

Mapper

```
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 Mapper 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)
        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));
```

```
        }
```

```
    }
```

Driver:

```

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 Driver {

```

```

    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(Driver.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(AWReducer.class);

```

```

        job.setOutputKeyClass(Text.class);

```

```

        job.setOutputValueClass(IntWritable.class);

```

```

    }

```

```

}

```

```

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

```


Reducer

```

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

```

```

public class Reducer extends Reducer<Text, IntWritable,
    Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable>
        values, Reducer<Text, IntWritable, Text, IntWritable> context
        throws IOException, InterruptedException {
        int maxTemp = 0;
        int count = 0;
        for (IntWritable value : values) {
            maxTemp += value.get();
            count++;
        }
        context.write(key, new IntWritable(maxTemp / count));
    }
}

```