BDA
Endterm

Madana Gopal.D
2018ICSE0385
7CSE6

MainCode

```
public class wordmain385 extends Configured implements Tool {

 @Override
  public int run (String[] args) throws Exception {
        if(args.length <2) {
     System.out.println("Please give correct input output directory");
     return -1;
   }
    JobConf conf = new JobConf(wordmain385.class)
    FileInputFormat.setInputPaths(Conf, new Path(args[0]));
    FileOutputFormat.setOutputPaths(Conf, new Path(args[1]));
   Conf.setMapperClass(wordmapper385.Class
   Conf.setReducerClass(wordreducer385.class);
   Conf.setMapOutputKeyclass(Text.class);
   Conf.setMapOutputValueclass(Text.class);
   Conf.setOutputKeyClass(Text.Class);
   Conf.setOutputValueClass(Intwritable.class);
   JobClient.runJob(conf);
   return 0;
 }
  public static void main(string args[]) throws Exception{
     int exitcode = ToolRunner.run(new wordmain385(), args);
     System.exit(exitcode);
   }
}
```

Madana Gopal.D
20181CSE0385
7CSE6

Mapper Code

```
public class wordmapper385  extends MapReduceBase implements
  Mapper<LongWritable, ~~Key~~, Text ~~value~~, Text,
 {
 public void map(LongWritable Key, Text value, OutputCollector
     <Text, IntWritable> output, Reporter r) throws IOException

 {
   String s = value.toString().
   for(String word : s.split(",")))
   {
    if (word.length()>0 && (word.equals("heaven") && (word.
          equals("earth"))))
     {
      ~~o~~ output.collect(new Text(word), new IntWritable(1)).
     }
   }
 }
}
```

Maclana Gopal.D
20181CSE0385
7CSE6

Reducer :-

```
public class wordreducer385  extends MapReduceBase implements
  Reducer<Text, IntWritable, Text, IntWritable>
{
 @Override
 public void reduce(Text Key, Iterator<IntWritable> arg1,
     OutputCollector<Text, IntWritable> arg2, Reporter arg3)
     throws IOException
 {
    int Count=0;
    while(arg1.hasNext())
     {
       IntWritable i= arg1.next().
        Count += i.get().
     }
    arg2.collect(arg0, new IntWritable(count)).
   }
}
```