

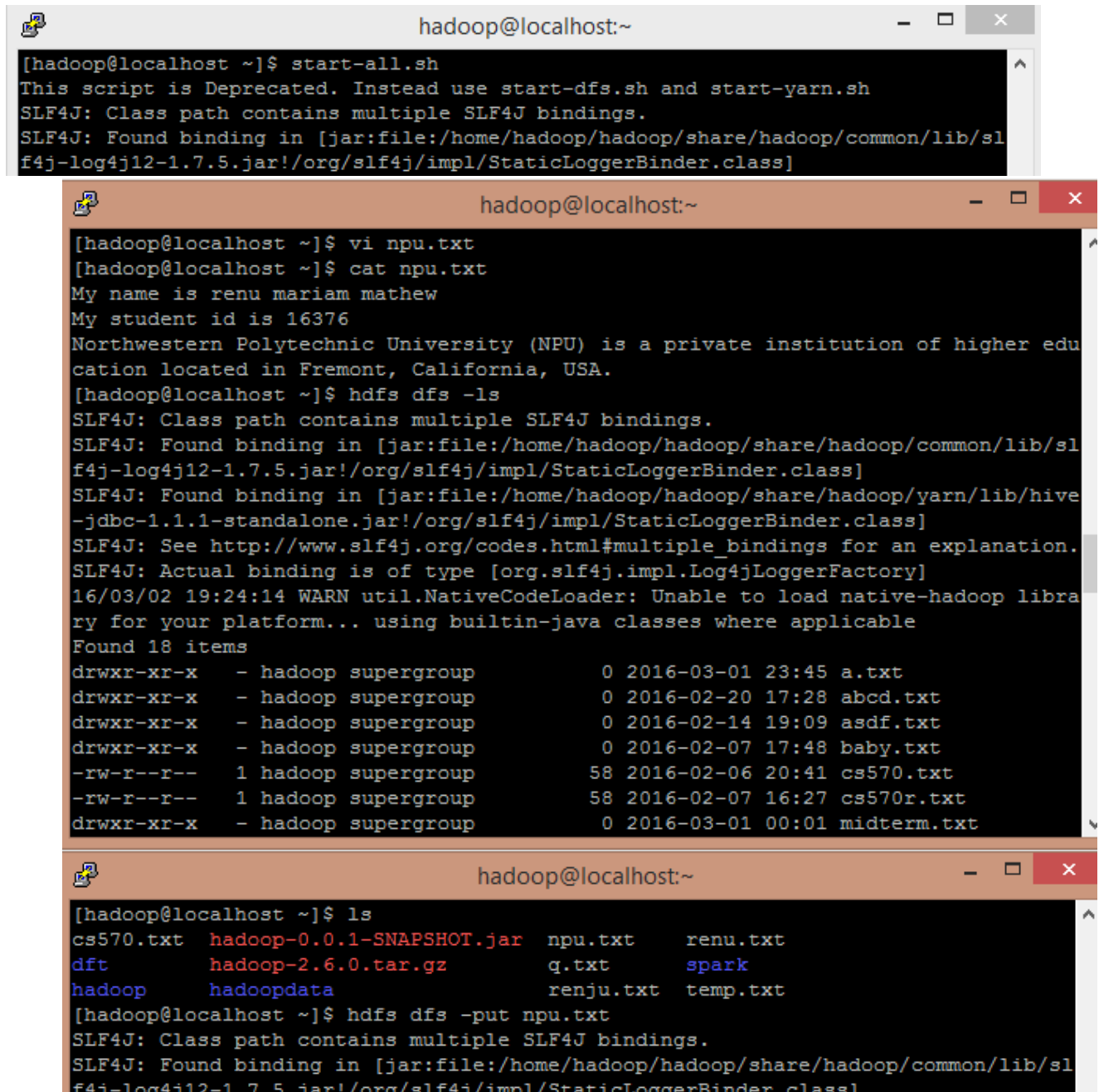
Name:Renu Mariam Mathew

Student Id:16376

Hw#2

Note: The first part of the document contains all the snapshots and the code for the program following that.

Initial steps:



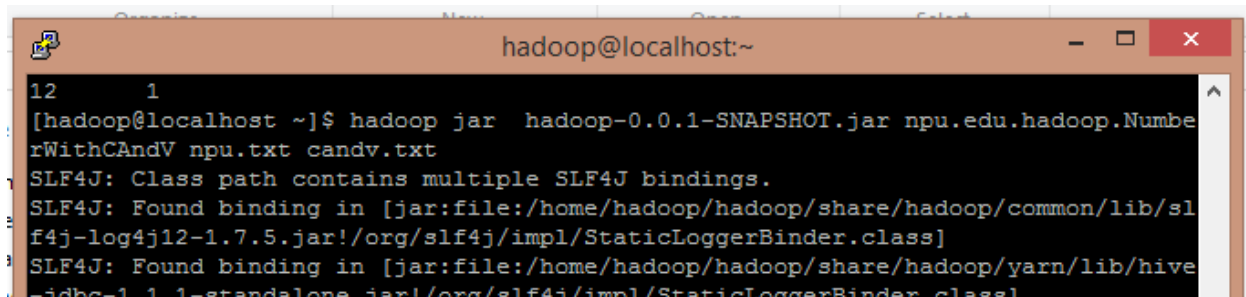
```
hadoop@localhost:~  
[hadoop@localhost ~]$ start-all.sh  
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
  
hadoop@localhost:~  
[hadoop@localhost ~]$ vi npu.txt  
[hadoop@localhost ~]$ cat npu.txt  
My name is renu mariam mathew  
My student id is 16376  
Northwestern Polytechnic University (NPU) is a private institution of higher education located in Fremont, California, USA.  
[hadoop@localhost ~]$ hdfs dfs -ls  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]  
16/03/02 19:24:14 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Found 18 items  
drwxr-xr-x - hadoop supergroup 0 2016-03-01 23:45 a.txt  
drwxr-xr-x - hadoop supergroup 0 2016-02-20 17:28 abcd.txt  
drwxr-xr-x - hadoop supergroup 0 2016-02-14 19:09 asdf.txt  
drwxr-xr-x - hadoop supergroup 0 2016-02-07 17:48 baby.txt  
-rw-r--r-- 1 hadoop supergroup 58 2016-02-06 20:41 cs570.txt  
-rw-r--r-- 1 hadoop supergroup 58 2016-02-07 16:27 cs570r.txt  
drwxr-xr-x - hadoop supergroup 0 2016-03-01 00:01 midterm.txt  
  
hadoop@localhost:~  
[hadoop@localhost ~]$ ls  
cs570.txt  hadoop-0.0.1-SNAPSHOT.jar  npu.txt  renu.txt  
dft        hadoop-2.6.0.tar.gz          q.txt    spark  
hadoop     hadoopdata                   renju.txt temp.txt  
[hadoop@localhost ~]$ hdfs dfs -put npu.txt  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
```

Name:Renu Mariam Mathew

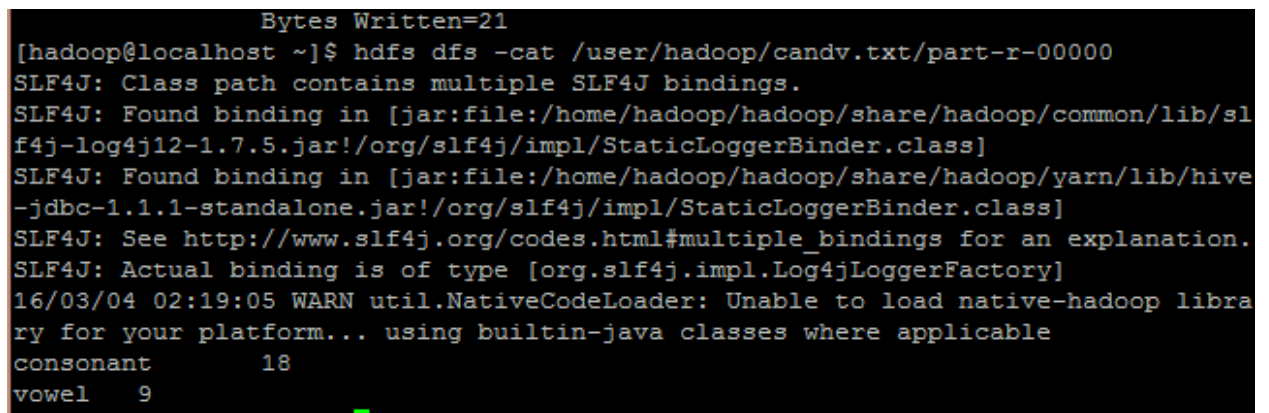
Student Id:16376

Hw#2

1. Write a map reduce program to find the number of words that start with a consonant and vowel

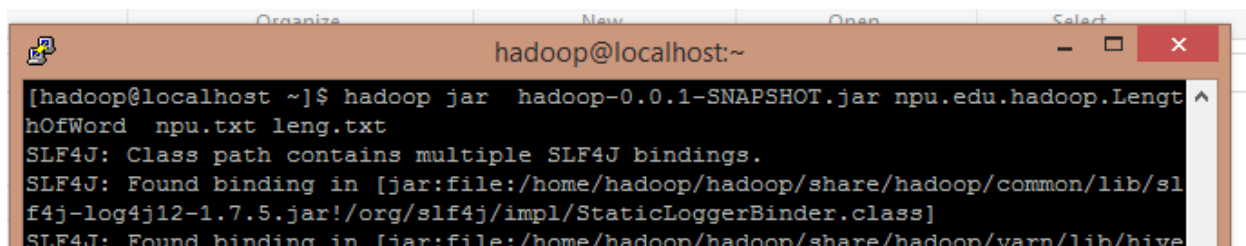
A terminal window titled 'hadoop@localhost:~' showing the execution of a Hadoop jar command. The command is 'hadoop jar hadoop-0.0.1-SNAPSHOT.jar npu.edu.hadoop.NumberWithCAndV npu.txt candv.txt'. The output shows SLF4J warnings about multiple bindings and the start of a file listing. The first line of the listing is '12 1'.

```
hadoop@localhost:~  
12      1  
[hadoop@localhost ~]$ hadoop jar hadoop-0.0.1-SNAPSHOT.jar npu.edu.hadoop.NumberWithCAndV npu.txt candv.txt  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
```

A terminal window showing the execution of an HDFS command. The command is 'hdfs dfs -cat /user/hadoop/candv.txt/part-r-00000'. The output shows SLF4J warnings and a file listing. The listing shows 'consonant 18' and 'vowel 9'.

```
Bytes Written=21  
[hadoop@localhost ~]$ hdfs dfs -cat /user/hadoop/candv.txt/part-r-00000  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]  
16/03/04 02:19:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
consonant      18  
vowel      9
```

2. Write a map reduce program to find the length of each words in a document

A terminal window titled 'hadoop@localhost:~' showing the execution of a Hadoop jar command. The command is 'hadoop jar hadoop-0.0.1-SNAPSHOT.jar npu.edu.hadoop.LengthOfWord npu.txt leng.txt'. The output shows SLF4J warnings about multiple bindings.

```
hadoop@localhost:~  
[hadoop@localhost ~]$ hadoop jar hadoop-0.0.1-SNAPSHOT.jar npu.edu.hadoop.LengthOfWord npu.txt leng.txt  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
```

Name:Renu Mariam Mathew

Student Id:16376

Hw#2

```
hadoop@localhost:~  
File Output Format Counters  
    Bytes Written=47  
[hadoop@localhost ~]$ hdfs dfs -cat /user/hadoop/leng.txt/part-r-00000  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]  
16/03/04 01:57:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
1      1  
2      8  
4      3  
5      2  
6      3  
7      3  
8      1  
9      1  
10     1  
11     3  
12     1
```

3. Write a map reduce program to reverse all the words in a document.

```
hadoop@localhost:~  
vowel 9  
[hadoop@localhost ~]$ hadoop jar hadoop-0.0.1-SNAPSHOT.jar npu.edu.hadoop.ReverseOfAllWord npu.txt reverseWord.txt  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]  
16/03/04 02:22:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
16/03/04 02:22:12 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
```

Name:Renu Mariam Mathew
Student Id:16376
Hw#2

```
hadoop@localhost:~$ hdfs dfs -cat /user/hadoop/re.txt/part-r-00000
Bytes Written=177
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-jdbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
16/03/02 19:51:28 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
)UPN(
,ainrofilaC
,tnomerF
.ASU
67361
a
cinhcetyloP
detacol
di
eman
etavirp
fo
mairam
ni
```

Name:Renu Mariam Mathew
Student Id:16376
Hw#2

```
hadoop@localhost:~  
a  
cinhcetyloP  
detacol  
di  
eman  
etavirp  
fo  
mairam  
ni  
noitacude  
noitutitsni  
nretsewhtroN  
rehgiH  
si  
si  
si  
tneduts  
uner  
wehtam  
yM  
yM  
ytisrevinU  
[hadoop@localhost ~]$
```

4. Write a map reduce program to count the total number of words in a document.

```
hadoop@localhost:~  
drwxr-xr-x - hadoop supergroup 0 2016-02-06 00:28 test  
[hadoop@localhost ~]$ ls  
cs570.txt hadoop-0.0.1-SNAPSHOT.jar npu.txt renu.txt  
dft hadoop-2.6.0.tar.gz q.txt spark  
hadoop hadoopdata renju.txt temp.txt  
[hadoop@localhost ~]$ hadoop jar hadoop-0.0.1-SNAPSHOT.jar npu.edu.hadoop.WordCount npu.txt wc.txt  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/yarn/lib/hive-idbc-1.1.1-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
```

Name:Renu Mariam Mathew

Student Id:16376

Hw#2

```
hadoop@localhost:~  
Bytes Read=178  
File Output Format Counters  
Bytes Written=216  
[hadoop@localhost ~]$ hdfs dfs -cat /user/hadoop/wc/part-r-00000  
SLF4J: Class path contains multiple SLF4J bindings.
```

```
hadoop@localhost:~  
(NPU) 1  
16376 1  
California, 1  
Fremont, 1  
My 2  
Northwestern 1  
Polytechnic 1  
USA. 1  
University 1  
a 1  
education 1  
higher 1  
id 1  
in 1  
institution 1  
is 3  
located 1  
mariam 1  
mathew 1  
name 1  
of 1  
private 1  
renu 1  
student 1  
[hadoop@localhost ~]$
```

Name:Renu Mariam Mathew
Student Id:16376
Hw#2

1. Write a map reduce program to find the number of words that start with a consonant and vowel.

```
package npu.edu.hadoop;

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.Mapper.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class NumberWithCAndV {
    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 {

            String a = "vowel";
            String b = "consonant";

            StringTokenizer itr = new
StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {

                String w = itr.nextToken();

                if (w != null) {

                    if (w.toLowerCase().startsWith("a") ||
w.toLowerCase().startsWith("e")
```

Name:Renu Mariam Mathew

Student Id:16376

Hw#2

```

|| w.toLowerCase().startsWith("o")
|| w.toLowerCase().startsWith("i")
|| w.startsWith("u")) {

    word.set(a);

} else {
    word.set(b);
}

}
context.write(word, one);
}

}

public static class VowelsCountReducer 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 value : values) {
            sum += value.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}

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:
NumberOfWordsWithConosantAndVowel <in> <out>");
        System.exit(2);
    }

    Job job = new Job(conf, "NumberOfWordsWithConosantAndVowel");

    job.setJarByClass(NumberWithCAndV.class);

    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(VowelsCountReducer.class);
}
```


Name:Renu Mariam Mathew
Student Id:16376
Hw#2

```
        job.setReducerClass(VowelsCountReducer.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);
    }
}
```

2. Write a map reduce program to find the length of each words in a document

```
package npu.edu.hadoop;

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.Reducer.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
import npu.edu.hadoop.WordCount.IntSumReducer;
import npu.edu.hadoop.WordCount.TokenizerMapper;
```

Name:Renu Mariam Mathew

Student Id:16376

Hw#2

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

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

        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
            StringTokenizer itr = new
StringTokenizer(value.toString());

            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken().length());
                context.write(word, one);
            }
        }

        public static class IntSumReducer extends Reducer<IntWritable,
IntWritable, IntWritable, IntWritable> {
            private IntWritable result = new IntWritable();

            public void reduce(IntWritable 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();
            String[] otherArgs = new GenericOptionsParser(conf,
args).getRemainingArgs();
            if (otherArgs.length != 2) {
                System.err.println("Usage: LengthOfWord <in> <out>");
                System.exit(2);
            }
            Job job = new Job(conf, "LengthOfWord");
            job.setJarByClass(LengthOfWord.class);
            job.setMapperClass(TokenizerMapper.class);
            job.setCombinerClass(IntSumReducer.class);
            job.setReducerClass(IntSumReducer.class);
            job.setOutputKeyClass(IntWritable.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);
        }
    }
}
```

Name:Renu Mariam Mathew

Student Id:16376

Hw#2

```
    }  
}
```

3. Write a map reduce program to reverse all the words in a document.

```
package npu.edu.hadoop;  
  
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.NullWritable;  
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;  
  
public class ReverseOfAllWord {  
  
    public static class ReverseMapper extends Mapper<LongWritable, Text,  
Text, NullWritable> {  
        private Text word = new Text();  
        private final static IntWritable one = new IntWritable(1);  
  
        public void map(LongWritable key, Text value, Context context)  
throws IOException, InterruptedException {  
            String line = value.toString().concat("\n");  
            StringBuffer b = new StringBuffer(line);  
            String result = (b.reverse().toString());  
            StringTokenizer itr = new StringTokenizer(result);  
            while (itr.hasMoreTokens()) {  
  
                word.set(itr.nextToken());  
  
                context.write(word, NullWritable.get());  
            }  
        }  
    }  
  
    public static void main(String[] args) throws Exception {  
  
        Configuration conf = new Configuration();  
  
        Job job = new Job(conf, "ReverseOfAllWord");
```

Name:Renu Mariam Mathew

Student Id:16376

Hw#2

```
        job.setJarByClass (ReverseOfAllWord.class);

        job.setMapperClass (ReverseMapper.class);
        // job.setCombinerClass (.class);

        // job.setReducerClass (.class);

        job.setOutputKeyClass (Text.class);
        job.setOutputValueClass (NullWritable.class);

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

        job.waitForCompletion(true);
    }
}
```

4. Write a map reduce program to count the total number of words in a document.

```
package npu.edu.hadoop;

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;
import org.apache.hadoop.util.GenericOptionsParser;

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

Name:Renu Mariam Mathew

Student Id:16376

Hw#2

```
    }

    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();
        String[] otherArgs = new GenericOptionsParser(conf,
args).getRemainingArgs();
        if (otherArgs.length != 2) {
            System.err.println("Usage: WordCount <in> <out>");
            System.exit(2);
        }
        Job job = new Job(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(otherArgs[0]));
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```