

1. Wordcount Program

Mapper Code: You have to copy paste this program into the WCMapper Java Class file.

```
// Importing libraries
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper<LongWritable,
                                                 Text, Text,
                                                 IntWritable> {

    // Map function
    public void map(LongWritable key, Text value, OutputCollector<Text,
                    IntWritable> output, Reporter rep) throws IOException
    {

        String line = value.toString();

        // Splitting the line on spaces
        for (String word : line.split(" "))
        {
            if (word.length() > 0)
            {
                output.collect(new Text(word), new IntWritable(1));
            }
        }
    }
}
```

Reducer Code: You have to copy paste this program into the WCReducer Java Class file

```
// Importing libraries
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer<Text,
                                                 IntWritable, Text, IntWritable> {

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> value,
```

```

        OutputCollector<Text, IntWritable> output,
                    Reporter rep) throws IOException
{
    int count = 0;

    // Counting the frequency of each words
    while (value.hasNext())
    {
        IntWritable i = value.next();
        count += i.get();
    }

    output.collect(key, new IntWritable(count));
}
}

```

Driver Code: You have to copy paste this program into the WCDriver Java Class file.

```

// Importing libraries
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

    public int run(String args[]) throws IOException
    {
        if (args.length < 2)
        {
            System.out.println("Please give valid inputs");
            return -1;
        }

        JobConf conf = new JobConf(WCDriver.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
    }
}

```

```
        conf.setOutputValueClass(IntWritable.class);
        JobClient.runJob(conf);
        return 0;
    }

    // Main Method
    public static void main(String args[]) throws Exception
    {
        int exitCode = ToolRunner.run(new WCDriver(), args);
        System.out.println(exitCode);
    }
}
```

1.hadoop fs -copyFromLocal /home/hduser/Desktop/file1.txt
/rgs/test.txt

2.hadoop jar /home/hduser/Desktop/MapReduceClient.jar
WCDriver /pgb/test.txt /output/

3.hdfs dfs -cat /output/*

```
Administrator: Command Prompt
2021-04-24 14:55:13,844 INFO common.Storage: Storage directory C:\hadoop-3.3.0\data\namenode has been successfully formatted.
2021-04-24 14:55:13,895 INFO namenode.FSImageFormatProtobuf: Saving image file C:\hadoop-3.3.0\data\namenode\current\fsimage.ckpt_000000000000 using no compression
2021-04-24 14:55:14,002 INFO namenode.FSImageFormatProtobuf: Image file C:\hadoop-3.3.0\data\namenode\current\fsimage.ckpt_000000000000 of size 402 bytes saved in 0 seconds .
2021-04-24 14:55:14,115 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2021-04-24 14:55:14,121 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2021-04-24 14:55:14,121 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************/
SHUTDOWN_MSG: Shutting down NameNode at LAPTOP-JG329ESD/192.168.56.1
/************************************************************/

C:\hadoop-3.3.0\sbin>start-dfs

C:\hadoop-3.3.0\sbin>start-yarn
starting yarn daemons

C:\hadoop-3.3.0\sbin>jps
12276 NameNode
14776 DataNode
15512 NodeManager
1800 Jps
6764 ResourceManager

C:\hadoop-3.3.0\sbin>hdfs dfs -mkdir /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /
Found 1 items
drwxr-xr-x - Anusree supergroup          0 2021-04-24 14:56 /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -copyFromLocal C:\input_file.txt /input_dir
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/input_file.txt
Hello World
Hello Hadoop
This is Hadoop test file
C:\hadoop-3.3.0\sbin>hadoop jar C:\MapReduceClient.jar wordcount /input_dir /output_dir
2021-04-24 15:24:57,242 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-24 15:24:57,714 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging
/job_1619256355508_0002
2021-04-24 15:24:58,387 INFO input.FileInputFormat: Total input files to process : 1
2021-04-24 15:24:58,809 INFO mapreduce.JobSubmitter: number of splits:1
2021-04-24 15:24:59,255 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619256355508_0002
2021-04-24 15:24:59,255 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-24 15:24:59,450 INFO conf.Configuration: resource-types.xml not found
2021-04-24 15:24:59,451 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-24 15:24:59,533 INFO impl.YarnClientImpl: Submitted application application_1619256355508_0002
2021-04-24 15:24:59,581 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1619256355508_0002/
2021-04-24 15:24:59,582 INFO mapreduce.Job: Running job: job_1619256355508_0002
2021-04-24 15:25:12,857 INFO mapreduce.Job: Job job_1619256355508_0002 running in uber mode : false
2021-04-24 15:25:12,861 INFO mapreduce.Job: map 0% reduce 0%
2021-04-24 15:25:19,985 INFO mapreduce.Job: map 100% reduce 0%
2021-04-24 15:25:26,077 INFO mapreduce.Job: map 100% reduce 100%
2021-04-24 15:25:32,181 INFO mapreduce.Job: Job job_1619256355508_0002 completed successfully
2021-04-24 15:25:32,284 INFO mapreduce.Job: Counters: 54
    File System Counters
        FILE: Number of bytes read=85
        FILE: Number of bytes written=530945
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=162
        HDFS: Number of bytes written=51
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /output_dir/*
Hadoop 2
Hello 2
This 1
World 1
file 1
is 1
test 1
```

```
C:\hadoop-3.3.0\sbin>
```

← → C ① localhost:8088/cluster



All Applications

Cluster Metrics											
About	Apps Submitted 1										
Nodes											
Nodes Labels	Apps Pending 0										
Applications											
NEW	Apps Running 1										
NEW SAVING	Apps Completed 0										
SUBMITTED	Containers Running 0										
ACCEPTED	Memory Used 0 B										
RUNNING	Cluster Nodes Metrics										
FINISHED	Active Nodes 1										
FAILED	Decommissioning Nodes 0										
KILLED	Decommissioned Nodes 0										
Scheduler Metrics											
Scheduler	Lost Nodes 0										
Scheduler Type											
Capacity Scheduler	Scheduling Resource Type [memory-mb (unit=Mi), vcores]										
Minimum Allocation <memory:1024, vCores:1>											
Show 20 entries											
ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalSta
application_1619256355508_0001	Anusree	word count	MAPREDUCE	default	0	Sat Apr 24 14:58:44 +0550 2021	Sat Apr 24 14:58:46 +0550 2021	Sat Apr 24 14:59:25 +0550 2021	FINISHED	SUCCESS	

Showing 1 to 1 of 1 entries

- . For a given Text file, create a Map Reduce program to sort the content in an alphabetic order listing only top 'n' maximum occurrence 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));
    }
}
}

```

```

C:\hadoop-3.3.0\sbin>jps
11072 DataNode
20528 Jps
5620 ResourceManager
15532 NodeManager
6140 NameNode

C:\hadoop-3.3.0\sbin>hdfs dfs -mkdir /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /
Found 1 items
drwxr-xr-x  - Anusree supergroup      0 2021-05-08 19:46 /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -copyFromLocal C:\input.txt /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /input_dir
Found 1 items
-rw-r--r--  1 Anusree supergroup      36 2021-05-08 19:48 /input_dir/input.txt

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/input.txt
hello
world
hello
hadoop
bye

```

```
C:\hadoop-3.3.0\sbin>hadoop jar C:\sort.jar samples.topN /input_dir/input.txt /output_dir
2021-05-08 19:54:56,482 INFO client.DefaultHttpClient: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-08 19:54:55,291 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1620483374279_0001
2021-05-08 19:54:55,821 INFO input.FileInputFormat: Total input files to process : 1
2021-05-08 19:54:56,261 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1620483374279_0001
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-08 19:54:56,843 INFO conf.Configuration: resource-types.xml not found
2021-05-08 19:54:56,843 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-08 19:54:57,387 INFO impl.YarnClientImpl: Submitted application application_1620483374279_0001
2021-05-08 19:54:57,507 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1620483374279_0001/
2021-05-08 19:54:57,508 INFO mapreduce.Job: Running job: job_1620483374279_0001
2021-05-08 19:55:13,792 INFO mapreduce.Job: Job job_1620483374279_0001 running in uber mode : false
2021-05-08 19:55:13,794 INFO mapreduce.Job: map 0% reduce 0%
2021-05-08 19:55:20,020 INFO mapreduce.Job: map 100% reduce 0%
2021-05-08 19:55:27,116 INFO mapreduce.Job: map 100% reduce 100%
2021-05-08 19:55:33,199 INFO mapreduce.Job: Job job_1620483374279_0001 completed successfully
2021-05-08 19:55:33,334 INFO mapreduce.Job: Counters: 54
    File System Counters
        FILE: Number of bytes read=65
        FILE: Number of bytes written=530397
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=142
        HDFS: Number of bytes written=31
        HDFS: Number of read operations=8
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
        HDFS: Number of bytes read erasure-coded=0
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /output_dir/*
hello    2
hadoop   1
world    1
bye      1
C:\hadoop-3.3.0\sbin>
```

3. From the following link extract the weather data

<https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all>
Create a Map Reduce program to

- a) find average temperature for each year from NCDC data set.

AverageDriver

```
package temp;

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

public class AverageDriver {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
```

```

        System.err.println("Please Enter the input and output parameters");
        System.exit(-1);
    }
    Job job = new Job();
    job.setJarByClass(AverageDriver.class);
    job.setJobName("Max temperature");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.setMapperClass(AverageMapper.class);
    job.setReducerClass(AverageReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

AverageMapper
package temp;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class AverageMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    public static final int MISSING = 9999;

    public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, IntWritable>.Context context) throws IOException, InterruptedException {
        int temperature;
        String line = value.toString();
        String year = line.substring(15, 19);
        if (line.charAt(87) == '+') {
            temperature = Integer.parseInt(line.substring(88, 92));
        } else {
            temperature = Integer.parseInt(line.substring(87, 92));
        }
        String quality = line.substring(92, 93);
        if (temperature != 9999 && quality.matches("[01459]"))
            context.write(new Text(year), new IntWritable(temperature));
    }
}

```

AverageReducer

```

package temp;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AverageReducer 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 max_temp = 0;
        int count = 0;
        for (IntWritable value : values) {
            max_temp += value.get();
            count++;
        }
        context.write(key, new IntWritable(max_temp / count));
    }
}

```

```

        }
        context.write(key, new IntWritable(max_temp / count));
    }
}

```

```

C:\hadoop-3.3.0\sbin>hadoop jar C:\avgtemp.jar temp.AverageDriver /input_dir/temp.txt /avgtemp_outputdir
2021-05-15 14:52:50,635 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-15 14:52:51,005 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-15 14:52:51,111 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621060230696_0005
2021-05-15 14:52:51,735 INFO input.FileInputFormat: Total input files to process : 1
2021-05-15 14:52:52,751 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621060230696_0005
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-15 14:52:53,237 INFO conf.Configuration: resource-types.xml not found
2021-05-15 14:52:53,238 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-15 14:52:53,312 INFO impl.YarnClientImpl: Submitted application application_1621060230696_0005
2021-05-15 14:52:53,352 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621060230696_0005/
2021-05-15 14:52:53,353 INFO mapreduce.Job: Running job: job_1621060230696_0005
2021-05-15 14:53:06,640 INFO mapreduce.Job: Job job_1621060230696_0005 running in uber mode : false
2021-05-15 14:53:06,643 INFO mapreduce.Job: map 0% reduce 0%
2021-05-15 14:53:12,758 INFO mapreduce.Job: map 100% reduce 0%
2021-05-15 14:53:19,860 INFO mapreduce.Job: map 100% reduce 100%
2021-05-15 14:53:25,967 INFO mapreduce.Job: Job job_1621060230696_0005 completed successfully
2021-05-15 14:53:26,096 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=72210
    FILE: Number of bytes written=674341
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=894860
    HDFS: Number of bytes written=8
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=3782

```

```

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /avgtemp_outputdir
Found 2 items
-rw-r--r-- 1 Anusree supergroup          0 2021-05-15 14:53 /avgtemp_outputdir/_SUCCESS
-rw-r--r-- 1 Anusree supergroup          8 2021-05-15 14:53 /avgtemp_outputdir/part-r-00000

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /avgtemp_outputdir/part-r-00000
1901      46

C:\hadoop-3.3.0\sbin>

```

b) find the mean max temperature for every month

MeanMax

MeanMaxDriver.class

```
package meanmax;
```

```

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.lib.input.FileInputFormat;

```

```

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

public class MeanMaxDriver {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Please Enter the input and output parameters");
            System.exit(-1);
        }
        Job job = new Job();
        job.setJarByClass(MeanMaxDriver.class);
        job.setJobName("Max temperature");
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        job.setMapperClass(MeanMaxMapper.class);
        job.setReducerClass(MeanMaxReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

MeanMaxMapper.class

```

package meanmax;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MeanMaxMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    public static final int MISSING = 9999;

    public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, IntWritable>.Context context) throws IOException, InterruptedException {
        int temperature;
        String line = value.toString();
        String month = line.substring(19, 21);
        if (line.charAt(87) == '+') {
            temperature = Integer.parseInt(line.substring(88, 92));
        } else {
            temperature = Integer.parseInt(line.substring(87, 92));
        }
        String quality = line.substring(92, 93);
        if (temperature != 9999 && quality.matches("[01459]")) {
            context.write(new Text(month), new IntWritable(temperature));
        }
    }
}

```

MeanMaxReducer.class

```

package meanmax;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MeanMaxReducer 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 max_temp = 0;
        int total_temp = 0;
        int count = 0;

```

```

        int days = 0;
        for (IntWritable value : values) {
            int temp = value.get();
            if (temp > max_temp)
                max_temp = temp;
            count++;
            if (count == 3) {
                total_temp += max_temp;
                max_temp = 0;
                count = 0;
                days++;
            }
        }
        context.write(key, new IntWritable(total_temp / days));
    }
}

```

```

C:\hadoop-3.3.0\sbin>hadoop jar C:\meanmax.jar meanmax.MeanMaxDriver /input_dir/temp.txt /meanmax_output
2021-05-21 20:28:05,250 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-21 20:28:06,662 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-21 20:28:06,916 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621608943095_0001
2021-05-21 20:28:08,426 INFO input.FileInputFormat: Total input files to process : 1
2021-05-21 20:28:09,107 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621608943095_0001
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-21 20:28:10,029 INFO conf.Configuration: resource-types.xml not found
2021-05-21 20:28:10,030 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-21 20:28:10,676 INFO impl.YarnClientImpl: Submitted application application_1621608943095_0001
2021-05-21 20:28:11,005 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621608943095_0001/
2021-05-21 20:28:11,006 INFO mapreduce.Job: Running job: job_1621608943095_0001
2021-05-21 20:28:29,385 INFO mapreduce.Job: Job job_1621608943095_0001 running in uber mode : false
2021-05-21 20:28:29,389 INFO mapreduce.Job: map 0% reduce 0%
2021-05-21 20:28:40,664 INFO mapreduce.Job: map 100% reduce 0%
2021-05-21 20:28:50,832 INFO mapreduce.Job: map 100% reduce 100%
2021-05-21 20:28:58,965 INFO mapreduce.Job: Job job_1621608943095_0001 completed successfully
2021-05-21 20:28:59,178 INFO mapreduce.Job: Counters: 54
File System Counters
  FILE: Number of bytes read=59082
  FILE: Number of bytes written=648091
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=894860
  HDFS: Number of bytes written=74
  HDFS: Number of read operations=8
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=8077
  Total time spent by all reduces in occupied slots (ms)=7511
  Total time spent by all map tasks (ms)=8077
  Total time spent by all reduce tasks (ms)=7511
  Total vcore-milliseconds taken by all map tasks=8077
  Total vcore-milliseconds taken by all reduce tasks=7511
  Total megabyte-milliseconds taken by all map tasks=8270848
  Total megabyte-milliseconds taken by all reduce tasks=7691264

```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /meanmax_output/*
01      4
02      0
03      7
04     44
05    100
06   168
07   219
08   198
09   141
10   100
11    19
12     3

C:\hadoop-3.3.0\sbin>
```

4. Create a Hadoop Map Reduce program to combine information from the users file along with Information from the posts file by using the concept of join and display user_id, Reputation and Score.

Create a Hadoop Map Reduce program to combine information from the users file along with Information from the posts file by using the concept of join and display user_id, Reputation and Score.

```
// JoinDriver.java
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.mapred.lib.MultipleInputs;
import org.apache.hadoop.util.*;

public class JoinDriver extends Configured implements Tool {

    public static class KeyPartitioner implements Partitioner<TextPair, Text> {
        @Override
```

```
public void configure(JobConf job) {}

@Override
public int getPartition(TextPair key, Text value, int numPartitions) {
    return (key.getFirst().hashCode() & Integer.MAX_VALUE) %
        numPartitions;
}
}

@Override

public int run(String[] args) throws Exception {

if (args.length != 3) {
    System.out.println("Usage: <Department Emp Strength input>
<Department Name input> <output>");
    return -1;
}

JobConf conf = new JobConf(getConf(), getClass());

conf.setJobName("Join 'Department Emp Strength input' with 'Department Name
input'");

Path AInputPath = new Path(args[0]);
Path BInputPath = new Path(args[1]);
Path outputPath = new Path(args[2]);

MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class,
Posts.class);

MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class,
User.class);

FileOutputFormat.setOutputPath(conf, outputPath);

conf.setPartitionerClass(KeyPartitioner.class);

conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);

conf.setMapOutputKeyClass(TextPair.class);

conf.setReducerClass(JoinReducer.class);

conf.setOutputKeyClass(Text.class);
```

```

JobClient.runJob(conf);

return 0;
}

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

int exitCode = ToolRunner.run(new JoinDriver(), args);
System.exit(exitCode);
}
}

// JoinReducer.java
import java.io.IOException;
import java.util.Iterator;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class JoinReducer extends MapReduceBase implements Reducer<TextPair, Text,
Text,
Text> {

@Override
public void reduce (TextPair key, Iterator<Text> values, OutputCollector<Text, Text>
output, Reporter reporter)

throws IOException
{

Text nodeID = new Text(values.next());
while (values.hasNext()) {

Text node = values.next();
Text outValue = new Text(nodeID.toString() + "\t\t" + node.toString());
output.collect(key.getFirst(), outValue);
}
}
}

// User.java
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FSDataOutputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;

```

```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.IntWritable;

public class User extends MapReduceBase implements Mapper<LongWritable, Text,
TextPair,
Text> {

@Override
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reporter reporter)

throws IOException

{

String valueString = value.toString();

String[] SingleNodeData = valueString.split("\t");
output.collect(new TextPair(SingleNodeData[0], "1"), new

Text(SingleNodeData[1]));
}
}

//Posts.java
import java.io.IOException;

import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

public class Posts extends MapReduceBase implements Mapper<LongWritable, Text,
TextPair,
Text> {

@Override
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reporter reporter)
throws IOException
{
String valueString = value.toString();
String[] SingleNodeData = valueString.split("\t");
output.collect(new TextPair(SingleNodeData[3], "0"), new

Text(SingleNodeData[9]));
}
}
```

```
// TextPair.java
import java.io.*;
import org.apache.hadoop.io.*;

public class TextPair implements WritableComparable<TextPair> {

    private Text first;
    private Text second;

    public TextPair() {
        set(new Text(), new Text());
    }

    public TextPair(String first, String second) {
        set(new Text(first), new Text(second));
    }

    public TextPair(Text first, Text second) {
        set(first, second);
    }

    public void set(Text first, Text second) {
        this.first = first;
        this.second = second;
    }

    public Text getFirst() {
        return first;
    }

    public Text getSecond() {
        return second;
    }

    @Override
    public void write(DataOutput out) throws IOException {
        first.write(out);
        second.write(out);
    }

    @Override
    public void readFields(DataInput in) throws IOException {
        first.readFields(in);
        second.readFields(in);
    }

    @Override
    public int hashCode() {
```

```
return first.hashCode() * 163 + second.hashCode();
}

@Override
public boolean equals(Object o) {
if (o instanceof TextPair) {
TextPair tp = (TextPair) o;
return first.equals(tp.first) && second.equals(tp.second);
}
return false;
}

@Override
public String toString() {
return first + "\t" + second;
}

@Override
public int compareTo(TextPair tp) {
int cmp = first.compareTo(tp.first);
if (cmp != 0) {
return cmp;
}
return second.compareTo(tp.second);
}
// ^^ TextPair

// vv TextPairComparator
public static class Comparator extends WritableComparator {

private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

public Comparator() {
super(TextPair.class);
}

@Override
public int compare(byte[] b1, int s1, int l1,
byte[] b2, int s2, int l2) {

try {
int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);
int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);
int cmp = TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);
if (cmp != 0) {
return cmp;
}
return TEXT_COMPARATOR.compare(b1, s1 + firstL1, l1 - firstL1,

```

```

b2, s2 + firstL2, l2 - firstL2);
} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}
}

static {
WritableComparator.define(TextPair.class, new Comparator());
}
public static class FirstComparator extends WritableComparator {

private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

public FirstComparator() {
super(TextPair.class);
}

@Override
public int compare(byte[] b1, int s1, int l1,
byte[] b2, int s2, int l2) {

try {
int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);
int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);
return TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);
} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}
}

@Override
public int compare(WritableComparable a, WritableComparable b) {
if (a instanceof TextPair && b instanceof TextPair) {
return ((TextPair) a).first.compareTo(((TextPair) b).first);
}
return super.compare(a, b);
}
}
}

```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/sampleposts.tsv
"2312" "Feedback on Audio Quality" "cs101 production audio" "100005361" "<p>We are looking for feedback on the audio in our videos. Tell us what you think and try to be as <em>specific</em> as possible.</p>" "question" "\W" "\W" "2012-02-23 00:28:02.321344+00" "2" "" "\W" "201398145" "2014-01-14 17:18:35.613939+00" "2960" "\W" "\W" "524" "f"
"2014856" "" "cs101" "100022094" "<p>I also would like to know the answer to this question. An 'open exam' sounds great, but on the other hand it also seems pretty easy to cheat now: solutions have been posted and anybody only interested in a certificate wouldn't have much of a problem getting the highest distinction. So where is the catch?</p>" "answer" "2014706" "2014706"
"2012-07-01 10:32:36.302782+00" "0" "" "\W" "100022094" "2012-07-01 10:32:36.302782+00" "2020501" "\W" "\W" "0" "f"
"2004004" "" "cs101" "100018705" "<p>But then why even the new variable q? Why not just modify the variable p?</p>" "comment" "2003997" "2003993" "2012-05-03 21:07:52.028935+00"
"2" "" "\W" "100018705" "2012-05-03 21:07:52.028935+00" "2005150" "\W" "\W" "0" "f"
"200005361" "2" "\W" "100018705" "2012-05-03 21:07:52.028935+00" "2005150" "\W" "\W" "0" "f"
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/sampleusers.tsv
"100006402" "18" "0" "0" "0"
"100022094" "6354" "4" "12" "50"
"100018705" "76" "0" "3" "4"
"100005361" "36134" "73" "220" "333"
```

LAB8/Department_Employee_join_example/DeptName.txt

```
C:\hadoop-3.3.0\sbin>hdfs dfs -ls /join8_output/
Found 2 items
-rw-r--r-- 1 Anusree supergroup          0 2021-06-13 12:16 /join8_output/_SUCCESS
-rw-r--r-- 1 Anusree supergroup        71 2021-06-13 12:16 /join8_output/part-00000
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /join8_output/part-00000
"100005361" "2" "36134"
"100018705" "2" "76"
"100022094" "0" "6354"
```