Name : Ravi Kumar

USN : 1BM17CS150

Sign :

## Map Reduce → Find Max. of N numbers.

```
import  java.io.IOException;
import  java.util.StringTokenizer;
import  org.qpache.hadoop.io.IntWritable;
import  org.qpache.hadoop.io.longWritable;
import  org.apache.hadoop.io.Text;
import  org.apache.hadoop.mapreduce.Mapper;
import  org.apache.hadoop.mapreduce.Reducer;
import  org.apache.hadoop.conf.Configuration;
import  org.qpache.hadoop.mapreduce.Job;
import  org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import  org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import  org.apache.hadoop.mapreduce.lib.input.FileInput Format;
import  org.apache.hadoop.mapreduce.lib.output.fileOutputFormat;
import  org.apache.hadoop.fs.Path;


public class MaxNum {

    public static class MyMapper extends
            Mapper<longWritable, Text, Text, IntWritable >{
        protected void map(longWritable key, Text value, Context
                context) throws IOException, InterruptedException
        {
```

Name : Ravi Kumar
USN : 1BM17CS150
Sign :

```
String[] numbers = value.toString().split(" ");
for(String num : numbers)
{
    content.write(new Text("Max"), new IntWritable
                              (Integer.parseInt(num));
}

}}

public static class MyReducer extends Reducer<Text,
        IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterable<IntWritable>
            values, Context context) throws Exception
        {
            Integer maxnum = -1;
          for (IntWritable val : values)
            {
              if (val.get() > maxnum)
                {
                    maxnum = val.get();
                }
            }
          context.write (key, new IntWritable(maxnum));
        }
    }
```

Name : Ravi Kumar

USN : 1BM17CS150

Sign :

```
public static void main (String[] args) throws IOException,
            InterruptedException, ClassNotFoundException {

    if (args.length != 2)
    {
      System.err.println(" Please Enter the input & output params");
      System.exit(-1);
    }

    Job job = new Job();

     job.setJarByClass("MaxNum");
     job.setJobName("MaxNumber");

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

     job.setMapperClass(MyMapper class);
     job.setReducerClass(MyReducer.class);

     job.setOutputKeyClass(Text.class);
     job.setOutputValueClass(IntWritable.class);
     System.exit(job.waitForCompletion(true)?0:1);

    }

}
```