**EXPERIMENT-3**

**AIM:** Develop a MapReduce program to find the maximum temperature in each year.

**PROCEDURE:**

**STEP-1:**

**MaximumTemperature.java**

import java.io.IOException;

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;

public class MaximumTemperature {

// Mapper Class

public static class MaxTemperatureMapper extends Mapper<Object, Text, Text, IntWritable> {

private Text year = new Text();

private IntWritable temperature = new IntWritable();

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

String line = value.toString();

String[] fields = line.split("\\s+"); // Assuming space-separated values

if (fields.length >= 2) {

try {

year.set(fields[0]); // Year as Key

int temp = Integer.parseInt(fields[1]); // Temperature as Value

temperature.set(temp);

context.write(year, temperature);

} catch (NumberFormatException e) {

// Ignore invalid records

}

}

}

}

// Reducer Class

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

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

throws IOException, InterruptedException {

int maxTemp = Integer.MIN\_VALUE;

for (IntWritable val : values) {

maxTemp = Math.max(maxTemp, val.get());

}

context.write(key, new IntWritable(maxTemp));

}

}

// Driver Class

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Maximum Temperature Finder");

job.setJarByClass(MaximumTemperature.class);

job.setMapperClass(MaxTemperatureMapper.class);

job.setReducerClass(MaxTemperatureReducer.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.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);

}

}

**STEP:2** Create a temperature\_data.txt

2021 34

2021 30

2022 36

2022 40

2021 35

2023 33

2023 32

2023 37

**STEP 3:**

**c:\chandrika3>** javac -classpath "C:\hadoop-3.3.6\etc\hadoop;C:\hadoop-3.3.6\share\hadoop\common;C:\hadoop-3.3.6\share\hadoop\common\lib\\*;C:\hadoop-3.3.6\share\hadoop\common\\*;C:\hadoop-3.3.6\share\hadoop\hdfs;C:\hadoop-3.3.6\share\hadoop\hdfs\lib\\*;C:\hadoop-3.3.6\share\hadoop\hdfs\\*;C:\hadoop-3.3.6\share\hadoop\yarn;C:\hadoop-3.3.6\share\hadoop\yarn\lib\\*;C:\hadoop-3.3.6\share\hadoop\yarn\\*;C:\hadoop-3.3.6\share\hadoop\mapreduce\\*" -d . MaximumTemperature.javas

**c:\chandrika3>**jar cf maxTemp.jar MaximumTemperature\*.class

**c:\chandrika3>**hdfs dfs -mkdir -p chandrikasurya

**c:\chandrika3>**hdfs dfs -put temperature\_data.txt /chandrikasurya

**c:\chandrika3>**hadoop jar maxTemp.jar MaximumTemperature /chandrikasurya /suryaoutput

2025-02-06 14:22:09,336 INFO input.FileInputFormat: Total input files to process : 1

2025-02-06 14:22:09,479 INFO mapreduce.JobSubmitter: number of splits:1

2025-02-06 14:22:09,705 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1738831484317\_0001

2025-02-06 14:22:09,705 INFO mapreduce.JobSubmitter: Executing with tokens: []

2025-02-06 14:22:26,044 INFO mapreduce.Job: map 0% reduce 0%

2025-02-06 14:22:34,225 INFO mapreduce.Job: map 100% reduce 0%

2025-02-06 14:22:43,398 INFO mapreduce.Job: map 100% reduce 100%

2025-02-06 14:22:44,430 INFO mapreduce.Job: Job job\_1738831484317\_0001 completed successfully

2025-02-06 14:22:44,585 INFO mapreduce.Job: Counters: 54

File System Counters……………………………………………

**OUTPUT:**

**COMMAND PROMPT**

**c:\chandrika3>**hdfs dfs -cat /suryaoutput/part-r-00000

2021 35

2022 40

2023 37

**HADOOP BROWSER:**





