

*****Problem Statement*****

We have a dataset of sales of different TV sets across different locations.

Records look like:

Samsung|Optima|14|Madhya Pradesh|132401|14200

The fields are arranged like:

Company Name|Product Name|Size in inches|State|Pin Code|Price

There are some invalid records which contain 'NA' in either Company Name or Product Name.

Task 1:

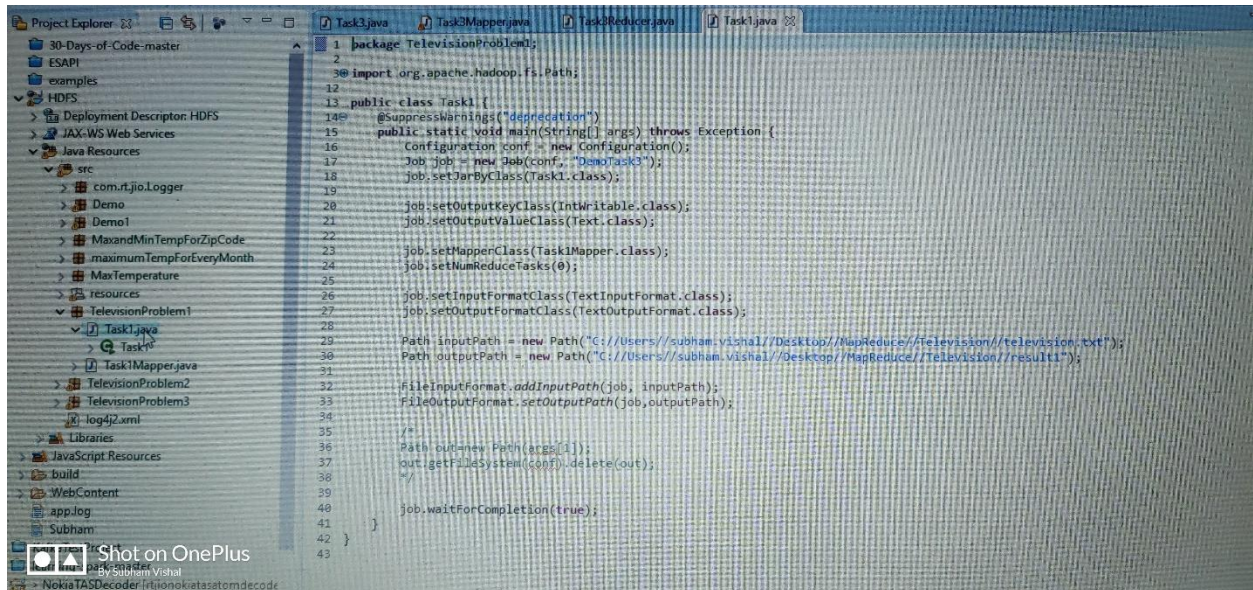
Write a Map Reduce program to filter out the invalid records. Map only job will fit for this context.

Solution:

The ScreenShots of the problem is :



The Driver Class :



```
1 package TelevisionProblem1;
2
3 import org.apache.hadoop.fs.Path;
4
5 public class Task1 {
6     @SuppressWarnings("deprecation")
7     public static void main(String[] args) throws Exception {
8         Configuration conf = new Configuration();
9         Job job = new Job(conf, "DemoTask1");
10        job.setJarByClass(Task1.class);
11
12        job.setOutputKeyClass(IntWritable.class);
13        job.setOutputValueClass(Text.class);
14
15        job.setMapperClass(Task1Mapper.class);
16        job.setNumReduceTasks(0);
17
18        job.setInputFormatClass(TextInputFormat.class);
19        job.setOutputFormatClass(TextOutputFormat.class);
20
21        Path inputPath = new Path("C:/Users/subham.vishal/Desktop/MapReduce/Television/television.txt");
22        Path outputPath = new Path("C:/Users/subham.vishal/Desktop/MapReduce/Television/result1");
23
24        FileInputFormat.addInputPath(job, inputPath);
25        FileOutputFormat.setOutputPath(job, outputPath);
26
27        /*
28         * Path out = new Path(args[1]);
29         * out.getFileSystem(conf).delete(out);
30         */
31
32        job.waitForCompletion(true);
33    }
34 }
```

The Mapper Class :

```

1 package TelevisionProblem1;
2
3 import java.io.*;
4
5
6
7
8
9
10 public class TaskMapper extends Mapper {
11
12     String company;
13     String model;
14     FileWriter fw;
15
16     @Override
17     public void setup(Context context) {
18         company = new String();
19         model = new String();
20     }
21
22     @Override
23     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
24         String[] lineArray = value.toString().split("\\|");
25         String ti = "";
26
27         company = lineArray[0];
28         model = lineArray[1];
29
30         if ((company.equalsIgnoreCase("NA")) || (model.equalsIgnoreCase("NA"))) {
31             System.out.println(company + " " + model);
32         } else {
33             for (int i = 0; i < lineArray.length - 1; i++) {
34                 ti += lineArray[i] + "|";
35             }
36             ti = ti + lineArray[lineArray.length - 1];
37             context.write(null, ti);
38         }
39     }
40 }
41
42

```

The original Data Set:

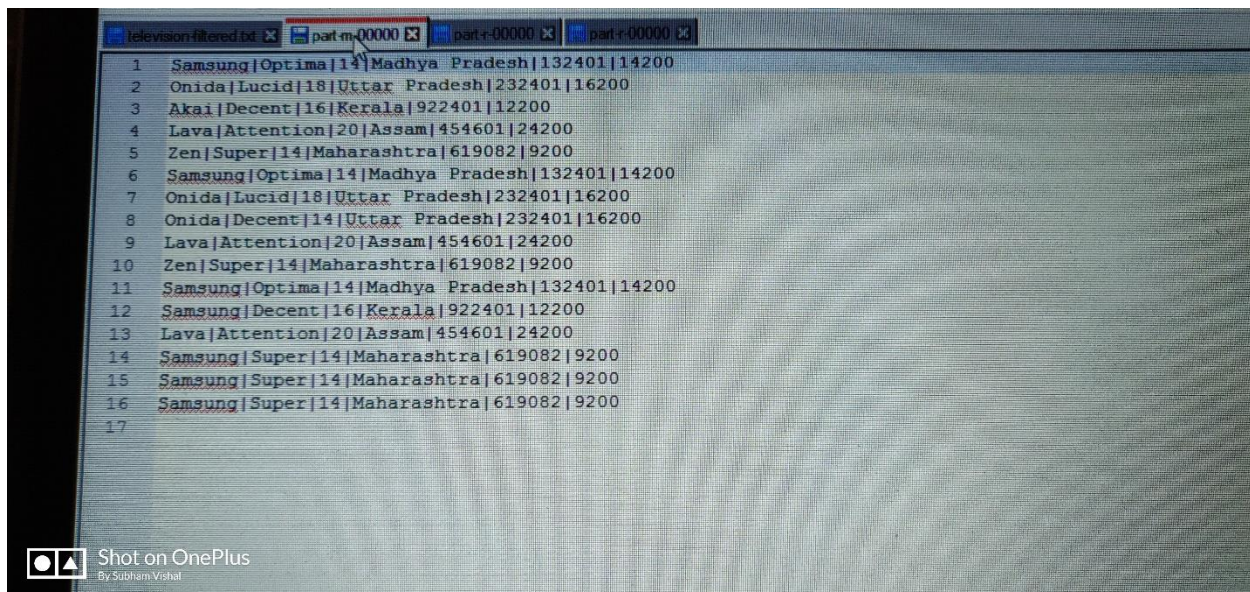
```

television - Notepad
File Edit Format View Help

Samsung|Optima|14|Madhya Pradesh|132401|14200
Onida|Lucid|18|Uttar Pradesh|232401|16200
Akai|Decent|16|Kerala|922401|12200
Lava|Attention|20|Assam|454601|24200
Zen|Super|14|Maharashtra|619082|9200
Samsung|Optima|14|Madhya Pradesh|132401|14200
Onida|Lucid|18|Uttar Pradesh|232401|16200
Onida|Decent|14|Uttar Pradesh|232401|16200
Onida|NA|16|Kerala|922401|12200
Lava|Attention|20|Assam|454601|24200
Zen|Super|14|Maharashtra|619082|9200
Samsung|Optima|14|Madhya Pradesh|132401|14200
NA|Lucid|18|Uttar Pradesh|232401|16200
Samsung|Decent|16|Kerala|922401|12200
Lava|Attention|20|Assam|454601|24200
Samsung|Super|14|Maharashtra|619082|9200
Samsung|Super|14|Maharashtra|619082|9200
Samsung|Super|14|Maharashtra|619082|9200

```

The Filtered DataSet:



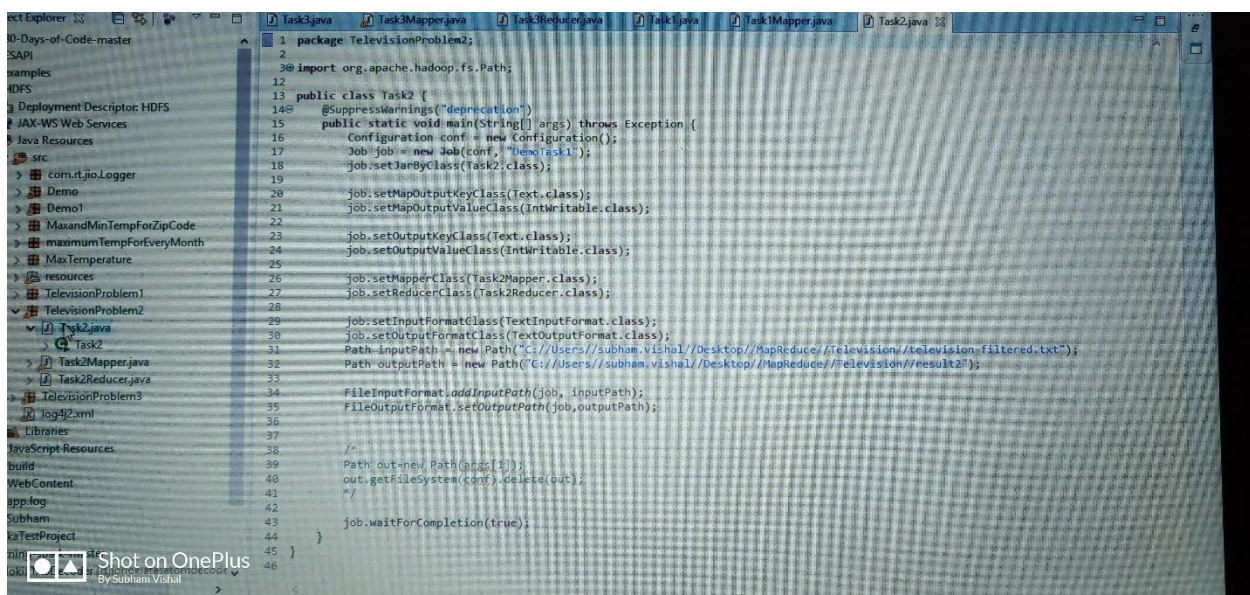
This dataset has all the lines with NA values removed.

Task 2:

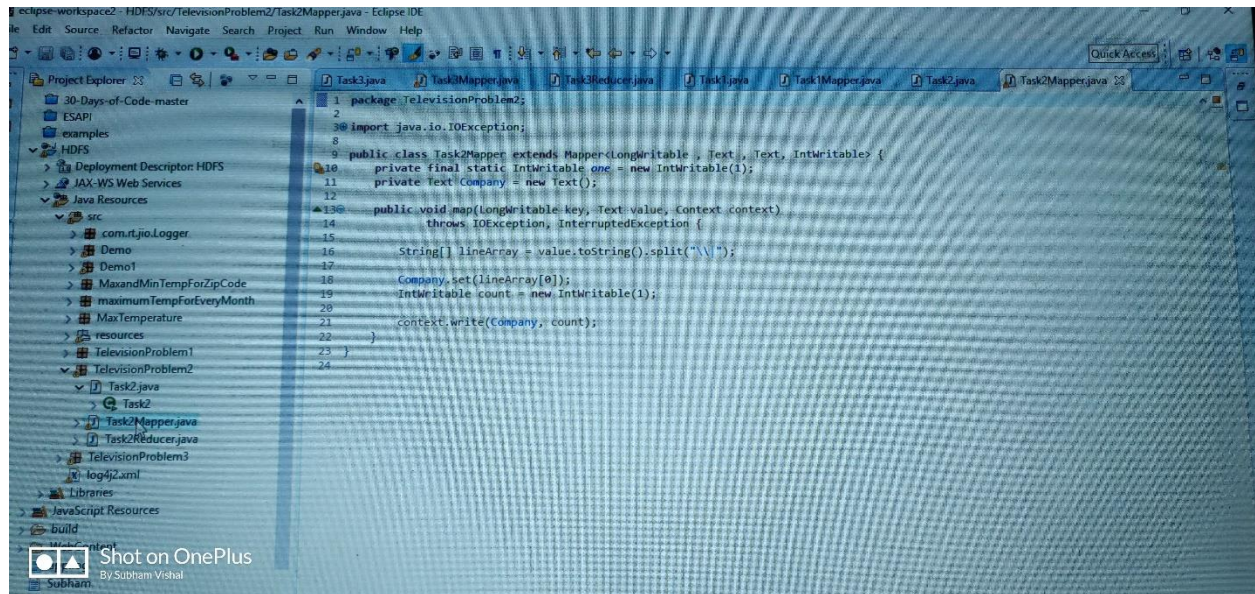
Write a Map Reduce program to calculate the total units sold for each Company.

Solution →

The Driver class :

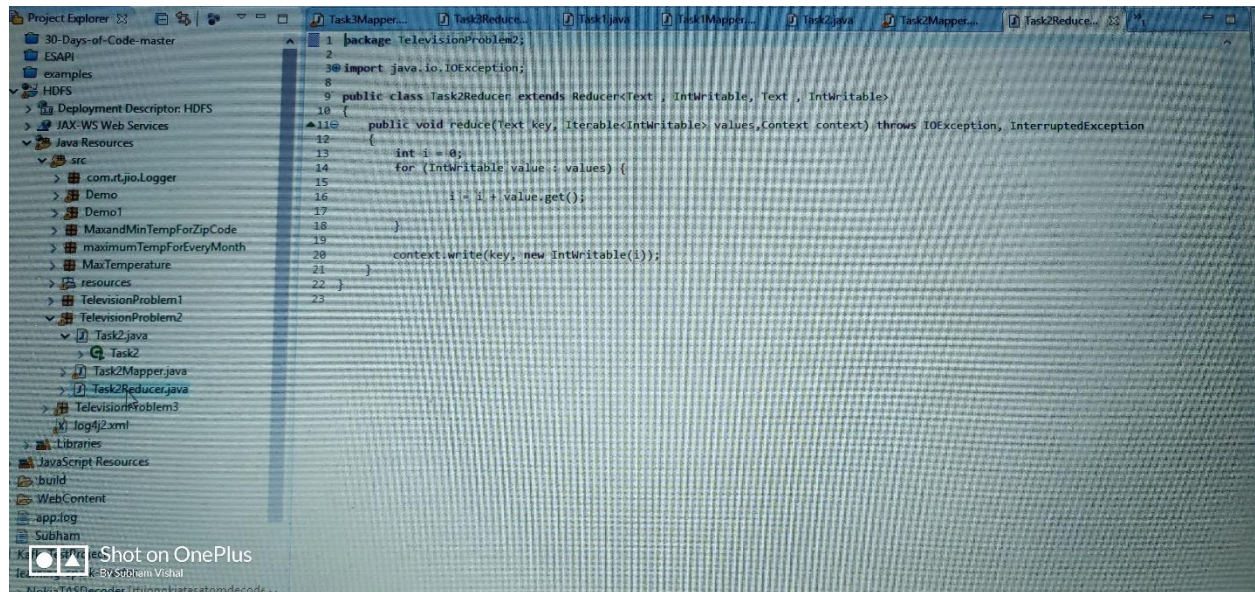


The Mapper Class :



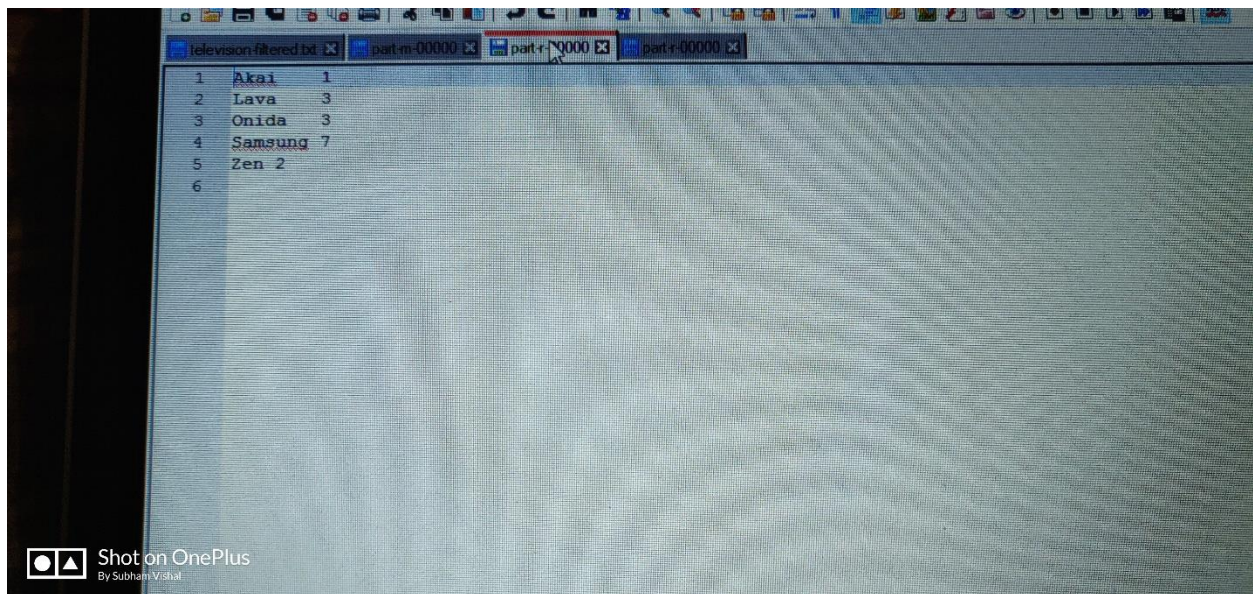
```
1 package TelevisionProblem2;
2
3 import java.io.IOException;
4
5 public class Task2Mapper extends Mapper<LongWritable, Text, Text, IntWritable> {
6     private final static IntWritable one = new IntWritable(1);
7     private Text company = new Text();
8
9     public void map(LongWritable key, Text value, Context context)
10         throws IOException, InterruptedException {
11         String[] lineArray = value.toString().split("\\|");
12         company.set(lineArray[0]);
13         IntWritable count = new IntWritable(1);
14         context.write(company, count);
15     }
16 }
```

The Reducer Class :



```
1 package TelevisionProblem2;
2
3 import java.io.IOException;
4
5 public class Task2Reducer extends Reducer<Text, IntWritable, Text, IntWritable> {
6     public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
7         int i = 0;
8         for (IntWritable value : values) {
9             i = i + value.get();
10         }
11         context.write(key, new IntWritable(i));
12     }
13 }
```

Final Result:

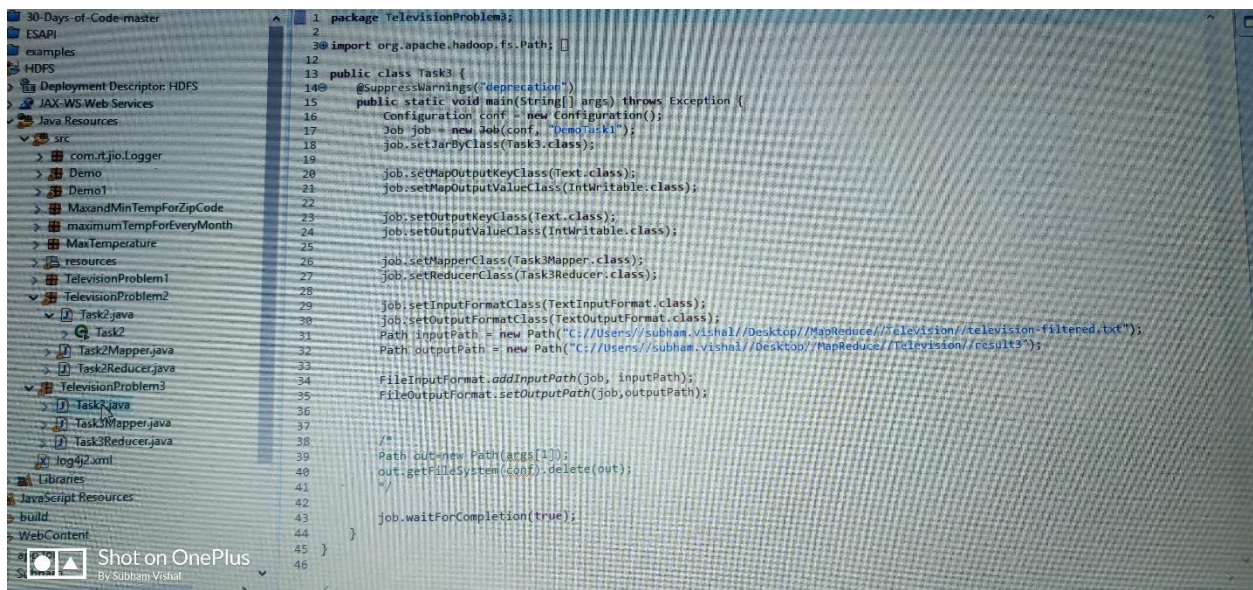


Task 3:

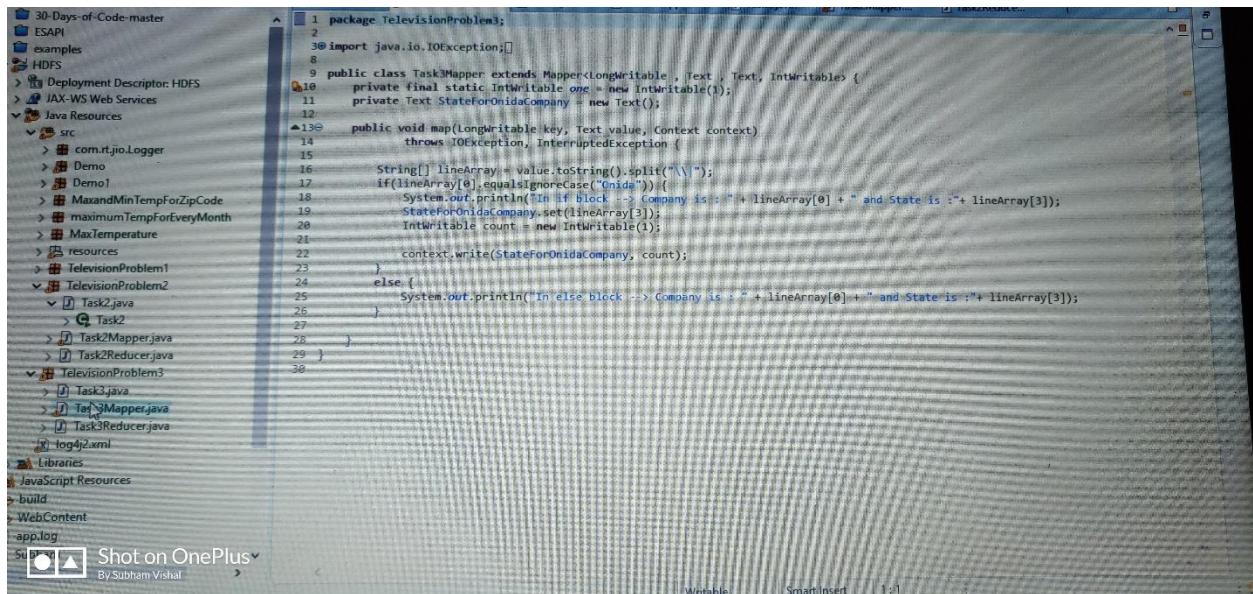
Write a Map Reduce program to calculate the total units sold in each state for Onida company.

Solution →

The Driver class :

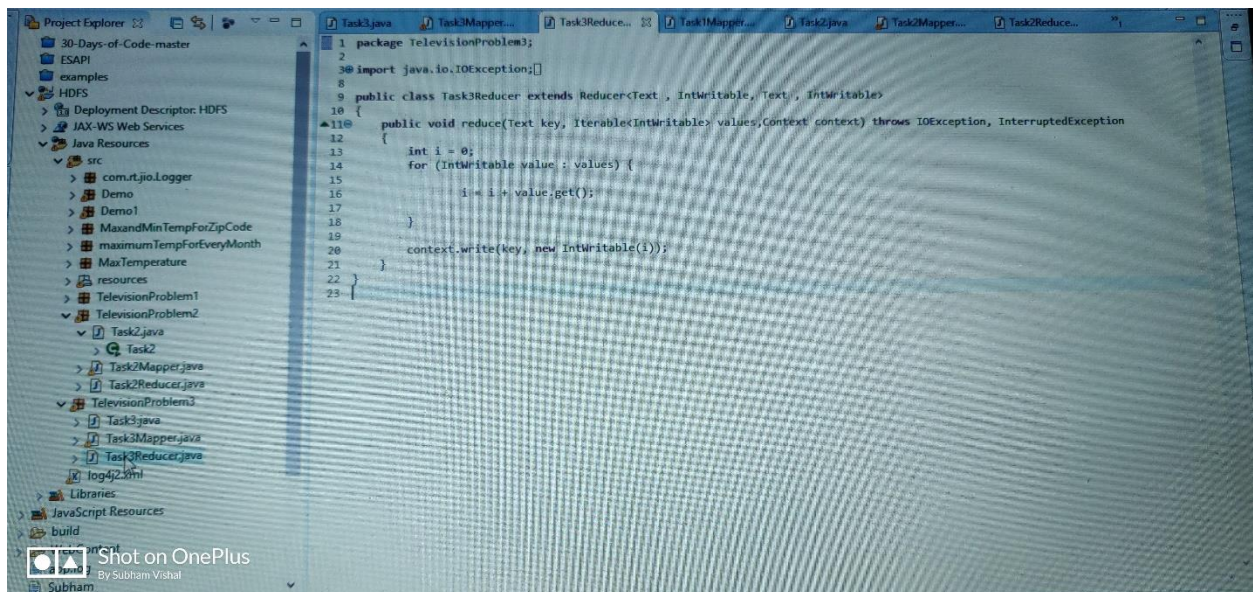


The Mapper Class :



```
1 package TelevisionProblem3;
2
3 import java.io.IOException;
4
5 public class Task3Mapper extends Mapper<LongWritable, Text, Text, IntWritable> {
6     private final static IntWritable one = new IntWritable(1);
7     private Text StateForOnidaCompany = new Text();
8
9     public void map(LongWritable key, Text value, Context context)
10         throws IOException, InterruptedException {
11         String[] lineArray = value.toString().split("\\\\");
12         if(lineArray[0].equalsIgnoreCase("Onida")) {
13             System.out.println("In if block -> Company is : " + lineArray[0] + " and State is : " + lineArray[3]);
14             StateForOnidaCompany.set(lineArray[3]);
15             IntWritable count = new IntWritable(1);
16             context.write(StateForOnidaCompany, count);
17         }
18         else {
19             System.out.println("In else block -> Company is : " + lineArray[0] + " and State is : " + lineArray[3]);
20         }
21     }
22 }
23
24
25
26
27
28
29
30
```

The Reducer Class :



```
1 package TelevisionProblem3;
2
3 import java.io.IOException;
4
5 public class Task3Reducer extends Reducer<Text, IntWritable, Text, IntWritable> {
6     {
7         int i = 0;
8         for (IntWritable value : values) {
9             i = i + value.get();
10         }
11         context.write(key, new IntWritable(i));
12     }
13 }
14
15
16
17
18
19
20
21
22
23
```

Final Result:

