LAB – 7

For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words.

Driver-TopN.class

```
package samples.topn;

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.lib.input.FileInputFormat;

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

import org.apache.hadoop.util.GenericOptionsParser;


public class TopN {

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

    Configuration conf = new Configuration();

    String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();

    if (otherArgs.length != 2) {

      System.err.println("Usage: TopN <in> <out>");

      System.exit(2);

    }

    Job job = Job.getInstance(conf);

    job.setJobName("Top N");

    job.setJarByClass(TopN.class);
    job.setMapperClass(TopNMapper.class);
    job.setReducerClass(TopNReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }

  public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
    private static final IntWritable one = new IntWritable(1);
    private Text word = new Text();
    private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";
    public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
      String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
      StringTokenizer itr = new StringTokenizer(cleanLine);
      while (itr.hasMoreTokens()) {
        this.word.set(itr.nextToken().trim());
        context.write(this.word, one);
      }
    }
  }
}
```

TopNCombiner.class

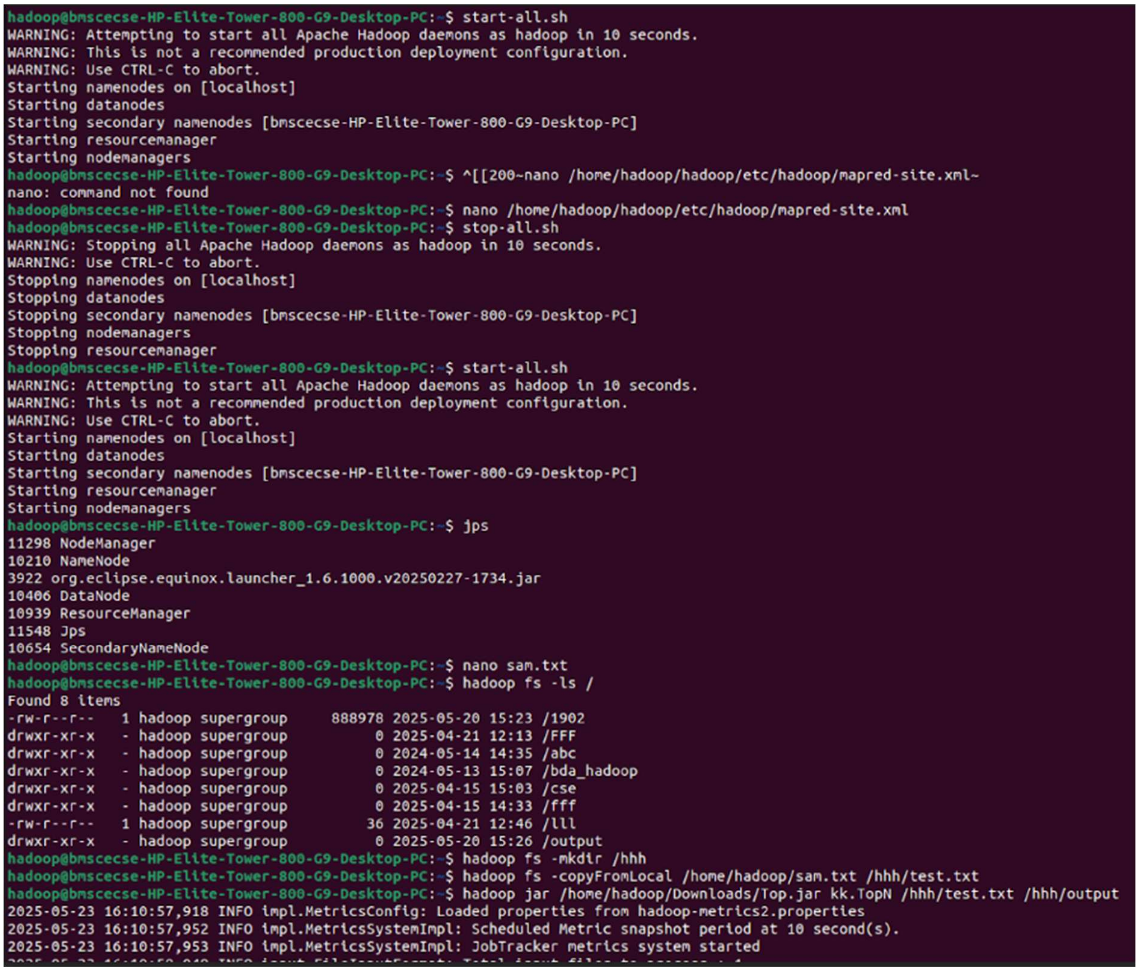
```
package samples.topn;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

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

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

    int sum = 0;

    for (IntWritable val : values)

      sum += val.get();

    context.write(key, new IntWritable(sum));

  }

}
```

TopNMapper.class

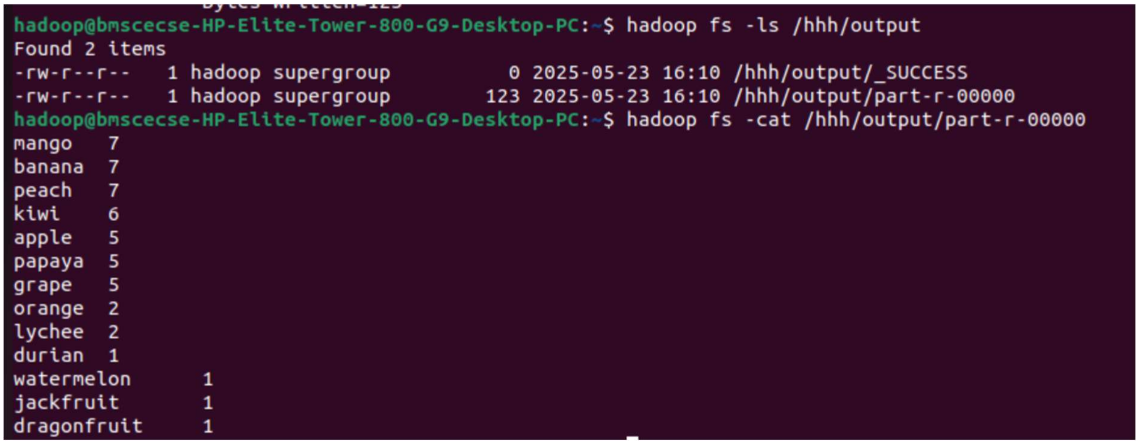
```
package samples.topn;

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {

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

  private Text word = new Text();

  private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";

  public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {

    String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");

    StringTokenizer itr = new StringTokenizer(cleanLine);

    while (itr.hasMoreTokens()) {

      this.word.set(itr.nextToken().trim());
      context.write(this.word, one);
    }
  }
}
```

TopNReducer.class

```
package samples.topn;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;
public class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
  private Map<Text, IntWritable> countMap = new HashMap<>();
  public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values)
      sum += val.get();
    this.countMap.put(new Text(key), new IntWritable(sum));
  }
  protected void cleanup(Reducer<Text, IntWritable, Text, IntWritable>.Context context)
throws IOException, InterruptedException {
    Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(this.countMap);
    int counter = 0;
    for (Text key : sortedMap.keySet()) {
      if (counter++ == 20)
        break;
      context.write(key, sortedMap.get(key));
    }
  }
}
```



```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ^[[200~nano /home/hadoop/hadoop/etc/hadoop/mapred-site.xml~
nano: command not found
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ nano /home/hadoop/hadoop/etc/hadoop/mapred-site.xml
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Stopping nodemanagers
Stopping resourcemanager
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ jps
11298 NodeManager
10210 NameNode
3922 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
10406 DataNode
10939 ResourceManager
11548 Jps
10654 SecondaryNameNode
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ nano sam.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
Found 8 items
-rw-r--r--   1 hadoop supergroup     888978 2025-05-20 15:23 /1902
drwxr-xr-x   - hadoop supergroup          0 2025-04-21 12:13 /FFF
drwxr-xr-x   - hadoop supergroup          0 2024-05-14 14:35 /abc
drwxr-xr-x   - hadoop supergroup          0 2024-05-13 15:07 /bda_hadoop
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 15:03 /cse
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 14:33 /fff
-rw-r--r--   1 hadoop supergroup         36 2025-04-21 12:46 /lll
drwxr-xr-x   - hadoop supergroup          0 2025-05-20 15:26 /output
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /hhh
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/sam.txt /hhh/test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Downloads/Top.jar kk.TopN /hhh/test.txt /hhh/output
2025-05-23 16:10:57,918 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-23 16:10:57,952 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-23 16:10:57,953 INFO impl.MetricsSystemImpl: JobTracker metrics system started
```



```
                Bytes Written=123
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /hhh/output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-23 16:10 /hhh/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup        123 2025-05-23 16:10 /hhh/output/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /hhh/output/part-r-00000
mango   7
banana  7
peach   7
kiwi    6
apple   5
papaya  5
grape   5
orange  2
lychee  2
durian  1
watermelon      1
jackfruit       1
dragonfruit     1
```