Q) **Develop a MapReduce program to calculate the frequency of a given word in a given file.**

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

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

```java
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}

public static class IntSumReducer
     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, "word count");
```

```
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Execution Steps

1. **mkdir usn_prog1**
2. **cd usn_prog1**
3. **gedit WordCount.java**
4. **start-all.sh**
5. **jps**
6. **export HADOOP_CLASSPATH=$(hadoop classpath)**
7. **mkdir Input**
8. **cd Input**

9. **gedit input_data.txt**

   //Write some text

10. **cd . .**

11. **hadoop fs -mkdir /wordcount_usn**

12. **hadoop fs -mkdir /wordcount_usn/Input**

13. **hadoop fs -put ./Input/inputdata_data.txt/ /wordcount_usn/Input**

14. **export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64**

15. **export PATH=$JAVA_HOME/bin:$PATH**

16. **javac -classpath $(hadoop classpath) -d . WordCount.java**

17. **jar -cvf wordcount.jar -C . .**

18. **hadoop jar wordcount.jar WordCount /wordcount_usn/Input /wordcount_usn/Input/output**

19. **hadoop fs -cat /wordcount_usn/Input/output/part-r-00000**