## TASK-1→

## Below is the mapper code-

```
package task_1;

import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;


public class FilterMapper extends Mapper<LongWritable, Text,NullWritable,Text>{

	NullWritable key;
	Text line;

	public void setup(Context context){
		line = new Text();
	}

	public void map(LongWritable key1, Text value, Context context)
				throws IOException, InterruptedException{
		String lineArray = value.toString();
		line = new Text(lineArray);

		if(!lineArray.contains("NA")){
			context.write(key,line);
		}
	}
}
```

## Below is the Driver Code-

```
package task_1;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.io.Text;

public class FilterInvalidMain {

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

		Configuration conf = new Configuration();
		Job job = new Job(conf, "Map Reduce Assignment 4");
		job.setJarByClass(FilterInvalidMain.class);

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

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

		job.setMapperClass(FilterMapper.class);
		job.setNumReduceTasks(0);

		job.setInputFormatClass(TextInputFormat.class);
		job.setOutputFormatClass(TextOutputFormat.class);

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

		job.waitForCompletion(true);
	}
}
```

Once the jar is ready we run the jar using below command and specify two parameters-

a. /televison.txt→ i/p file present at HDFS
b. /A4task-1→ Output directory where output will be stored.

```
[acadgild@localhost Assignment-4]$ hadoop jar filterinvalidrec.jar /television.txt /A4task-1
18/07/09 11:49:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/07/09 11:50:16 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/07/09 11:50:51 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application wit
h ToolRunner to remedy this.
18/07/09 11:51:04 INFO input.FileInputFormat: Total input paths to process : 1
18/07/09 11:51:09 INFO mapreduce.JobSubmitter: number of splits:1
18/07/09 11:51:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1531116485784_0001
```

```
[acadgild@localhost Assignment-4]$ hadoop fs -cat /invalidcount/*
18/07/09 12:58:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
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
Lava|Attention|20|Assam|454601|24200
Zen|Super|14|Maharashtra|619082|9200
Samsung|Optima|14|Madhya Pradesh|132401|14200
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
[acadgild@localhost Assignment-4]$
```

Above diagram shows there are no records with text- “NA”.

**TASK-2-**

**Below is the mapper code-**

```
package task_2;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;

public class UnitCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

	IntWritable value;
	Text tvname;

	public void setup(Context context){
		value = new IntWritable(1);
		tvname = new Text();
	}

	public void map(LongWritable key1, Text company, Context context)
			throws IOException, InterruptedException{
		String[] lineArray = company.toString().split("\\|");

		if(!(lineArray[0].equals("NA") || (lineArray[1].equals("NA")))){
			tvname.set((lineArray[0]));
			context.write(tvname,value);
		}
	}
}
```

**Below is the reduer code-**

```
package task_2;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

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

	public void reduce(Text tvname, Iterable<IntWritable> count, Context context)
			throws IOException, InterruptedException{
		int sum = 0;
		for(IntWritable value: count){
			sum+= value.get();
		}
		context.write(tvname,new IntWritable(sum));
	}
}
```

**Below is the driver code-**

```
package task_2;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
```

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;

public class UnitCount {

	public static void main(String[] args) throws Exception {
		Configuration conf = new Configuration();
		Job job = new Job(conf, "Assignment 4");
		job.setJarByClass(UnitCount.class);

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

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

		job.setMapperClass(UnitCountMapper.class);
		job.setReducerClass(UnitCountReducer.class);

		job.setNumReduceTasks(2);
		job.setInputFormatClass(TextInputFormat.class);
		job.setOutputFormatClass(TextOutputFormat.class);

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

		job.waitForCompletion(true);
	}
}
```

**Unitcount.jar** is the jar file to be executed-

```
[acadgild@localhost Assignment-4]$ ls -l
total 12
-rw-rw-r--. 1 acadgild acadgild 2271 Jul  9 11:45 filterinvalidrec.jar
-rw-rw-r--. 1 acadgild acadgild  733 Jul  9 11:38 television.txt
-rw-rw-r--. 1 acadgild acadgild 1239 Jul  9 13:36 unitcount.jar
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Assignment-4]$
```

We execute the mapreduce job using below command-

Here **/television.txt** is the input file and **/unitcount** is the output directory

```
[acadgild@localhost Assignment-4]$ hadoop jar unitcount.jar /television.txt /unitcount
18/07/09 13:50:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/07/09 13:50:13 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/07/09 13:50:15 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application wit
h ToolRunner to remedy this.
18/07/09 13:50:16 INFO input.FileInputFormat: Total input paths to process : 1
18/07/09 13:50:16 INFO mapreduce.JobSubmitter: number of splits:1
18/07/09 13:50:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1531120796281_0002
18/07/09 13:50:17 INFO impl.YarnClientImpl: Submitted application application_1531120796281_0002
18/07/09 13:50:17 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1531120796281_0002/
18/07/09 13:50:17 INFO mapreduce.Job: Running job: job_1531120796281_0002
```

Below screenshot shows the output -

```
[acadgild@localhost Assignment-4]$ hadoop fs -cat /unitcount/*
18/07/09 13:52:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library
Onida   3
Zen     2
Akai    1
Lava    3
Samsung 7
[acadgild@localhost Assignment-4]$
```

## TASK-3-

Below is the mapper code-

```
package task_3;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;

public class OnidaCountMapper extends Mapper<LongWritable, Text, Text, IntWritable>{


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

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

## Below is the reducer code-

```
package task_3;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

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

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

            int sum = 0;
            for(IntWritable value:values){
                  sum+= value.get();
            }
            context.write(state,new IntWritable(sum));

      }
}
```

## Below is the driver code-

```
package task_3;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;


public class OnidaStateCount {


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

            Configuration conf = new Configuration();

            Job job = new Job(conf, "Assignment 4");

            job.setJarByClass(OnidaStateCount.class);

            job.setMapOutputKeyClass(Text.class);
```

```
            job.setMapOutputValueClass(IntWritable.class);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(IntWritable.class);
            job.setMapperClass(OnidaCountMapper.class);
            job.setReducerClass(OnidaCountReducer.class);
            job.setNumReduceTasks(1);
            job.setInputFormatClass(TextInputFormat.class);
            job.setOutputFormatClass(TextOutputFormat.class);
            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job,new Path(args[1]));
            job.waitForCompletion(true);
      }
}
```

Below diagram shows we are executing the onidastatecount jar

```
[acadgild@localhost Assignment-4]$ hadoop jar onidastatecount.jar /television.txt /onidastatecount
18/07/09 14:28:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/07/09 14:28:45 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/07/09 14:28:46 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your a
h ToolRunner to remedy this.
18/07/09 14:28:47 INFO input.FileInputFormat: Total input paths to process : 1
18/07/09 14:28:47 INFO mapreduce.JobSubmitter: number of splits:1
18/07/09 14:28:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1531120796281_0003
18/07/09 14:28:48 INFO impl.YarnClientImpl: Submitted application application_1531120796281_0003
18/07/09 14:28:48 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1531120796281_0003/
18/07/09 14:28:48 INFO mapreduce.Job: Running job: job_1531120796281_0003
```

Below diagram shows the output generated at /onidastatecount location

```
[acadgild@localhost Assignment-4]$ hadoop fs -cat /onidastatecount/*
18/07/09 14:38:16 WARN util.NativeCodeLoader: Unable to load native-hadoo
Kerala  1
Uttar Pradesh   3
[acadgild@localhost Assignment-4]$
```