Write a MapReduce program to analyze the given natural numbers and generate statistics for the number as Odd or Even and print their sum.

driver.java

package oddeven;

import java.io.\*;

import java.util.\*;

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

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

import org.apache.hadoop.fs.Path;

public class driver

{

public static void main(String args[]) throws IOException

{

JobConf conf=new JobConf(driver.class);

conf.setMapperClass(mapper.class);

conf.setReducerClass(reducer.class);

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

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

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

JobClient.runJob(conf);

}

}

mapper.java

package oddeven;

import java.io.\*;

import java.util.\*;

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

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

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

{

public void map(LongWritable key,Text value,OutputCollector<Text,IntWritable> output,Reporter r) throws IOException

{

String[] line=value.toString().split(" ");

for(String num:line){

int number=Integer.parseInt(num);

if(number%2==0) {

output.collect(new Text("even"),new IntWritable(number));

}

else{

output.collect(new Text("odd"),new IntWritable(number));

}

}

}

}

reducer.java

package oddeven;

import java.io.\*;

import java.util.\*;

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

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

public class reducer extends MapReduceBase implements Reducer<Text,IntWritable,Text,IntWritable>

{

public void reduce(Text key,Iterator<IntWritable> value,OutputCollector<Text,IntWritable> output ,Reporter r) throws IOException

{

int sum=0,count=0;

while(value.hasNext()){

sum+=value.next().get();

count++;

}

output.collect(new Text("Sum of "+key+" Numbers"),new IntWritable(sum));

output.collect(new Text(key+" Number count"),new IntWritable(count));

}

}

Write a MapReduce program to analyze the given Weather Report Data and to generate a report with cities having maximum and minimum temperature for a particular year.

Driver.java

package weather;

import java.util.\*;

import java.io.\*;

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

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

import org.apache.hadoop.fs.Path;

public class driver

{

public static void main(String args[]) throws IOException

{

JobConf conf=new JobConf(driver.class);

conf.setMapperClass(mapper.class);

conf.setReducerClass(reducer.class);

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(DoubleWritable.class);

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

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

JobClient.runJob(conf);

}

}

mapper.java

package weather;

import java.util.\*;

import java.io.\*;

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

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

public class mapper extends MapReduceBase implements Mapper<LongWritable, Text,Text,DoubleWritable>{

public void map(LongWritable key , Text value , OutputCollector<Text,DoubleWritable> output, Reporter r) throws IOException

{

String line=value.toString();

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

Double temp=Double.parseDouble(line.substring(87,92));

output.collect(new Text(year), new DoubleWritable(temp));

}

}

reducer.java

package weather;

import java.util.\*;

import java.io.\*;

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

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

class reducer extends MapReduceBase implements Reducer<Text,DoubleWritable,Text,DoubleWritable> {

public void reduce(Text key, Iterator<DoubleWritable> value, OutputCollector<Text,DoubleWritable> output, Reporter r) throws IOException{

Double max=-9999.0;

Double min=9999.0;

while(value.hasNext()){

Double temp=value.next().get();

max=Math.max(max,temp);

min=Math.min(min,temp);

}

output.collect(new Text("Max temp at "+ key), new DoubleWritable(max));

output.collect(new Text("Min temp at "+ key), new DoubleWritable(min));

}

}