Practical - 5

2CS702 – Big Data Analytics

Harshit Gajipara
19BCE059

Aim:

Apply MapReduce algorithms to find phrase frequency from given dataset.

Steps:

- Create a new project using Ant in apache netbeans
- Add libraries from below path to library and set jdk as 1.8.0

“D:/BDA/hadoop-3.2.1/share/hadoop/common/hadoop-common-3.2.1.jar”

“D:/BDA/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.2.1.jar”

- After saving java files, clean and build the project and run hadoop command after uploading num.txt file.

Hadoop jar “C:\Users\HARSHIT\Documents\NetBeansProjects\Practical5\dist\Practical5.jar” /prac5/wordfile.txt /prac5/output

- Now run this command to view the output

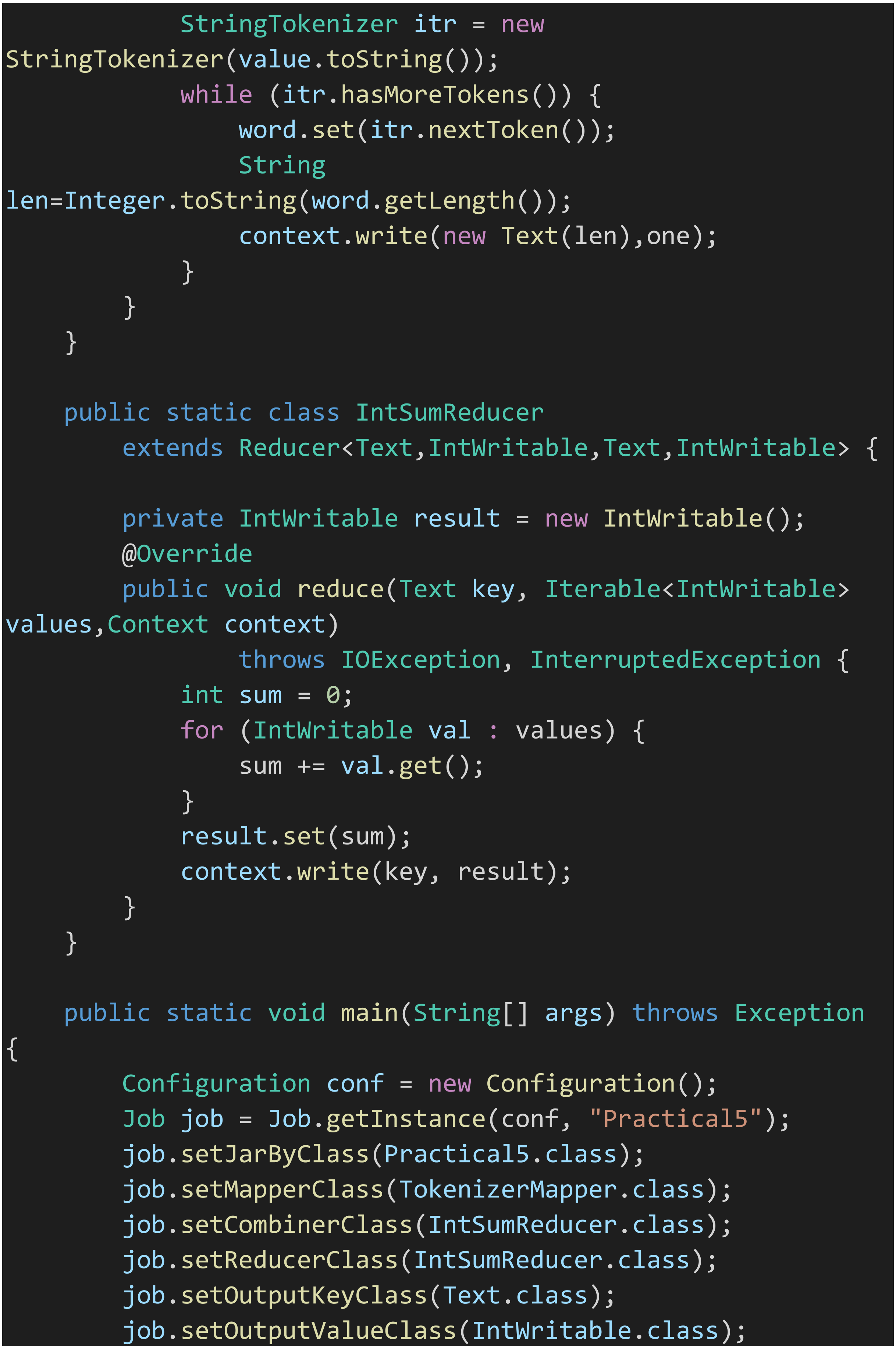
hdfs dfs –cat /prac5/output/part-r-00000

Code:

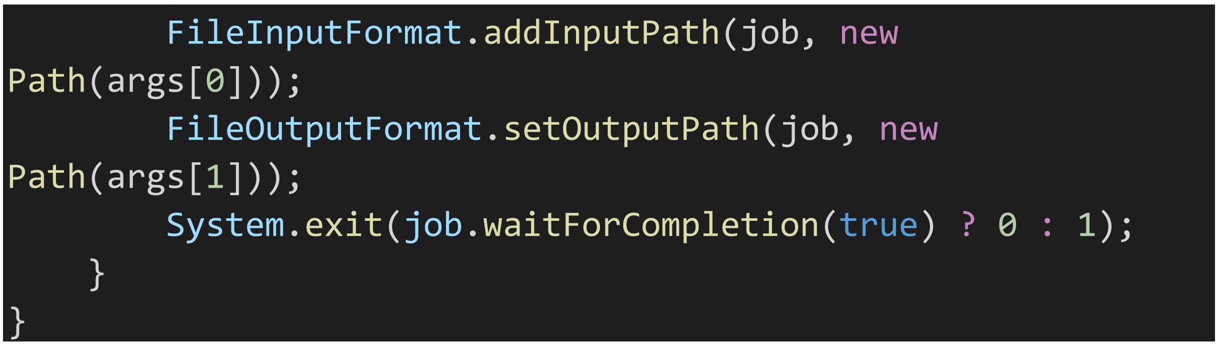
`Practical5.java`

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
```

```
package practical5;

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.LongWritable;
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;

/**
 *
 * @author HARSHIT
 */

public class Practical5 {

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

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

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



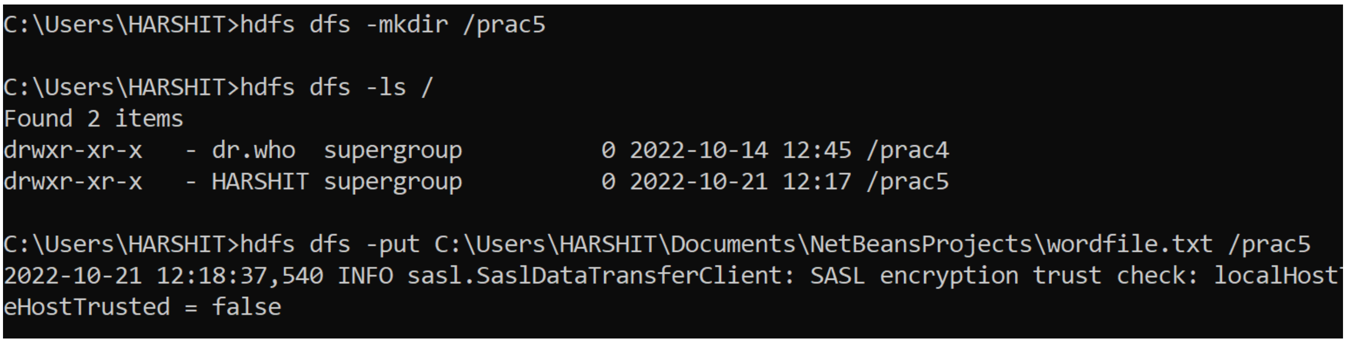
```
            StringTokenizer itr = new
StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                String
len=Integer.toString(word.getLength());
                context.write(new Text(len),one);
            }
        }
    }

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

        private IntWritable result = new IntWritable();
        @Override
        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, "Practical5");
        job.setJarByClass(Practical5.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);
    }
}
```

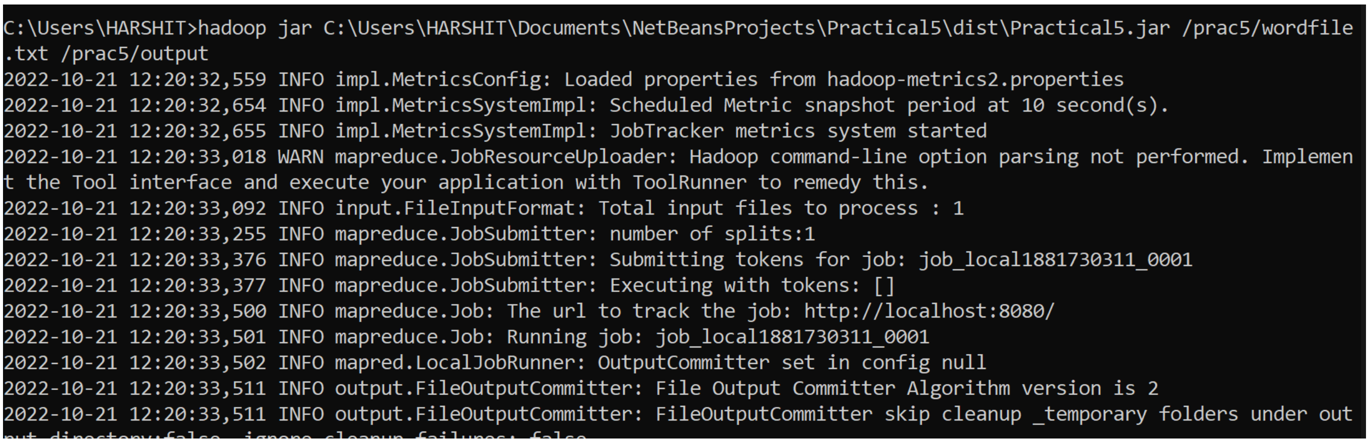
Output:

Uploading wordfile.txt on hdfs in prac5 folder

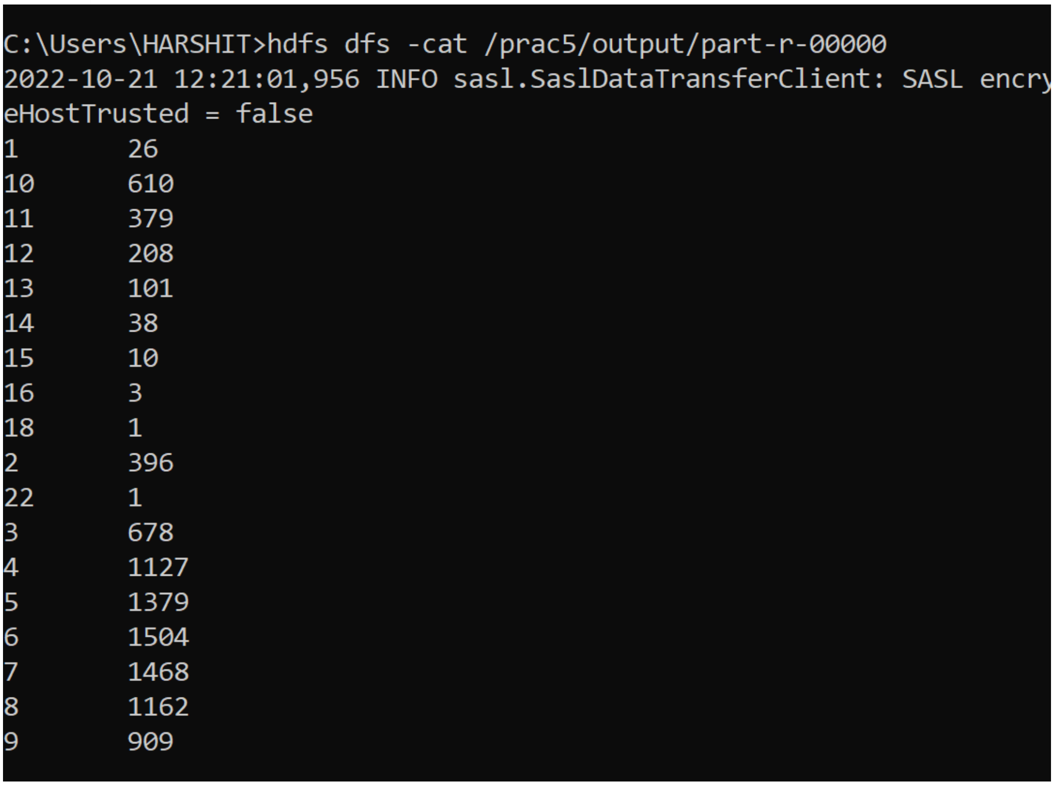


```
C:\Users\HARSHIT>hdfs dfs -mkdir /prac5

C:\Users\HARSHIT>hdfs dfs -ls /
Found 2 items
drwxr-xr-x   - dr.who  supergroup          0 2022-10-14 12:45 /prac4
drwxr-xr-x   - HARSHIT supergroup          0 2022-10-21 12:17 /prac5

C:\Users\HARSHIT>hdfs dfs -put C:\Users\HARSHIT\Documents\NetBeansProjects\wordfile.txt /prac5
2022-10-21 12:18:37,540 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHost
eHostTrusted = false
```

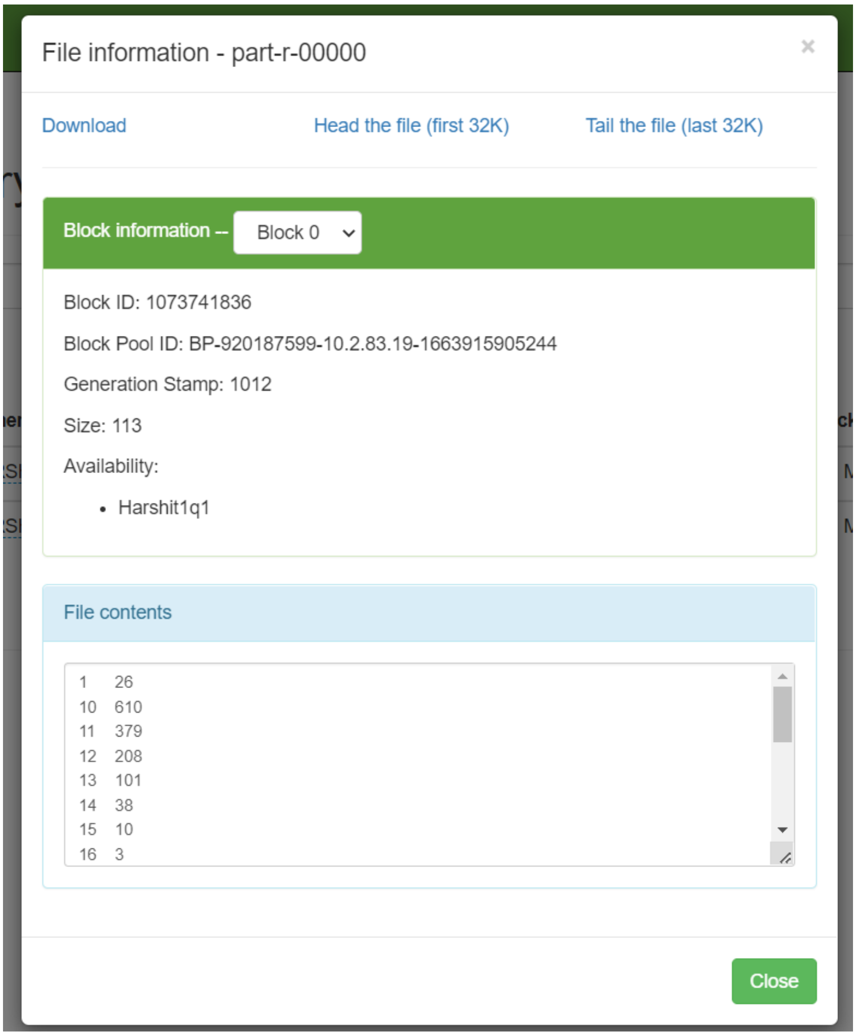
Running jar command using input file and generating output file in hadoop



```
C:\Users\HARSHIT>hadoop jar C:\Users\HARSHIT\Documents\NetBeansProjects\Practical5\dist\Practical5.jar /prac5/wordfile
.txt /prac5/output
2022-10-21 12:20:32,559 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2022-10-21 12:20:32,654 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2022-10-21 12:20:32,655 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2022-10-21 12:20:33,018 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implemen
t the Tool interface and execute your application with ToolRunner to remedy this.
2022-10-21 12:20:33,092 INFO input.FileInputFormat: Total input files to process : 1
2022-10-21 12:20:33,255 INFO mapreduce.JobSubmitter: number of splits:1
2022-10-21 12:20:33,376 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1881730311_0001
2022-10-21 12:20:33,377 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-10-21 12:20:33,500 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2022-10-21 12:20:33,501 INFO mapreduce.Job: Running job: job_local1881730311_0001
2022-10-21 12:20:33,502 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2022-10-21 12:20:33,511 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-10-21 12:20:33,511 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under out
put directory:false, ignore cleanup failures: false
```



```
C:\Users\HARSHIT>hdfs dfs -cat /prac5/output/part-r-00000
2022-10-21 12:21:01,956 INFO sasl.SaslDataTransferClient: SASL encry
eHostTrusted = false
1       26
10      610
11      379
12      208
13      101
14      38
15      10
16      3
18      1
2       396
22      1
3       678
4       1127
5       1379
6       1504
7       1468
8       1162
9       909
```



File information - part-r-00000

Block ID: 1073741836

Block Pool ID: BP-920187599-10.2.83.19-1663915905244

Generation Stamp: 1012

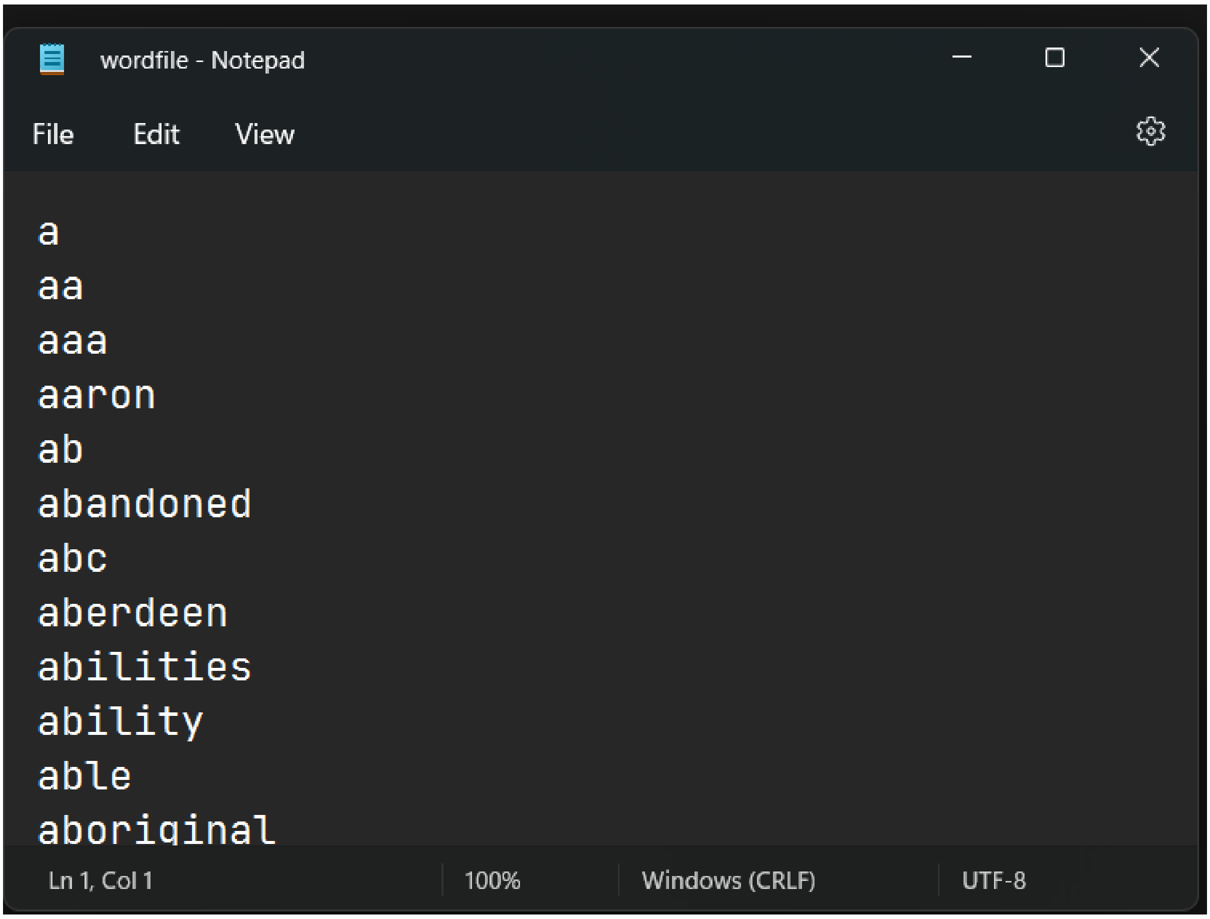
Size: 113

Availability:

- Harshit1q1

File contents

```
1   26
10  610
11  379
12  208
13  101
14  38
15  10
16  3
```



```
a
aa
aaa
aaron
ab
abandoned
abc
aberdeen
abilities
ability
able
aboriginal
```

Conclusion:

In this practical, we learnt that using mapper and reducer class in hadoop distributed file system, we can process word file data in parallel and faster.