LAB 7 HADOOP MAPREDUCE PROGRAMMING TEMPERATURE:-

Find the average temperature for each year from the NCDC data set.

// AverageDriver.java package temperature;

import org.apache.hadoop.io.*; import org.apache.hadoop.fs.*; import org.apache.hadoop.mapreduce.*; 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);

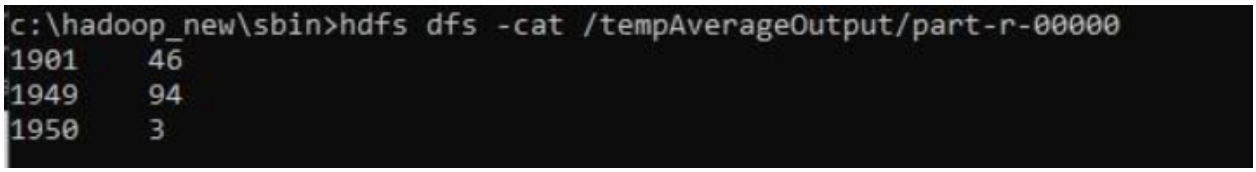
}

}

```
//AverageMapper.java package temperature;

import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.*; import
java.io.IOException;

public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>

{ public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException

{

String line = value.toString(); String year = line.substring(15,19); int temperature; 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 != MISSING && quality.matches("[01459]")) context.write(new
Text(year),new IntWritable(temperature)); }




}

//AverageReducer.java package temperature;


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

public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable>

{

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

{
```

```
int max_temp = 0; int count = 0;

for (IntWritable value : values)

{

max_temp += value.get();

count+=1;

}

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

}
}
```

OUTPUT:-



//TempDriver.java package temperatureMax;

```
import org.apache.hadoop.io.*; import org.apache.hadoop.fs.*; import
org.apache.hadoop.mapreduce.*; import
org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class TempDriver

{ 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(TempDriver.class); job.setJobName(“Max temperature”);

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

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

job.setMapperClass(TempMapper.class); job.setReducerClass(TempReducer.class);

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

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

}

}

TempMapper:-

//TempMapper.java package temperatureMax;

import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.*; import java.io.IOException;

public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>

{

public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException

{

String line = value.toString(); String month = line.substring(19,21); int temperature; 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 != MISSING && quality.matches(“[01459]”)) context.write(new Text(month),new IntWritable(temperature)); }

}

TempReducer:-

//TempReducer.java package temperatureMax;

import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.*; import java.io.IOException;

public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>

{ public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException

{

String line = value.toString(); String month = line.substring(19,21); int temperature; 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 != MISSING &&

quality.matches(“[01459]”)) context.write(new Text(month),new IntWritable(temperature));

}

}

OUTPUT:-

```
c:\hadoop_new\sbin>hdfs dfs -cat /tempMaxOutput/part-r-00000
01      44
02      17
03      111
04      194
05      256
06      278
07      317
08      283
09      211
10      156
11      89
12      117
```