Task 1:
Write a Map Reduce program to filter out the invalid records. Map only job will fit for this context.

Program

```
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


import java.io.IOException;
import java.util.logging.Logger;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.TextInputFormat;

public class TvSales {

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

    public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException{

         if(recordIsInvalid(value)==false){
           Text record = new Text();
              record = value;
           context.write(key,record);
         }
       }

       private boolean recordIsInvalid(Text record){

      String[] lineArray = record.toString().split("\\|");
         boolean isInvalid = false;
         for(int i=0;i<lineArray.length;i++){
           if(lineArray[i].equals("NA")){
             isInvalid = true;
           }
         }
         return isInvalid;
  }

       public static void main(String[] args) throws Exception{

         Configuration conf = new Configuration();

         Job job = Job.getInstance(conf, "Tv Sales Invalid REcords");
```

```
        job.setJarByClass(TvSales.class);

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

        job.setMapperClass(TvSalesMapper.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

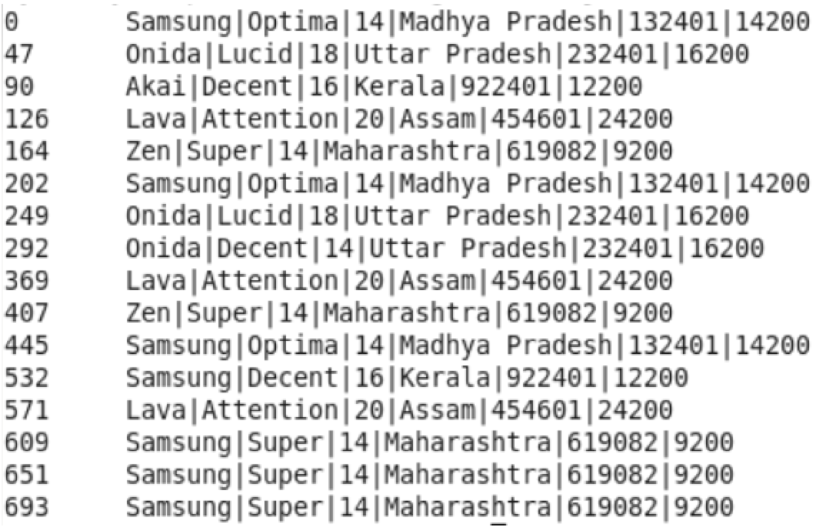
Export to a jar file in eclipse and type at command prompt

```
hadoop jar /home/acadgild/workspace/TvSales.jar television.txt
/skip-bad-records
```

To check the results type

```
hadoop fs –cat /filter-bad-records/part-r-00000
```

Output



```
0       Samsung|Optima|14|Madhya Pradesh|132401|14200
47      Onida|Lucid|18|Uttar Pradesh|232401|16200
90      Akai|Decent|16|Kerala|922401|12200
126     Lava|Attention|20|Assam|454601|24200
164     Zen|Super|14|Maharashtra|619082|9200
202     Samsung|Optima|14|Madhya Pradesh|132401|14200
249     Onida|Lucid|18|Uttar Pradesh|232401|16200
292     Onida|Decent|14|Uttar Pradesh|232401|16200
369     Lava|Attention|20|Assam|454601|24200
407     Zen|Super|14|Maharashtra|619082|9200
445     Samsung|Optima|14|Madhya Pradesh|132401|14200
532     Samsung|Decent|16|Kerala|922401|12200
571     Lava|Attention|20|Assam|454601|24200
609     Samsung|Super|14|Maharashtra|619082|9200
651     Samsung|Super|14|Maharashtra|619082|9200
693     Samsung|Super|14|Maharashtra|619082|9200
```

Task 2:
Write a Map Reduce program to calculate the total units sold for each Company.

Program

```
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


import java.io.IOException;
import java.util.logging.Logger;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.TextInputFormat;

public class TvSales {

    public static class TvSalesMapper extends Mapper<LongWritable, Text, Text, IntWritable>{

        public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException{

         if(recordIsInvalid(value)==false){
          Text company = new Text();
            IntWritable unit = new IntWritable();
            company = new Text(value.toString().split("\\|")[0]);
            unit = new IntWritable(1);
            context.write(company, unit );
         }

        }

        private boolean recordIsInvalid(Text record){

            String[] lineArray = record.toString().split("\\|");
            boolean isInvalid = false;
      for(int i=0;i<lineArray.length;i++){
        if(lineArray[i].equals("NA")){
            isInvalid = true;
        }
      }
      return isInvalid;
        }
    }
    public static class TvSalesReducer 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 sum = 0;
                              for(IntWritable val: values){
                                        sum += val.get();
                              }
                              result.set(sum);
                              context.write(key, result);
                    }
          }

          public static void main(String[] args) throws Exception{

             Configuration conf = new Configuration();

             Job job = Job.getInstance(conf, "Tv Sales Invalid REcords");
             job.setJarByClass(TvSales.class);

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

             job.setOutputKeyClass(Text.class);
             job.setOutputValueClass(IntWritable.class);

             job.setMapperClass(TvSalesMapper.class);
             job.setCombinerClass(TvSalesReducer.class);
             job.setReducerClass(TvSalesReducer.class);

             FileInputFormat.addInputPath(job, new Path(args[0]));
             FileOutputFormat.setOutputPath(job, new Path(args[1]));

             System.exit(job.waitForCompletion(true) ? 0 : 1);
          }
}
```

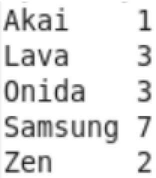
Export to a jar file in eclipse and type at command prompt

```
hadoop jar /home/acadgild/workspace/TvSales.jar television.txt
/total-units-sold
```

To check the results type

```
hadoop fs –cat /total-units-sold/part-r-00000
```

Output



```
Akai    1
Lava    3
Onida   3
Samsung 7
Zen     2
```

Task 3:
Write a Map Reduce program to calculate the total units sold in each state for Onida company.

Program

```
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


import java.io.IOException;
import java.util.logging.Logger;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.TextInputFormat;

public class TvSales {

        public static class TvSalesMapper extends Mapper<LongWritable, Text, Text, IntWritable>{

                public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException{

                  if(recordIsInvalid(value)==false & value.toString().split("\\|")[0].equals("Onida")){
                    Text state = new Text();
                        IntWritable unit = new IntWritable();
                        state = new Text(value.toString().split("\\|")[3]);
                        unit = new IntWritable(1);
                        context.write(state, unit );
                  }

                }

                private boolean recordIsInvalid(Text record){

                        String[] lineArray = record.toString().split("\\|");
                        boolean isInvalid = false;
            for(int i=0;i<lineArray.length;i++){
                if(lineArray[i].equals("NA")){
                        isInvalid = true;
                }
            }
            return isInvalid;
                }
        }
        public static class TvSalesReducer 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 sum = 0;
                        for(IntWritable val: values){
                                sum += val.get();
                        }
                        result.set(sum);
                        context.write(key, result);
                }
        }

        public static void main(String[] args) throws Exception{

          Configuration conf = new Configuration();

          Job job = Job.getInstance(conf, "Tv Sales Invalid REcords");
          job.setJarByClass(TvSales.class);

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

          job.setOutputKeyClass(Text.class);
          job.setOutputValueClass(IntWritable.class);

          job.setMapperClass(TvSalesMapper.class);
          job.setCombinerClass(TvSalesReducer.class);
          job.setReducerClass(TvSalesReducer.class);

          FileInputFormat.addInputPath(job, new Path(args[0]));
          FileOutputFormat.setOutputPath(job, new Path(args[1]));

          System.exit(job.waitForCompletion(true) ? 0 : 1);
        }
}
```

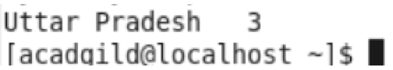
Export to a jar file in eclipse and type at command prompt

```
hadoop jar /home/acadgild/workspace/TvSales.jar television.txt
/total-units-onida
```

To check the results type

```
hadoop fs –cat /total-units-onida/part-r-00000
```

Output



```
Uttar Pradesh   3
[acadgild@localhost ~]$ █
```