

## Rami Ibrahim ING 2 TD 2

### Compte rendu TP3 BIG DATA

#### Exercise 1:

To identify anagrams, the mapper will take each word and sort the letters of the word in alphabetical order, where the key is the sorted word, and the value is the actual word. The reducer will then return only the list of actual words.

Let's create `Anagrammes.java`

```
GNU nano 7.2                                         rami_ibrahim@fedora:~/Documents/Asma/TP2 — nano Anagramme.java                                         rami_ibrahim@fedora:~/Documents/Asma/TP2 — sudo nano input.txt
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class Anagramme {

    public static class Mapper extends org.apache.hadoop.mapreduce.Mapper<Object, Text, Text, Text> {
        private Text sortedKey = new Text();
        private Text originalWord = new Text();

        @Override
        protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String word = value.toString().trim();
            char[] letters = word.toLowerCase().toCharArray();
            java.util.Arrays.sort(letters);
            String sortedWord = new String(letters);
            sortedKey.set(sortedWord);
            originalWord.set(word);
            context.write(sortedKey, originalWord);
        }
    }

    public static class Reducer extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, Text> {
        private Text result = new Text();

        @Override
        protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            List<String> listOfWords = new ArrayList<>();
            for (Text value : values) {
                listOfWords.add(value.toString());
            }
            if (listOfWords.size() > 1) {
                result.set(String.join(" ", listOfWords));
                context.write(key, result);
            }
        }
    }

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: anagramme <input path> <output path>");
            System.exit(-1);
        }
    }
}

^G Help      ^P Write Out   ^W Where Is   ^X Cut          ^T Execute   ^C Location   M-U Undo   M-A Set Mark   M-[ To Bracket   M-Q Previous   ^B Back      ^A Prev Word   ^H Home
^X Exit      ^O Read File   ^M Replace   ^Y Paste         ^S Justify    M-C Go To Line M-E Redo   M-G Copy     M-Q Where Was   M-W Next    M-F Forward  M-N Next Word  M-B End
```

```
GNU nano 7.2                                         rami_ibrahim@fedora:~/Documents/Asma/TP2 — nano Anagramme.java                                         rami_ibrahim@fedora:~/Documents/Asma/TP2 — sudo nano input.txt
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import java.io.IOException;
import java.util.List;
import java.util.StringJoiner;

public class Anagramme {

    public static class Mapper extends org.apache.hadoop.mapreduce.Mapper<Object, Text, Text, Text> {
        private Text sortedKey = new Text();
        private Text originalWord = new Text();

        @Override
        protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String word = value.toString().trim();
            char[] letters = word.toLowerCase().toCharArray();
            java.util.Arrays.sort(letters);
            String sortedWord = new String(letters);
            sortedKey.set(sortedWord);
            originalWord.set(word);
            context.write(sortedKey, originalWord);
        }
    }

    public static class Reducer extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, Text> {
        private Text result = new Text();

        @Override
        protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            List<String> listOfWords = new ArrayList<>();
            for (Text value : values) {
                listOfWords.add(value.toString());
            }
            if (listOfWords.size() > 1) {
                result.set(String.join(" ", listOfWords));
                context.write(key, result);
            }
        }
    }

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: anagramme <input path> <output path>");
            System.exit(-1);
        }
    }

    Configuration conf = new Configuration();
    Job job = new Job(conf, "anagramme");

    job.setJarByClass(Anagramme.class);
    job.setMapperClass(Mapper.class);
    job.setReducerClass(Reducer.class);

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

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

    System.exit(job.waitForCompletion(true) ? 0 : 1);
}

^G Help      ^P Write Out   ^W Where Is   ^X Cut          ^T Execute   ^C Location   M-U Undo   M-A Set Mark   M-[ To Bracket   M-Q Previous   ^B Back      ^A Prev Word   ^H Home
^X Exit      ^O Read File   ^M Replace   ^Y Paste         ^S Justify    M-C Go To Line M-E Redo   M-G Copy     M-Q Where Was   M-W Next    M-F Forward  M-N Next Word  M-B End
```

Move it into the folder

```
C:\Windows\System32\docker-hadoop>docker cp "C:/Users/hp/Desktop/Anagramme.java" namenode:/Classes/exercice1/Anagramme.java
Successfully copied 4.61kB to namenode:/Classes/exercice1/Anagramme.java

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# cd Classes/exercice1
root@e3564f682d93:/Classes/exercice1# ls
Anagramme.java
root@e3564f682d93:/Classes/exercice1# javac -classpath "$(hadoop classpath)" -d . Anagramme.java
root@e3564f682d93:/Classes/exercice1# ls org
Anagramme.java  org
root@e3564f682d93:/Classes/exercice1# ls /org
ls: cannot access '/org': No such file or directory
root@e3564f682d93:/Classes/exercice1# ls org
example
root@e3564f682d93:/Classes/exercice1# ls org/example
Anagramme$Mapper.class  Anagramme$Reducer.class  Anagramme.class
root@e3564f682d93:/Classes/exercice1#
```

Create the jar

```
root@e3564f682d93:/Classes/exercice1# jar -cvf Anagramme.jar -C .
added manifest
adding: Anagramme.java(in = 2709) (out= 894)(deflated 66%)
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/example/(in = 0) (out= 0)(stored 0%)
adding: org/example/Anagramme.class(in = 1584) (out= 876)(deflated 44%)
adding: org/example/Anagramme$Mapper.class(in = 1700) (out= 771)(deflated 54%)
adding: org/example/Anagramme$Reducer.class(in = 1972) (out= 871)(deflated 55%)
root@e3564f682d93:/Classes/exercice1# ls
Anagramme.jar  Anagramme.java  org
root@e3564f682d93:/Classes/exercice1#
```

Prepare the input file

```
GNU nano 7.2                               input.txt
crane
imaginer
surface
metropole
migraine
structure
ancre

[ Wrote 8 lines ]
^G Help      ^O Write Out    ^W Where Is    ^K Cut          ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File     ^\ Replace     ^U Paste        ^J Justify    ^/ Go To Line M-E Redo
```

Let's now copy Fichiers on the partition HDFS

```
root@e3564f682d93:/# hdfs dfs -put Fichiers
2024-12-05 16:51:05,409 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
```

Verify

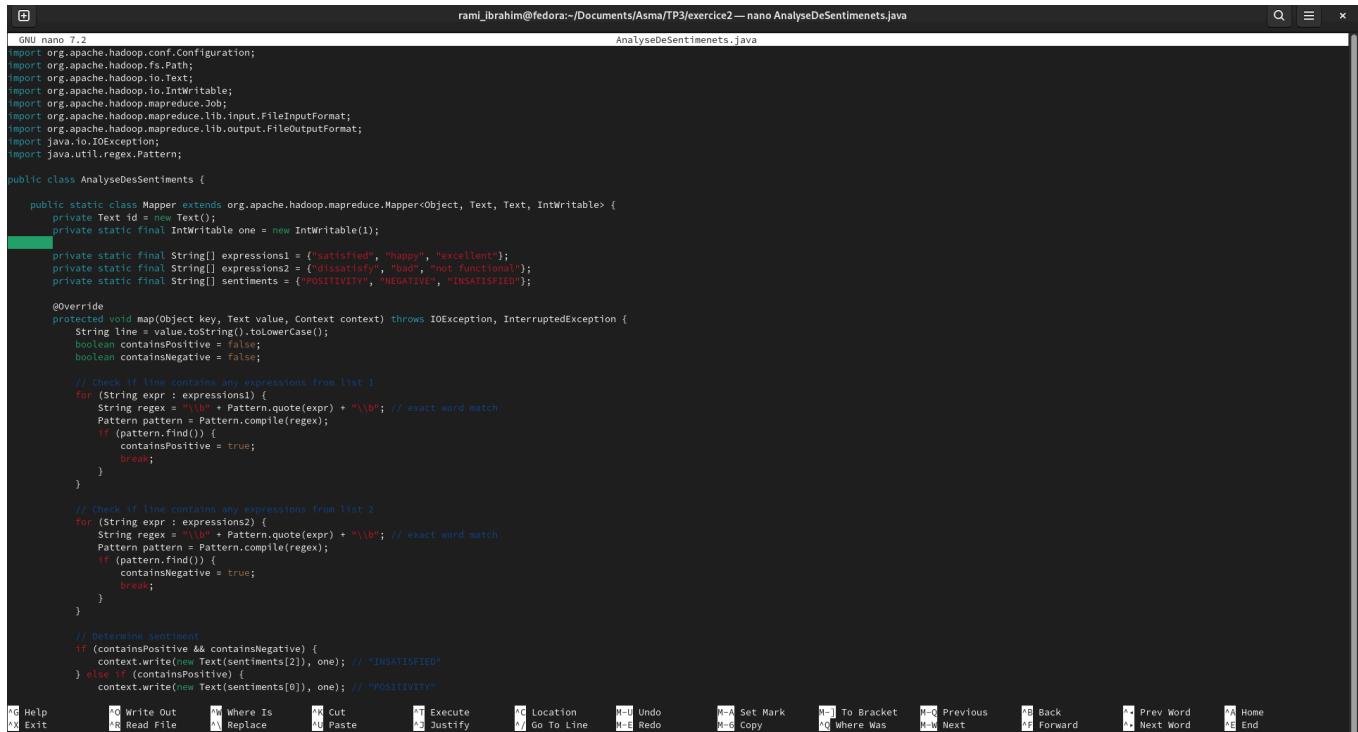
```
Found 4 items
drwxr-xr-x  - root supergroup          0 2024-12-05 16:51 Fichiers
drwxr-xr-x  - root supergroup          0 2024-12-01 20:57 input
drwxr-xr-x  - root supergroup          0 2024-12-01 21:37 output
drwxr-xr-x  - root supergroup          0 2024-12-01 22:13 outputData
root@e3564f682d93:/# hdfs dfs -ls Fichiers
Found 1 items
drwxr-xr-x  - root supergroup          0 2024-12-05 16:51 Fichiers/exercice1
root@e3564f682d93:/#
```

Now let's execute the hadoop program and verify the result

```
root@e3564f682d93:/# hdfs dfs -ls /outputExercice1
Found 2 items
-rw-r--r-- 3 root supergroup          0 2024-12-05 17:00 /outputExercice1/_SUCCESS
-rw-r--r-- 3 root supergroup          47 2024-12-05 17:00 /outputExercice1/part-r-00000
root@e3564f682d93:/# hdfs dfs -cat /outputExercice1/part-r-00000
2024-12-05 17:01:16,388 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
acnre  ancre, crane
aegiimmr  migraine, imaginer
root@e3564f682d93:/#
```

## Exercise 2: Customer sentiment analysis on Twitter

Let's create AnalyseDesSentiments.java



```
GNU nano 7.2                                     AnalyseDesSentiments.java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import java.util.regex.Pattern;
import java.util.regex.Matcher;

public class AnalyseDesSentiments {

    public static class Mapper extends org.apache.hadoop.mapreduce.Mapper<Object, Text, Text, IntWritable> {
        private Text id = new Text();
        private static final IntWritable one = new IntWritable(1);

        private static final String[] expressions1 = {"satisfied", "happy", "excellent"};
        private static final String[] expressions2 = {"disatisfy", "bad", "not functional"};
        private static final String[] sentiments = {"POSITIVITY", "NEGATIVE", "INSATISFIED"};

        @Override
        protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String line = value.toString().toLowerCase();
            boolean containsPositive = false;
            boolean containsNegative = false;

            // Check if line contains any expressions from list 1
            for (String expr : expressions1) {
                String regex = "\\b" + Pattern.quote(expr) + "\\b"; // exact word match
                Pattern pattern = Pattern.compile(regex);
                if (pattern.find()) {
                    containsPositive = true;
                    break;
                }
            }

            // Check if line contains any expressions from list 2
            for (String expr : expressions2) {
                String regex = "\\b" + Pattern.quote(expr) + "\\b"; // exact word match
                Pattern pattern = Pattern.compile(regex);
                if (pattern.find()) {
                    containsNegative = true;
                    break;
                }
            }

            // Determine sentiment
            if (containsPositive && containsNegative) {
                context.write(new Text(sentiments[2]), one); // "INSATISFIED"
            } else if (containsPositive) {
                context.write(new Text(sentiments[0]), one); // "POSITIVITY"
            } else {
                context.write(new Text(sentiments[1]), one); // "NEGATIVE"
            }
        }
    }

    public static class Reducer extends org.apache.hadoop.mapreduce.Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        @Override
        protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
            int count = 0;
            for (IntWritable value : values) {
                count += value.get();
            }
            result.set(count);
            context.write(key, result);
        }
    }
}

public static void main(String[] args) throws Exception {
    if (args.length != 2) {
        System.err.println("Usage: un a besoin de <input path> <output path>");
        System.exit(-1);
    }

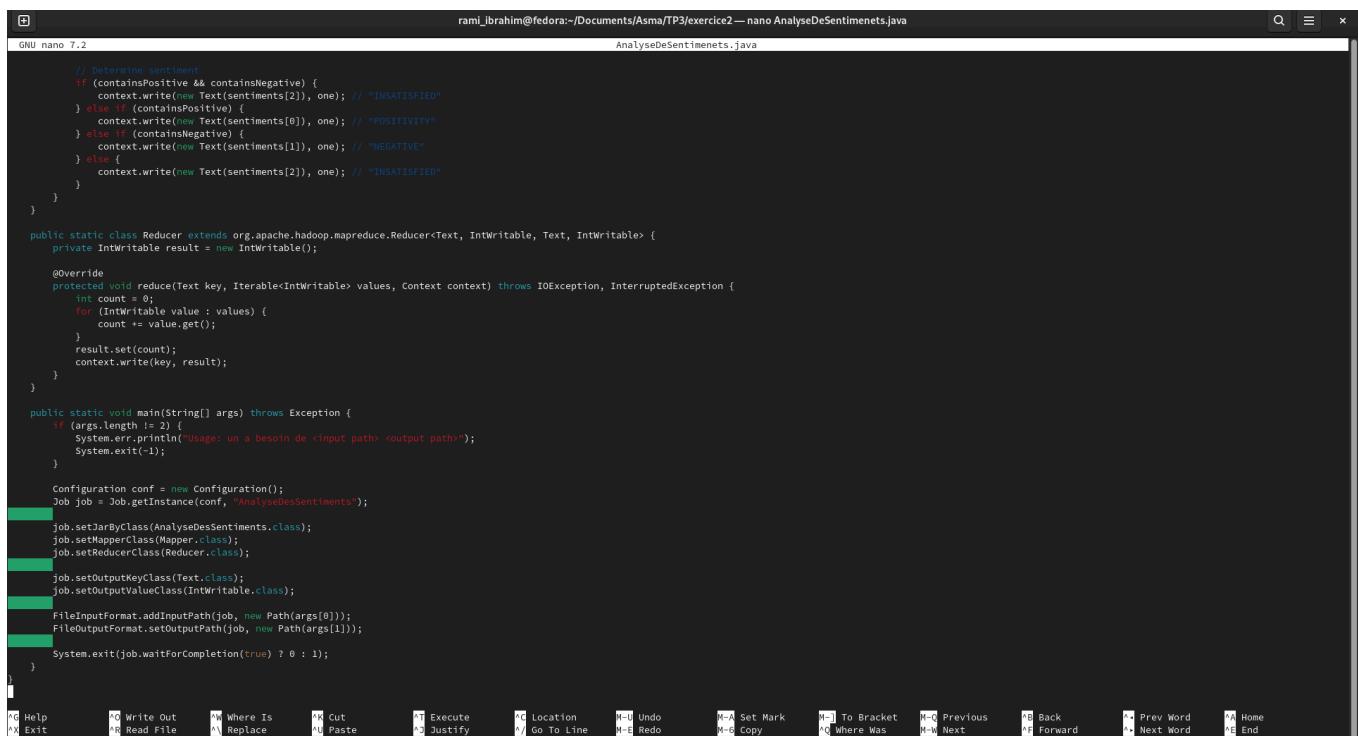
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "AnalyseDesSentiments");

    job.setMapperClass(AnalyseDesSentiments.class);
    job.setMapperClass(Mapper.class);
    job.setReducerClass(Reducer.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);
}
```



```
GNU nano 7.2                                     AnalyseDesSentiments.java
// Determination des points
if (containsPositive && containsNegative) {
    context.write(new Text(sentiments[2]), one); // "INSATISFIED"
} else if (containsPositive) {
    context.write(new Text(sentiments[0]), one); // "POSITIVITY"
} else if (containsNegative) {
    context.write(new Text(sentiments[1]), one); // "NEGATIVE"
} else {
    context.write(new Text(sentiments[2]), one); // "INSATISFIED"
}

public static class Reducer extends org.apache.hadoop.mapreduce.Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        int count = 0;
        for (IntWritable value : values) {
            count += value.get();
        }
        result.set(count);
        context.write(key, result);
    }
}

public static void main(String[] args) throws Exception {
    if (args.length != 2) {
        System.err.println("Usage: un a besoin de <input path> <output path>");
        System.exit(-1);
    }

    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "AnalyseDesSentiments");

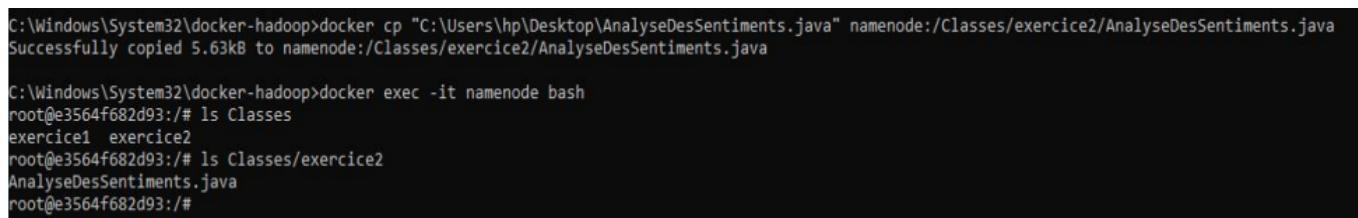
    job.setMapperClass(AnalyseDesSentiments.class);
    job.setMapperClass(Mapper.class);
    job.setReducerClass(Reducer.class);

    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);
}
```

Move the code into the folder



```
C:\Windows\System32\docker-hadoop>docker cp "C:\Users\hp\Desktop\AnalyseDesSentiments.java" namenode:/Classes/exercice2/AnalyseDesSentiments.java
Successfully copied 5.63kB to namenode:/Classes/exercice2/AnalyseDesSentiments.java

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# ls Classes
exercice1 exercice2
root@e3564f682d93:/# ls Classes/exercice2
AnalyseDesSentiments.java
root@e3564f682d93:/#
```

Now let's compile the class .java

```
root@e3564f682d93:/Classes/exercice2# javac -encoding UTF-8 -classpath "$(hadoop classpath)" -d . AnalyseDesSentiments.java
root@e3564f682d93:/Classes/exercice2# ls
AnalyseDesSentiments.java org
root@e3564f682d93:/Classes/exercice2# ls org/example
AnalyseDesSentiments$Mapper.class AnalyseDesSentiments$Reducer.class AnalyseDesSentiments.class
root@e3564f682d93:/Classes/exercice2#
```

Create the jar

```
root@e3564f682d93:/Classes/exercice2# jar -cvf AnalyseDesSentiments.jar -C . .
added manifest
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/example/(in = 0) (out= 0)(stored 0%)
adding: org/example/AnalyseDesSentiments.class(in = 1677) (out= 905)(deflated 46%)
adding: org/example/AnalyseDesSentiments$Reducer.class(in = 1784) (out= 753)(deflated 57%)
adding: org/example/AnalyseDesSentiments$Mapper.class(in = 2389) (out= 1124)(deflated 52%)
adding: AnalyseDesSentiments.java(in = 3794) (out= 1069)(deflated 71%)
root@e3564f682d93:/Classes/exercice2# ls
AnalyseDesSentiments.jar AnalyseDesSentiments.java org
root@e3564f682d93:/Classes/exercice2#
```

Prepare the `input.txt`

```
rami_ibrahim@fedora:~/Docum... ✘ rami_ibrahim@fedora:~/Docum... ✘
GNU nano 7.2                               input.txt
"@acme Votre service client est nul"
"@acme 30min d'attente... tres insatisfait"
"Très satisfait par un produit super !! @acme"
"merci d'avoir TY @acme"
"@acme produit déjà cassé, super insatisfait !"
```

Copy the file into the container

```
C:\Windows\System32\docker-hadoop>docker cp "C:\Users\hp\Desktop\input2.txt" namenode:/Fichiers/exercice2/input.txt
Successfully copied 2.05kB to namenode:/Fichiers/exercice2/input.txt

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# ls Fichiers/exercice2
input.txt
root@e3564f682d93:/#
```

Copy the folder on the partition HDFS

```
root@e3564f682d93:/# hdfs dfs -ls
Found 4 items
drwxr-xr-x  - root supergroup      0 2024-12-05 16:51 Fichiers
drwxr-xr-x  - root supergroup      0 2024-12-01 20:57 input
drwxr-xr-x  - root supergroup      0 2024-12-01 21:37 output
drwxr-xr-x  - root supergroup      0 2024-12-01 22:13 outputData
root@e3564f682d93:/# hdfs dfs -put Fichiers/exercice2
2024-12-06 12:09:18,267 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@e3564f682d93:/# hdfs dfs -ls
2024-12-06 12:09:18,267 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Found 5 items
drwxr-xr-x  - root supergroup      0 2024-12-05 16:51 Fichiers
drwxr-xr-x  - root supergroup      0 2024-12-06 12:09 exercice2
drwxr-xr-x  - root supergroup      0 2024-12-01 20:57 input
drwxr-xr-x  - root supergroup      0 2024-12-01 21:37 output
drwxr-xr-x  - root supergroup      0 2024-12-01 22:13 outputData
root@e3564f682d93:/#
```

Let's execute the Hadoop program

```

Map output bytes=78
Map output materialized bytes=50
Input split bytes=115
Combine input records=0
Combine output records=0
Reduce input groups=3
Reduce shuffle bytes=50
Reduce input records=5
Reduce output records=3
Spilled Records=10
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=76
CPU time spent (ms)=710
Physical memory (bytes) snapshot=473788416
Virtual memory (bytes) snapshot=13555097600
Total committed heap usage (bytes)=401604608
Peak Map Physical memory (bytes)=285028352
Peak Map Virtual memory (bytes)=5102911488
Peak Reduce Physical memory (bytes)=188760064
Peak Reduce Virtual memory (bytes)=8452186112
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=209
File Output Format Counters
  Bytes Written=40
root@e3564f682d93:/Classes/exercice2# hdfs dfs -cat /outputExercice26/part-r-00000
2024-12-06 13:08:12,554 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
inconcluant      2
insatisfait      2
satisfait        1

```

### Exercise 3: Inverted Web Index.

Let's create `IndexWeb.java`

```

GNU nano 7.2                                         rami_ibrahim@fedora:~/Documents/Asma/TP3— nano IndexWeb.java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class IndexWeb {

    public static class Mapper extends org.apache.hadoop.mapreduce.Mapper<Object, Text, Text, Text> {
        private Text cle = new Text();
        private Text valeur = new Text();

        @Override
        protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String[] parts = value.toString().split("\n");
            String dateRefNumber = parts[0].trim();
            String dateRefNumber = referenceNumber.split("-")[0];
            cle.set(dateRefNumber);
            valeur.set(referenceNumber);
            context.write(cle, valeur);
        }
    }

    public static class Reducer extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, Text> {
        private Text result = new Text();

        @Override
        protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            List<String> refs = new ArrayList<>();
            for (Text val : values) {
                refs.add(val.toString());
            }
            if (refs.size() > 0) {
                result.set(String.join(" ", refs));
                context.write(key, result);
            }
        }
    }

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: un a besoin de <input path> <output path>");
            System.exit(-1);
        }
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "HabIMOB");

        Help   W Write Out  W Where Is  C Cut  E Execute  L Location  U Undo  A Set Mark  T To Bracket  B Previous  B Back  P Prev Word  H Home
        Exit  PR Read File  A Replace  P Paste  J Justify  G Go To Line  M-D Redo  M-C Copy  W Where Was  N Next  F Forward  N-1 Next Word  E End
    }
}

```

```

public static void main(String[] args) throws Exception {
    if (args.length != 2) {
        System.err.println("Usage: un a besoin de <input path> <output path>");
        System.exit(-1);
    }
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "HabIMOB");

    job.setMapperClass(HabIMOB_mapper.class);
    job.setMapperClass(Mapper.class);
    job.setReducerClass(Reducer.class);

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

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

    System.exit(job.waitForCompletion(true) ? 0 : 1);
}

Help   W Write Out  W Where Is  C Cut  E Execute  L Location  U Undo  A Set Mark  T To Bracket  B Previous  B Back  P Prev Word  H Home
Exit  PR Read File  A Replace  P Paste  J Justify  G Go To Line  M-D Redo  M-C Copy  W Where Was  N Next  F Forward  N-1 Next Word  E End

```

Move the code into the container

```
C:\Windows\System32\docker-hadoop>docker cp "C:\Users\hp\Desktop\IndexWeb.java" namenode:/Classes/exercice3/IndexWeb.java
Successfully copied 4.61kB to namenode:/Classes/exercice3/IndexWeb.java

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# ls Classes/exercice3
IndexWeb.java
```

Compile the java file

```
root@e3564f682d93:/# cd Classes/exercice3
root@e3564f682d93:/Classes/exercice3# javac -encoding UTF-8 -classpath "$(hadoop classpath)" -d . IndexWeb.java
root@e3564f682d93:/Classes/exercice3# ls
IndexWeb.java org
root@e3564f682d93:/Classes/exercice3# ls org/example
IndexWeb$Mapper.class IndexWeb$Reducer.class IndexWeb.class
root@e3564f682d93:/Classes/exercice3#
```

Create the jar file

```
root@e3564f682d93:/Classes/exercice3# jar -cvf IndexWeb.jar -C .
added manifest
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/example/(in = 0) (out= 0)(stored 0%)
adding: org/example/IndexWeb$Mapper.class(in = 1771) (out= 790)(deflated 55%)
adding: org/example/IndexWeb$Reducer.class(in = 1965) (out= 864)(deflated 56%)
adding: org/example/IndexWeb.class(in = 1579) (out= 877)(deflated 44%)
adding: IndexWeb.java(in = 2582) (out= 829)(deflated 67%)
root@e3564f682d93:/Classes/exercice3# ls
IndexWeb.jar IndexWeb.java org
root@e3564f682d93:/Classes/exercice3#
```

Prepare the input.txt

```
rami_ibrahim@fedora:~/Docum... × rami_ibrahim@fedora:~/Docum... ×
GNU nano 7.2                                     input.txt
P1 : P2 , P3
P2 : P1 , P4
P3 : P1 , P4
P4 : P1 , P2 , P3 █

[ Wrote 4 lines ]
^G Help      ^O Write Out    ^W Where Is    ^K Cut        ^T Execute
^X Exit      ^R Read File   ^V Replace     ^U Paste      ^J Justify
```

Move it into the container

```
C:\Windows\System32\docker-hadoop>docker cp "C:\Users\hp\Desktop\input3.txt" namenode:/Fichiers/exercice3/input.txt
Successfully copied 2.05kB to namenode:/Fichiers/exercice3/input.txt

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# ls Fichiers/exercice3
input.txt
root@e3564f682d93:/#
```

Move it on the HDFS partition

```
root@e3564f682d93:/# hdfs dfs -put Fichiers/exercice3
2024-12-06 13:38:18,236 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@e3564f682d93:/# hdfs dfs -ls
Found 6 items
drwxr-xr-x  - root supergroup          0 2024-12-05 16:51 Fichiers
drwxr-xr-x  - root supergroup          0 2024-12-06 13:03 exercice2
drwxr-xr-x  - root supergroup          0 2024-12-06 13:38 exercice3
drwxr-xr-x  - root supergroup          0 2024-12-01 20:57 input
drwxr-xr-x  - root supergroup          0 2024-12-01 21:37 output
drwxr-xr-x  - root supergroup          0 2024-12-01 22:13 outputData
root@e3564f682d93:/#
```

Execute the hadoop program and verify

```

root@e3564f682d93:/Classes/exercice3# hdfs dfs -ls
Found 6 items
drwxr-xr-x - root supergroup          0 2024-12-05 16:51 Fichiers
drwxr-xr-x - root supergroup          0 2024-12-06 13:03 exercice2
drwxr-xr-x - root supergroup          0 2024-12-06 13:38 exercice3
drwxr-xr-x - root supergroup          0 2024-12-01 20:57 input
drwxr-xr-x - root supergroup          0 2024-12-01 21:37 output
drwxr-xr-x - root supergroup          0 2024-12-01 22:13 outputData
root@e3564f682d93:/Classes/exercice3# hdfs dfs -ls /
Found 13 items
drwxr-xr-x - root supergroup          0 2024-11-29 13:53 /TPs
drwxrwxrwt - root root               0 2024-12-01 21:05 /app-logs
drwxr-xr-x - root supergroup          0 2024-12-05 17:00 /outputExercice1
drwxr-xr-x - root supergroup          0 2024-12-06 12:13 /outputExercice2
drwxr-xr-x - root supergroup          0 2024-12-06 12:24 /outputExercice22
drwxr-xr-x - root supergroup          0 2024-12-06 12:30 /outputExercice222
drwxr-xr-x - root supergroup          0 2024-12-06 12:39 /outputExercice24
drwxr-xr-x - root supergroup          0 2024-12-06 13:04 /outputExercice25
drwxr-xr-x - root supergroup          0 2024-12-06 13:07 /outputExercice26
drwxr-xr-x - root supergroup          0 2024-12-06 13:40 /outputExercice3
drwxr-xr-x - root supergroup          0 2024-11-28 15:53 /rmstate
drwxr----- - root supergroup          0 2024-12-01 21:05 /tmp
drwxr-xr-x - root supergroup          0 2024-11-29 13:00 /user
root@e3564f682d93:/Classes/exercice3# hdfs dfs -ls /outputExercice3
Found 2 items
-rw-r--r--  3 root supergroup          0 2024-12-06 13:40 /outputExercice3/_SUCCESS
-rw-r--r--  3 root supergroup          44 2024-12-06 13:40 /outputExercice3/part-r-00000
root@e3564f682d93:/Classes/exercice3# hdfs dfs -cat /outputExercice3/part-r-00000
2024-12-06 13:40:54,466 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
P1      P4, P3, P2
P2      P4, P1
P3      P4, P1
P4      P3, P2
root@e3564f682d93:/Classes/exercice3# 

```

## Exercise 4: Average Temperatures

Let's create MoyenneTemperature.java

```

GNU nano 7.2                                     MoyenneTemperature.java
Modified

package org.example;

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

public class MoyenneTemperature {

    public static class Mapper extends org.apache.hadoop.mapreduce.Mapper<Object, Text, Text, IntWritable> {
        private Text cle = new Text();
        private IntWritable valeur = new IntWritable();

        @Override
        protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String[] parts = value.toString().split(",");
            String temperature = parts[1].trim();
            cle.set(temperature);
            valeur.set(Integer.parseInt(temperature));
            context.write(cle, valeur);
        }
    }

    public static class Reducer extends org.apache.hadoop.mapreduce.Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        @Override
        protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
            int sum = 0;
            int count = 0;
            for (IntWritable value : values) {
                sum += value.get();
                count++;
            }
            result.set(sum/count);
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: besoin de <input path> <output path>");
            System.exit(-1);
        }

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "MoyenneTemperature");
    }
}


```

```

Job job = Job.getInstance(conf, "MoyenneTemperature");
job.setJarByClass(MoyenneTemperature.class);
job.setMapperClass(Mapper.class);
job.setReducerClass(Reducer.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);
}

```

Move the code java file into the folder

```
C:\Windows\System32\docker-hadoop>docker cp "C:\Users\hp\Desktop\MoyenneTemperature.java" namenode:/Classes/exercice4/MoyenneTemperature.java
Successfully copied 4.61kB to namenode:/Classes/exercice4/MoyenneTemperature.java

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# cd Classes/exercice4
root@e3564f682d93:/Classes/exercice4# ls
MoyenneTemperature.java
root@e3564f682d93:/Classes/exercice4#
```

Compile the code

```
root@e3564f682d93:/Classes/exercice4# javac -encoding UTF-8 -classpath "$(hadoop classpath)" -d . MoyenneTemperature.java
root@e3564f682d93:/Classes/exercice4# ls
MoyenneTemperature.java  org
root@e3564f682d93:/Classes/exercice4# ls org/example
MoyenneTemperature$Mapper.class  MoyenneTemperature$Reducer.class  MoyenneTemperature.class
root@e3564f682d93:/Classes/exercice4#
```

Create the jar file

```
root@e3564f682d93:/Classes/exercice4# jar -cvf MoyenneTemperature.jar -C .
added manifest
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/example/(in = 0) (out= 0)(stored 0%)
adding: org/example/MoyenneTemperature.class(in = 1667) (out= 902)(deflated 45%)
adding: org/example/MoyenneTemperature$Reducer.class(in = 1706) (out= 721)(deflated 57%)
adding: org/example/MoyenneTemperature$Mapper.class(in = 1850) (out= 768)(deflated 58%)
adding: MoyenneTemperature.java(in = 2652) (out= 807)(deflated 69%)
root@e3564f682d93:/Classes/exercice4# ls
MoyenneTemperature.jar  MoyenneTemperature.java  org
```

Prepare the input.txt

```
rami_ibrahim@fedora:~/Docum...  x      rami_ibrahim@fedora:~/Docum...  x
                                         input.txt

GNU nano 7.2
200606,90
200708,100
200701,95
200501,100
200701,80

[ Wrote 5 lines ]
^G Help      ^O Write Out    ^W Where Is    ^K Cut        ^T Exec
^X Exit      ^R Read File   ^\ Replace     ^U Paste      ^J Justify
```

Move input.txt into the folder

```
C:\Windows\System32\docker-hadoop>docker cp "C:\Users\hp\Desktop\input4.txt" namenode:/Fichiers/exercice4/input.txt
Successfully copied 2.05kB to namenode:/Fichiers/exercice4/input.txt

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# ls Fichiers/exercice4
input.txt
root@e3564f682d93:/#
```

Move it on the HDFS partition

```
root@e3564f682d93:/# hdfs dfs -put Fichiers/exercice4
2024-12-06 14:05:05,793 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@e3564f682d93:/# hdfs dfs -ls
Found 7 items
drwxr-xr-x  - root supergroup          0 2024-12-05 16:51 Fichiers
drwxr-xr-x  - root supergroup          0 2024-12-06 13:03 exercice2
drwxr-xr-x  - root supergroup          0 2024-12-06 13:38 exercice3
drwxr-xr-x  - root supergroup          0 2024-12-06 14:05 exercice4
drwxr-xr-x  - root supergroup          0 2024-12-01 20:57 input
drwxr-xr-x  - root supergroup          0 2024-12-01 21:37 output
drwxr-xr-x  - root supergroup          0 2024-12-01 22:13 outputData
root@e3564f682d93:/#
```

Execute the hadoop program and verify

```

root@e3564f682d93:/Classes/exercice4# hdfs dfs -ls /
Found 14 items
drwxr-xr-x  - root supergroup          0 2024-11-29 13:53 /TPs
drwxrwxrwt  - root root              0 2024-12-01 21:05 /app-logs
drwxr-xr-x  - root supergroup          0 2024-12-05 17:00 /outputExercice1
drwxr-xr-x  - root supergroup          0 2024-12-06 12:13 /outputExercice2
drwxr-xr-x  - root supergroup          0 2024-12-06 12:24 /outputExercice22
drwxr-xr-x  - root supergroup          0 2024-12-06 12:30 /outputExercice222
drwxr-xr-x  - root supergroup          0 2024-12-06 12:39 /outputExercice24
drwxr-xr-x  - root supergroup          0 2024-12-06 13:04 /outputExercice25
drwxr-xr-x  - root supergroup          0 2024-12-06 13:07 /outputExercice26
drwxr-xr-x  - root supergroup          0 2024-12-06 13:40 /outputExercice3
drwxr-xr-x  - root supergroup          0 2024-12-06 14:07 /outputExercice4
drwxr-xr-x  - root supergroup          0 2024-11-28 15:53 /rmstate
drwx-----  - root supergroup          0 2024-12-01 21:05 /tmp
drwxr-xr-x  - root supergroup          0 2024-11-29 13:00 /user
root@e3564f682d93:/Classes/exercice4# hdfs dfs -ls /outputExercice4
Found 2 items
-rw-r--r--  3 root supergroup          0 2024-12-06 14:07 /outputExercice4/_SUCCESS
-rw-r--r--  3 root supergroup          25 2024-12-06 14:07 /outputExercice4/part-r-00000
root@e3564f682d93:/Classes/exercice4# hdfs dfs -cat /outputExercice4/part-r-00000
2024-12-06 14:08:16,050 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2005   100
2006    90
2007    91

```

## Exercice 5 : Moyenne des températures avec Combiner

Let's create MoyenneTemperatureAvecCompteur.java

```

GNU nano 7.2
rami_ibrahim@fedora:~/Documents/Asma/TP3— nano MoyenneTemperatureAvecCompteur.java

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

public class MoyenneTemperatureAvecCompteur {

    public static class TemperatureMapper extends org.apache.hadoop.mapreduce.Mapper<Object, Text, Text, Text> {
        private Text year = new Text();
        private Text temperature = new Text();

        @Override
        protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String[] parts = value.toString().split(",");
            String[] dateTemp = parts[0].split("/");
            year.set(dateTemp[2]);
            temperature.set(parts[1]); // temperature
            context.write(year, temperature);
        }
    }

    public static class TemperatureCombiner extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, IntWritable> {
        private Text result = new Text();

        @Override
        protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            int totalTemp = 0;
            int count = 0;
            for (Text value : values) {
                totalTemp += Integer.parseInt(value.toString());
                count++;
            }
            result.set((totalTemp + " " + count));
            context.write(key, result);
        }
    }

    public static class TemperatureReducer extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        @Override
        protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            int totalTemp = 0;
            int total = 0;
            for (Text value : values) {

```

For (Text value : values) {  
String[] parts = value.toString().split(",");  
totalTemp += Integer.parseInt(parts[0]);  
total += Integer.parseInt(parts[1]);  
}  
result.set((totalTemp/total));  
context.write(key, result);  
}  
}

```

public static void main(String[] args) throws Exception {
    if (args.length != 2) {
        System.err.println("Usage: besoin de <input path> <output path>");
        System.exit(-1);
    }

    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "moyenneTemperatureAvecCompteur");

    job.setJarByClass(MoyenneTemperatureAvecCompteur.class);
    job.setMapperClass(TemperatureMapper.class);
    job.setCombinerClass(TemperatureCombiner.class);
    job.setReducerClass(TemperatureReducer.class);

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

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

    System.exit(job.waitForCompletion(true) ? 0 : 1);
}

```

Move the java file into the folder

```

C:\Windows\System32\docker-hadoop>docker cp "C:\Users\hp\Desktop\MoyenneTemperatureAvecCompteur.java" namenode:/Classes/exercice5/MoyenneTemperatureAvecCompteur.java
Successfully copied 5.12kB to namenode:/Classes/exercice5/MoyenneTemperatureAvecCompteur.java

C:\Windows\System32\docker-hadoop>docker exec -it namenode bash
root@e3564f682d93:/# ls Classes/exercice5
MoyenneTemperatureAvecCompteur.java
root@e3564f682d93:# 

```

## Compile the code

```
root@e3564f682d93:/# cd Classes/exercice5
root@e3564f682d93:/Classes/exercice5# javac -encoding UTF-8 -classpath "$(hadoop classpath)" -d . MoyenneTemperatureAvecCombiner.java
root@e3564f682d93:/Classes/exercice5# ls
MoyenneTemperatureAvecCombiner.java  org
root@e3564f682d93:/Classes/exercice5# ls org/example
MoyenneTemperatureAvecCombiner$TemperatureCombiner.class  MoyenneTemperatureAvecCombiner$TemperatureReducer.class
MoyenneTemperatureAvecCombiner$TemperatureMapper.class  MoyenneTemperatureAvecCombiner.class
root@e3564f682d93:/Classes/exercice5#
```

## Create the jar file

```
root@e3564f682d93:/Classes/exercice5# jar -cvf MoyenneTemperatureAvecCombiner.jar -C .
added manifest
adding: MoyenneTemperatureAvecCombiner.java(in = 3524) (out= 1033)(deflated 70%)
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/example/(in = 0) (out= 0)(stored 0%)
adding: org/example/MoyenneTemperatureAvecCombiner$TemperatureReducer.class(in = 2047) (out= 873)(deflated 57%)
adding: org/example/MoyenneTemperatureAvecCombiner$TemperatureCombiner.class(in = 2065) (out= 881)(deflated 57%)
adding: org/example/MoyenneTemperatureAvecCombiner$TemperatureMapper.class(in = 1875) (out= 757)(deflated 59%)
adding: org/example/MoyenneTemperatureAvecCombiner.class(in = 1912) (out= 961)(deflated 49%)
root@e3564f682d93:/Classes/exercice5# ls
MoyenneTemperatureAvecCombiner.jar  MoyenneTemperatureAvecCombiner.java  org
  org/example/MoyenneTemperatureAvecCombiner$TemperatureReducer.class  org/example/MoyenneTemperatureAvecCombiner$TemperatureCombiner.class  org/example/MoyenneTemperatureAvecCombiner$TemperatureMapper.class  org/MoyenneTemperatureAvecCombiner.class
```

## Execute the hadoop program and verify

```
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=94
CPU time spent (ms)=720
Physical memory (bytes) snapshot=481738752
Virtual memory (bytes) snapshot=13558095872
Total committed heap usage (bytes)=406847488
Peak Map Physical memory (bytes)=295260160
Peak Map Virtual memory (bytes)=5106376704
Peak Reduce Physical memory (bytes)=186478592
Peak Reduce Virtual memory (bytes)=8451719168
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=55
File Output Format Counters
  Bytes Written=25
root@e3564f682d93:/Classes/exercice5# hdfs dfs -ls /outputExercice5
root@e3564f682d93:/Classes/exercice5# hdfs dfs -ls /outputExercice55
Found 2 items
-rw-r--r--  3 root supergroup          0 2024-12-06 15:12 /outputExercice55/_SUCCESS
-rw-r--r--  3 root supergroup        25 2024-12-06 15:12 /outputExercice55/part-r-00000
root@e3564f682d93:/Classes/exercice5# hdfs dfs -cat /outputExercice55/part-r-00000
2024-12-06 15:12:47,314 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2005   100
2006    90
2007    91
```