

```

// Max Temperature Driver program

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;

//import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

public class MaxTemperature {

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

        if (args.length != 2) {

            System.err.println("Usage: MaxTemperature <input path> <output path>");

            System.exit(-1);

        }

        Job job = new Job();

        job.setJarByClass(MaxTemperature.class);

        job.setJobName("Max temperature");

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

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

        job.setMapperClass(MaxTemperatureMapper.class);

        job.setReducerClass(MaxTemperatureReducer.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(IntWritable.class);

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

    }

}

```

```
// Max Temperature Mapper program

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 MaxTemperatureMapper

extends Mapper<LongWritable, Text, Text, IntWritable> {

private static final int MISSING = 9999;

@Override

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

throws IOException, InterruptedException {

String line = value.toString();

String year = line.substring(15, 19);

int airTemperature;

if (line.charAt(87) == '+') { // parseInt doesn't like leading plus signs

airTemperature = Integer.parseInt(line.substring(88, 92));

} else {

airTemperature = Integer.parseInt(line.substring(87, 92));

}

String quality = line.substring(92, 93);

if (airTemperature != MISSING && quality.matches("[01459]")) {

context.write(new Text(year), new IntWritable(airTemperature));

}

}

}
```

```
// Max Temperature Reducer program

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class MaxTemperatureReducer

extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override

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

Context context)

throws IOException, InterruptedException {

int maxVal = Integer.MIN_VALUE;

for (IntWritable value : values) {

maxVal = Math.max(maxVal, value.get());

}

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

}

}
```