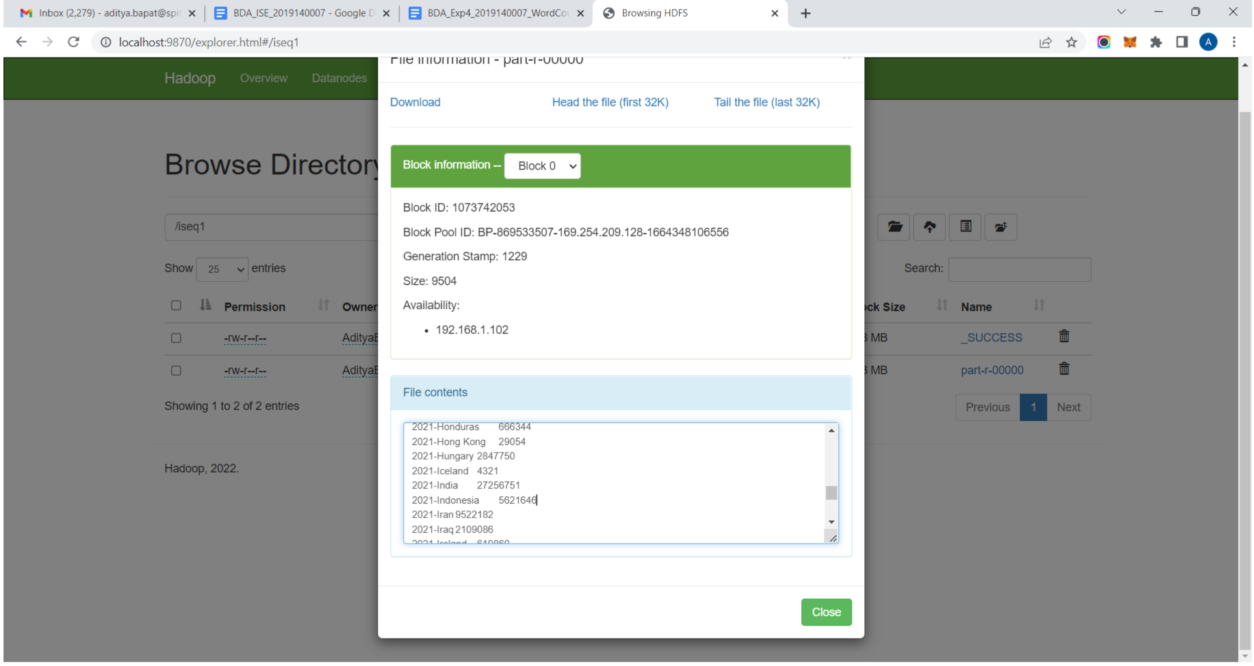
```
public class ConfCases {

  public static class TokenizerMapper extends Mapper<Object, Text, Text,
IntWritable> {

    public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
      String ObservationDate = value.toString().split(",")[1];
      if (ObservationDate == "ObservationDate" || ObservationDate == "" ||
ObservationDate == "NA") {
        return;
      }

      Text country = new Text(value.toString().split(",")[3]);
      String wd = value.toString().split(",")[5];
      if (wd == "Confirmed" || wd == "" || wd == "NA") {
        return;
      }
      System.out.println("->od:"+ ObservationDate);
      try {
        IntWritable confirmed = new IntWritable((int)
Float.parseFloat(value.toString().split(",")[6]));
        if (country != new Text("Country/Region") && ObservationDate !=
"ObservationDate") {
          Text year = new Text(ObservationDate.split("/")[2]);
          Text countryYear = new Text(year.toString() + "-" +
country.toString());
          context.write(countryYear, confirmed);
        }
      } catch (Exception e) {
        System.out.println(wd + " :Cannot be formatted " + e);
      }

    }
  }

  public static class IntConfReducer extends Reducer<Text, IntWritable,
Text, IntWritable> {
    private IntWritable result = new IntWritable();
```



```
    public void reduce(Text key, Iterable<IntWritable> values,
        Context context) throws IOException, InterruptedException {
      int ConfSum = 0;
      for (IntWritable val : values) {
        ConfSum += val.get();
      }
      System.out.println(key +":" + ConfSum);
      result.set(ConfSum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "conf cases");
    job.setJarByClass(ConfCases.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntConfReducer.class);
    job.setReducerClass(IntConfReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setJar("ConfCases.jar");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```



```
C:\Users\Aditya Bapat\Desktop\labs\wordcount>hadoop jar confcases.jar /input /iseq1
2022-11-02 03:24:48,291 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2022-11-02 03:24:48,748 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2022-11-02 03:24:48,779 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/AdityaBapat/.staging/job_1667339049345_0002
2022-11-02 03:24:49,031 INFO input.FileInputFormat: Total input files to process : 1
2022-11-02 03:24:49,083 INFO mapreduce.JobSubmitter: number of splits:1
2022-11-02 03:24:49,158 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1667339049345_0002
2022-11-02 03:24:49,158 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-02 03:24:49,264 INFO conf.Configuration: resource-types.xml not found
2022-11-02 03:24:49,264 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-02 03:24:49,305 INFO impl.YarnClientImpl: Submitted application application_1667339049345_0002
2022-11-02 03:24:49,332 INFO mapreduce.Job: The url to track the job: http://Aditya_B_Victus:8088/proxy/application_1667339049345_0002/
2022-11-02 03:24:49,332 INFO mapreduce.Job: Running job: job_1667339049345_0002
2022-11-02 03:24:56,436 INFO mapreduce.Job: Job job_1667339049345_0002 running in uber mode : false
2022-11-02 03:24:56,437 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-02 03:25:03,548 INFO mapreduce.Job:  map 100% reduce 0%
2022-11-02 03:25:07,591 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-02 03:25:07,594 INFO mapreduce.Job: Job job_1667339049345_0002 completed successfully
2022-11-02 03:25:07,658 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=9702
                FILE: Number of bytes written=574131
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=22535382
                HDFS: Number of bytes written=9504
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=4849
                Total time spent by all reduces in occupied slots (ms)=2204
                Total time spent by all map tasks (ms)=4849
                Total time spent by all reduce tasks (ms)=2204
                Total vcore-milliseconds taken by all map tasks=4849
                Total vcore-milliseconds taken by all reduce tasks=2204
                Total megabyte-milliseconds taken by all map tasks=4965376
                Total megabyte-milliseconds taken by all reduce tasks=2256896
        Map-Reduce Framework
                Map input records=306430
                Map output records=306429
                Map output bytes=5144068
                Map output materialized bytes=9702
```



File information - part-r-00000

Download   Head the file (first 32K)   Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073742053

Block Pool ID: BP-869533507-169.254.209.128-1664348106556

Generation Stamp: 1229

Size: 9504

Availability:

- 192.168.1.102

File contents

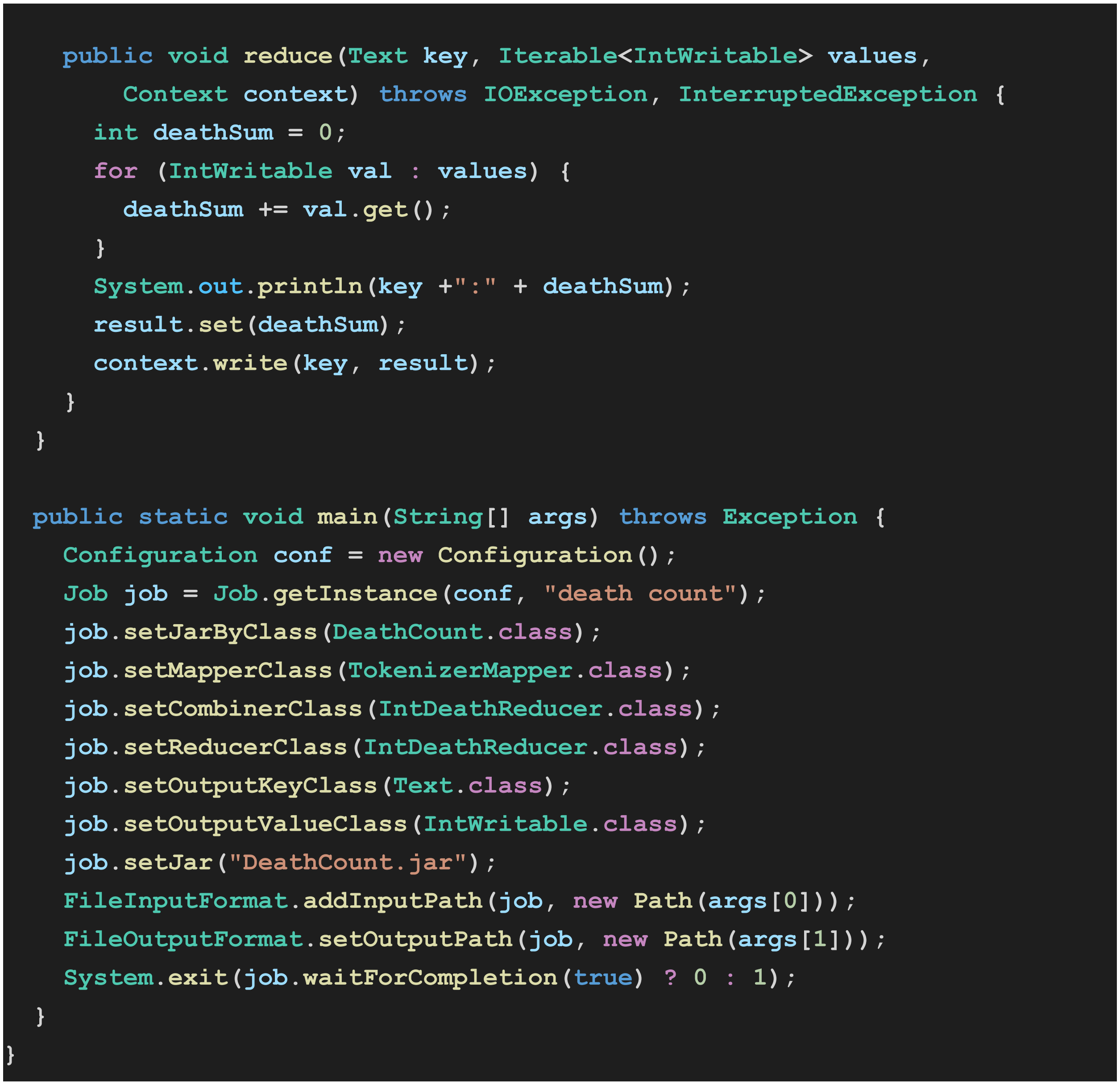
```
2021-Honduras	666344
2021-Hong Kong	29054
2021-Hungary	2847750
2021-Iceland	4321
2021-India	27256751
2021-Indonesia	5621646
2021-Iran	9522182
2021-Iraq	2109086
2021-Ireland	619860
```

Query 2: Find the number of death cases for each country:

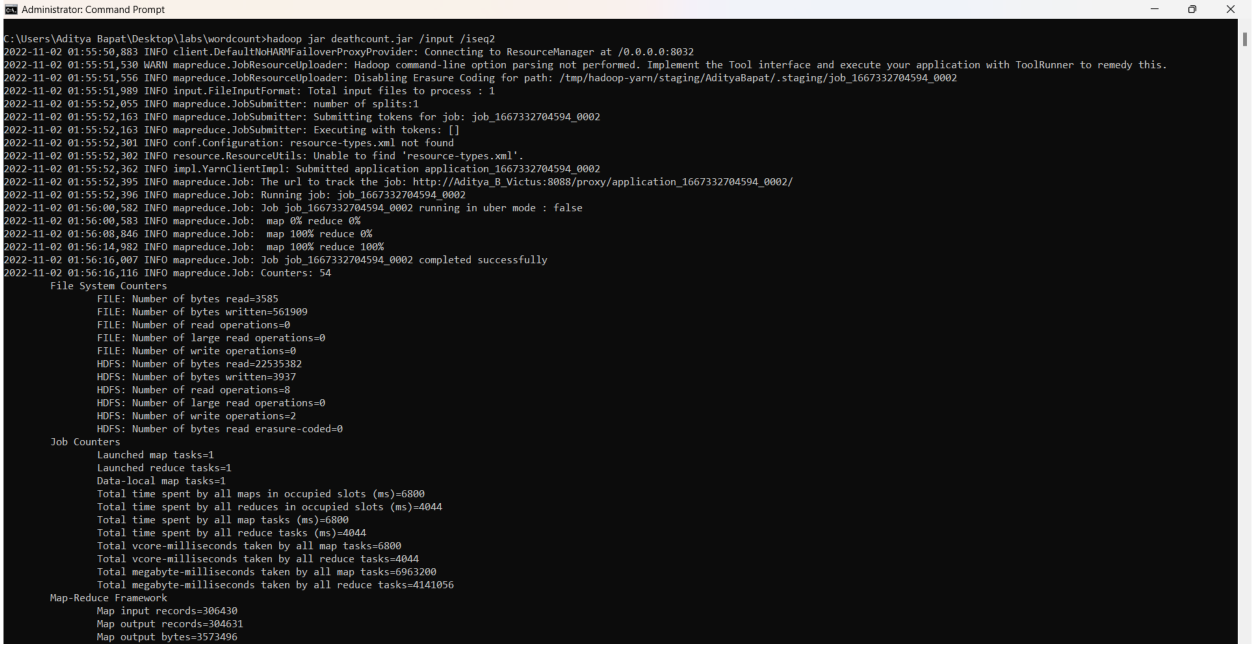
Code:

```
import java.io.IOException;
// import java.util.StringTokenizer;
```

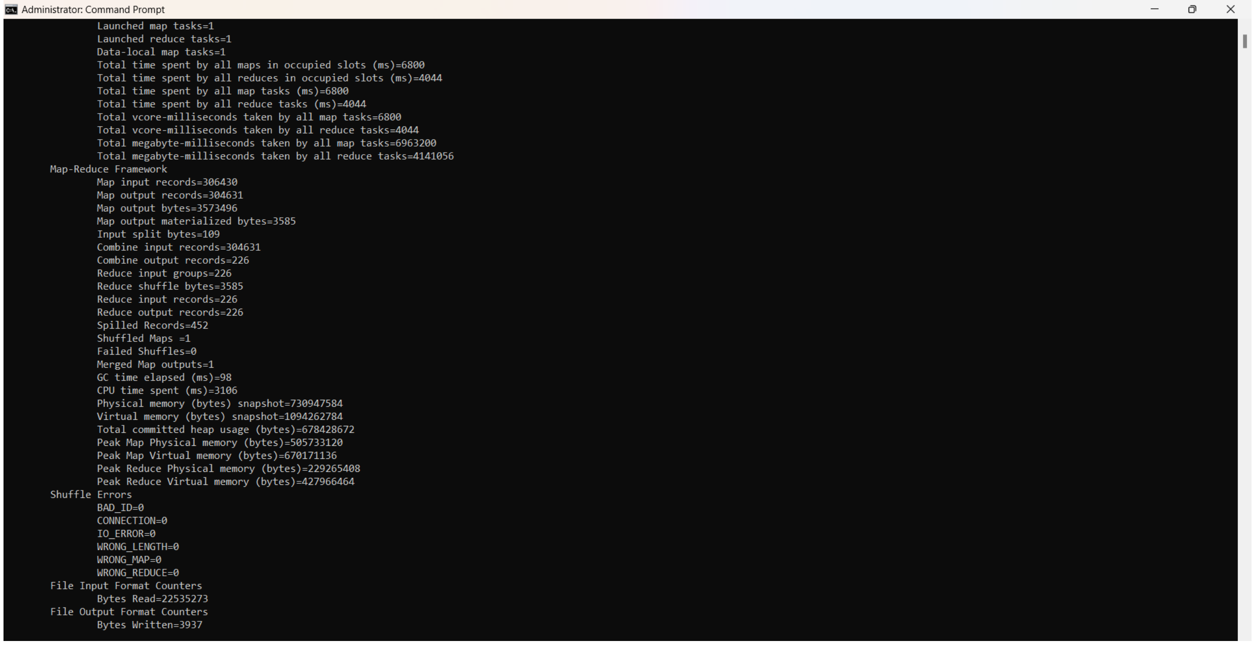
```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DeathCount {

  public static class TokenizerMapper extends Mapper<Object, Text, Text,
IntWritable> {

    public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
      Text country = new Text(value.toString().split(",")[3]);
      String wd = value.toString().split(",")[6];
      if (wd == "Confirmed" || wd == "" || wd == "NA") {
        return;
      }
      System.out.println("->wd:"+ wd);
      try {
        IntWritable confirmed = new IntWritable((int)
Float.parseFloat(value.toString().split(",")[6]));
        if (country != new Text("Country/Region")) {
          context.write(country, confirmed);
        }
      } catch (Exception e) {
        System.out.println(wd + " :Cannot be formatted " + e);
      }

    }
  }

  public static class IntDeathReducer extends Reducer<Text, IntWritable,
Text, IntWritable> {
    private IntWritable result = new IntWritable();
```



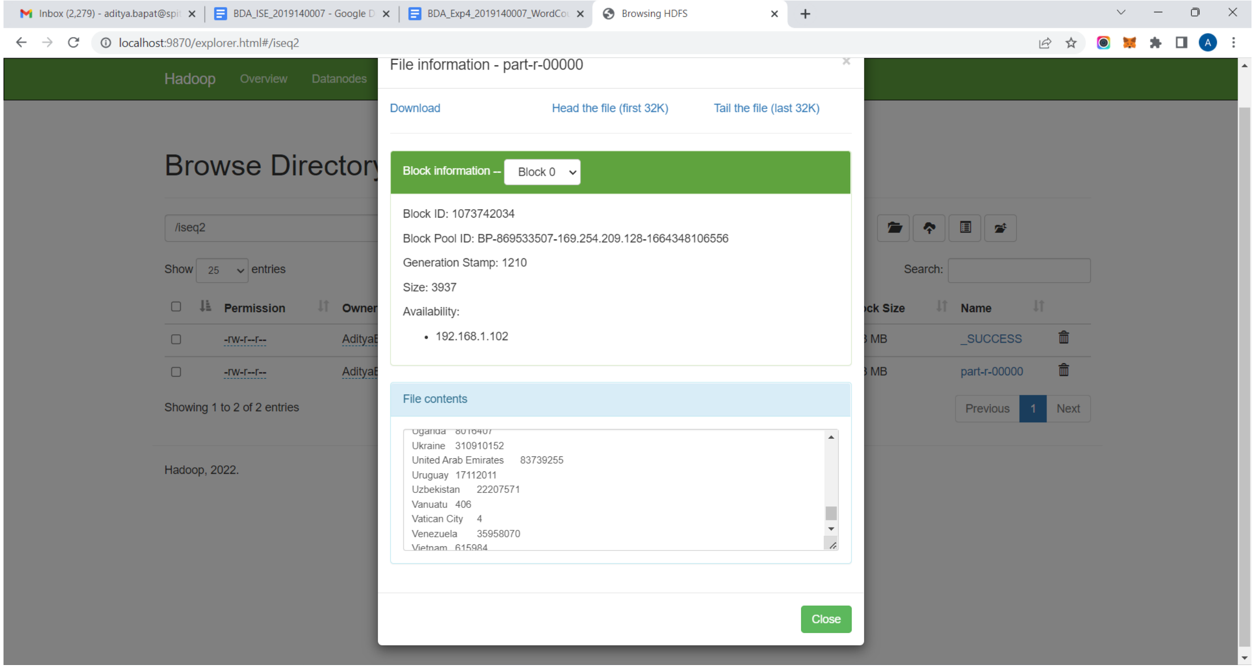
```
    public void reduce(Text key, Iterable<IntWritable> values,
        Context context) throws IOException, InterruptedException {
      int deathSum = 0;
      for (IntWritable val : values) {
        deathSum += val.get();
      }
      System.out.println(key +":" + deathSum);
      result.set(deathSum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "death count");
    job.setJarByClass(DeathCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntDeathReducer.class);
    job.setReducerClass(IntDeathReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setJar("DeathCount.jar");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```



```
C:\Users\Aditya Bapat\Desktop\labs\wordcount>hadoop jar deathcount.jar /input /iseq2
2022-11-02 01:55:50,883 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2022-11-02 01:55:51,530 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2022-11-02 01:55:51,556 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/AdityaBapat/.staging/job_1667332704594_0002
2022-11-02 01:55:51,989 INFO input.FileInputFormat: Total input files to process : 1
2022-11-02 01:55:52,055 INFO mapreduce.JobSubmitter: number of splits:1
2022-11-02 01:55:52,163 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1667332704594_0002
2022-11-02 01:55:52,163 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-02 01:55:52,301 INFO conf.Configuration: resource-types.xml not found
2022-11-02 01:55:52,302 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-02 01:55:52,362 INFO impl.YarnClientImpl: Submitted application application_1667332704594_0002
2022-11-02 01:55:52,395 INFO mapreduce.Job: The url to track the job: http://Aditya_B_Victus:8088/proxy/application_1667332704594_0002/
2022-11-02 01:55:52,396 INFO mapreduce.Job: Running job: job_1667332704594_0002
2022-11-02 01:56:00,582 INFO mapreduce.Job: Job job_1667332704594_0002 running in uber mode : false
2022-11-02 01:56:00,583 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-02 01:56:08,846 INFO mapreduce.Job:  map 100% reduce 0%
2022-11-02 01:56:14,982 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-02 01:56:16,007 INFO mapreduce.Job: Job job_1667332704594_0002 completed successfully
2022-11-02 01:56:16,116 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=3585
                FILE: Number of bytes written=561909
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=22535382
                HDFS: Number of bytes written=3937
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6800
                Total time spent by all reduces in occupied slots (ms)=4044
                Total time spent by all map tasks (ms)=6800
                Total time spent by all reduce tasks (ms)=4044
                Total vcore-milliseconds taken by all map tasks=6800
                Total vcore-milliseconds taken by all reduce tasks=4044
                Total megabyte-milliseconds taken by all map tasks=6963200
                Total megabyte-milliseconds taken by all reduce tasks=4141056
        Map-Reduce Framework
                Map input records=306430
                Map output records=304631
                Map output bytes=3573496
```



```
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6800
                Total time spent by all reduces in occupied slots (ms)=4044
                Total time spent by all map tasks (ms)=6800
                Total time spent by all reduce tasks (ms)=4044
                Total vcore-milliseconds taken by all map tasks=6800
                Total vcore-milliseconds taken by all reduce tasks=4044
                Total megabyte-milliseconds taken by all map tasks=6963200
                Total megabyte-milliseconds taken by all reduce tasks=4141056
        Map-Reduce Framework
                Map input records=306430
                Map output records=304631
                Map output bytes=3573496
                Map output materialized bytes=3585
                Input split bytes=109
                Combine input records=304631
                Combine output records=226
                Reduce input groups=226
                Reduce shuffle bytes=3585
                Reduce input records=226
                Reduce output records=226
                Spilled Records=452
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=98
                CPU time spent (ms)=3106
                Physical memory (bytes) snapshot=730947584
                Virtual memory (bytes) snapshot=1094262784
                Total committed heap usage (bytes)=678428672
                Peak Map Physical memory (bytes)=505733120
                Peak Map Virtual memory (bytes)=670171136
                Peak Reduce Physical memory (bytes)=229265408
                Peak Reduce Virtual memory (bytes)=427966464
        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=22535273
        File Output Format Counters
                Bytes Written=3937
```



File information - part-r-00000

Download   Head the file (first 32K)   Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073742034

Block Pool ID: BP-869533507-169.254.209.128-1664348106556

Generation Stamp: 1210

Size: 3937

Availability:

- 192.168.1.102

File contents

```
Uganda    8016407
Ukraine   310910152
United Arab Emirates    83739255
Uruguay  17112011
Uzbekistan     22207571
Vanuatu   406
Vatican City    4
Venezuela      35958070
Vietnam   615984
```

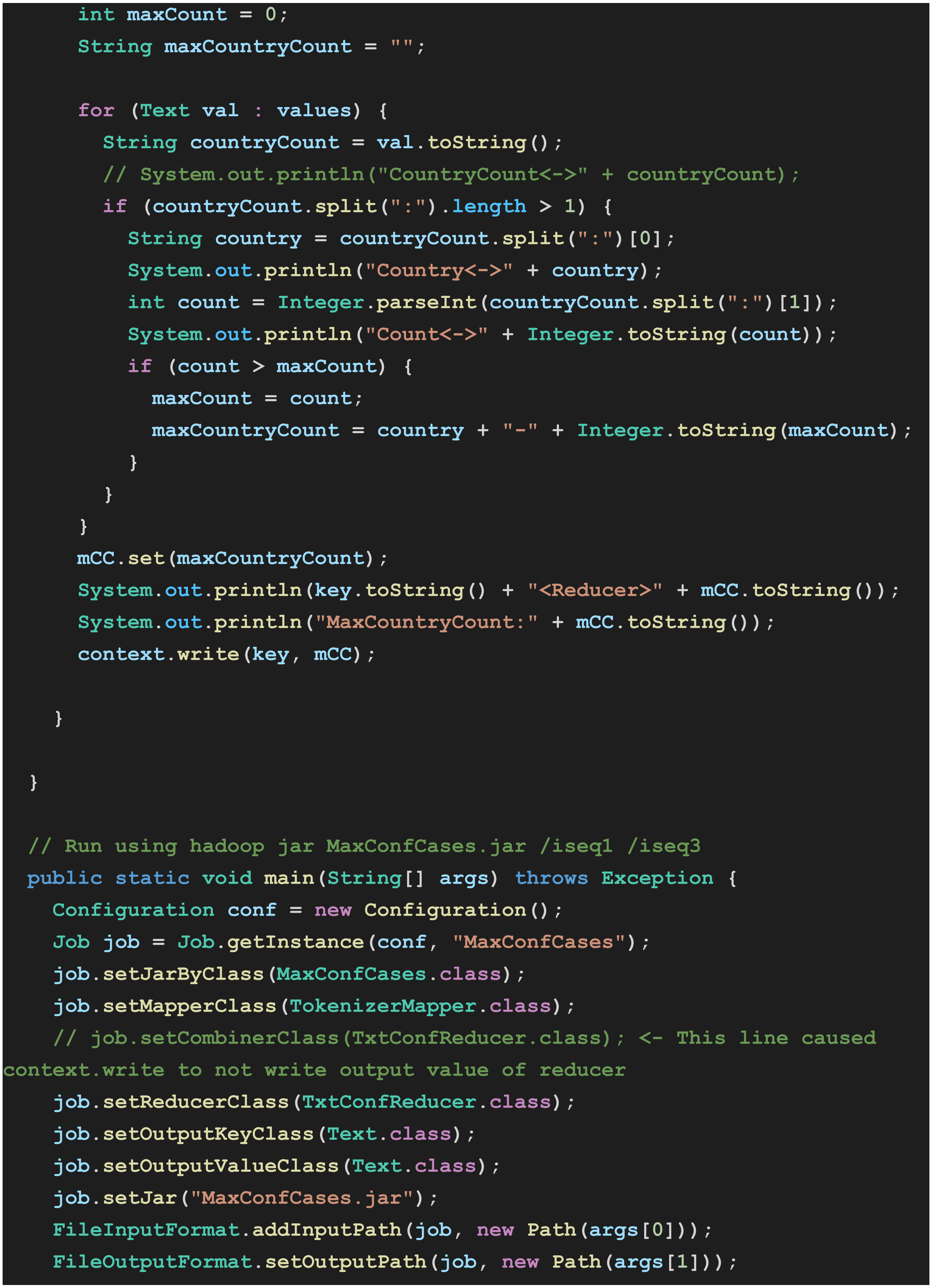
Close

ISE Query:

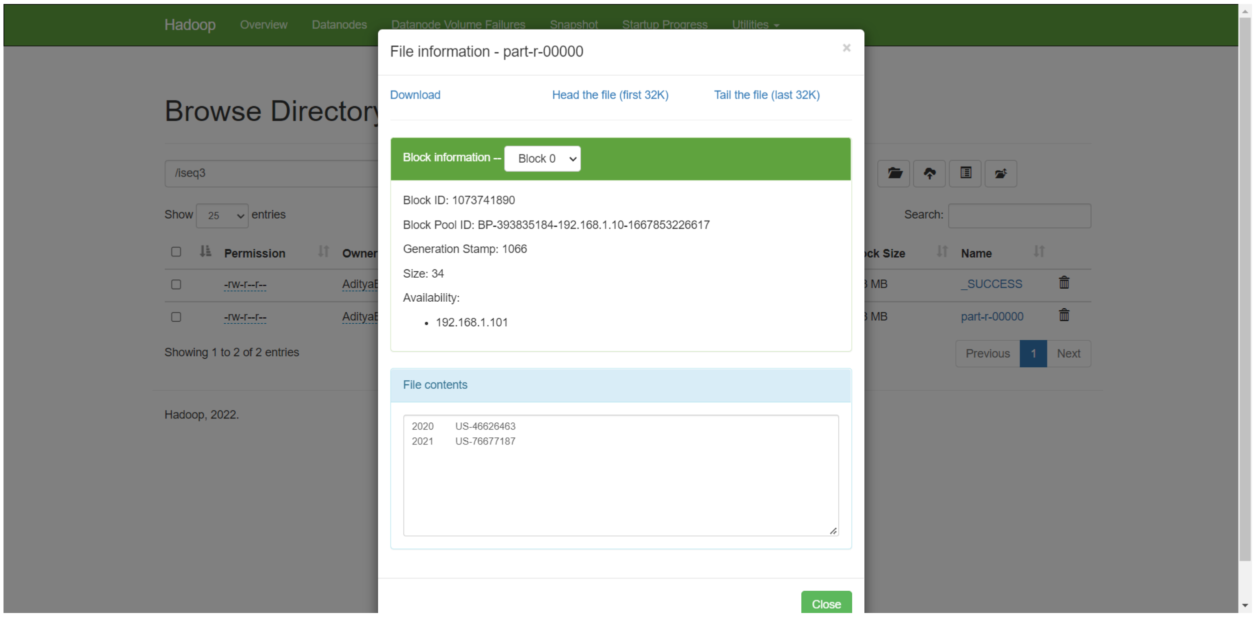
Find the max confirmed cases and their country for each of the years.

Code:

```java
import java.io.IOException;
// import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
// import org.apache.hadoop.io.IntWritable;
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 MaxConfCases {

  public static class TokenizerMapper extends Mapper<Object, Text, Text,
Text> {
    public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
      // gets input from confcases reducer output: eg 2021-Hong Kong 29054
(hdfs file
      // /iseq1)

      String year = "2022";
      String countryCount = "Aditya:1";
      try {
        String line = value.toString();
        // System.out.println("line:" + line);
        String yearCountry = line.split("\t")[0];
        // System.out.println("yearCountry:" + yearCountry);
        if (yearCountry.split("-").length > 1) {
          year = yearCountry.split("-")[0];
          // System.out.println("year:" + year);
          String country = yearCountry.split("-")[1];
          // System.out.println("Country1:" + country);
          String count = line.split("\t")[1];
          // System.out.println("Count1:" + count);
          countryCount = country + ":" + count;
        }
        // System.out.println("year:Country:Count (Mapper) <->" + year + "
: " +
        // countryCount);
        context.write(new Text(year), new Text(countryCount));
      } catch (ArrayIndexOutOfBoundsException e) {
        // ArrayIndexOutOfBoundsException
        System.out.println("Error1: " + e);
      }
    }
  }

  public static class TxtConfReducer extends Reducer<Text, Text, Text,
Text> {
    private Text mCC = new Text();

    public void reduce(Text key, Iterable<Text> values,
        Context context) throws IOException, InterruptedException {
```



```
      int maxCount = 0;
      String maxCountryCount = "";

      for (Text val : values) {
        String countryCount = val.toString();
        // System.out.println("CountryCount<->" + countryCount);
        if (countryCount.split(":").length > 1) {
          String country = countryCount.split(":")[0];
          System.out.println("Country<->" + country);
          int count = Integer.parseInt(countryCount.split(":")[1]);
          System.out.println("Count<->" + Integer.toString(count));
          if (count > maxCount) {
            maxCount = count;
            maxCountryCount = country + "-" + Integer.toString(maxCount);
          }
        }
      }
      mCC.set(maxCountryCount);
      System.out.println(key.toString() + "<Reducer>" + mCC.toString());
      System.out.println("MaxCountryCount:" + mCC.toString());
      context.write(key, mCC);

    }

  }

  // Run using hadoop jar MaxConfCases.jar /iseq1 /iseq3
  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "MaxConfCases");
    job.setJarByClass(MaxConfCases.class);
    job.setMapperClass(TokenizerMapper.class);
    // job.setCombinerClass(TxtConfReducer.class); <- This line caused
context.write to not write output value of reducer
    job.setReducerClass(TxtConfReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(Text.class);
    job.setJar("MaxConfCases.jar");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```



File information - part-r-00000

Permission	Owner	ck Size	Name
-rw-r--r--	AdityaB	3 MB	_SUCCESS
-rw-r--r--	AdityaB	3 MB	part-r-00000

Block ID: 1073741890

Block Pool ID: BP-393835184-192.168.1.10-1667853226617

Generation Stamp: 1066

Size: 34

Availability:

- 192.168.1.101

```
2020	US-46626463
2021	US-76677187
```

Conclusion: Hence, we found the max confirmed cases and their country for each of the years.