**JAVA**

**$ javac --version**

**$ sudo apt install default-jdk**

**$ sudo apt install default-jre**

**$ javac --version**

**$ cd <1MS23SCS/SCN\*\*>**

**Hadoop**

**$ tar xvzf hadoop-3.4.0.tar.gz**

**$ cd hadoop-3.4.0/**

**$ gedit bash.sh**

**export JAVA\_HOME=$(readlink -f $(which javac) | awk 'BEGIN {FS="/bin"} {print $1}')**

**export PATH=$(echo $PATH):$(pwd)/bin**

**export CLASSPATH=$(hadoop classpath)**

**$ source bash.sh**

**$ hadoop**

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

**mkdir weather**

**cd weather**

**gedit driver.java**

**gedit mapper.java**

**gedit reducer.java**

**gedit input.txt**

**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.jav**

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));

}

}

**input.txt**

0067011990999991950051507004+68750+023550FM-12+038299999V0203301N00671220001CN9999999N9+00001+99999999999

0043011990999991950051512004+68750+023550FM-12+038299999V0203201N00671220001CN9999999N9+00221+99999999999

0043011990999991950051518004+68750+023550FM-12+038299999V0203201N00261220001CN9999999N9-00111+99999999999

0043012650999991949032412004+62300+010750FM-12+048599999V0202701N00461220001CN0500001N9+01111+99999999999

0043012650999991949032418004+62300+010750FM-12+048599999V0202701N00461220001CN0500001N9+00781+99999999999

javac -d . \*.java

echo Main-Class: weather.driver > Manifest.txt

jar cfm weather.jar Manifest.txt weather/\*.class

hadoop jar weather.jar input.txt output

cat output/\*

**Write a MapReduce program using Java, to analyze the given Earthquake Data and generate statistics with region and magnitude/ region and depth/ region and latitude/ region and longitude**

**mkdir earthquake**

**cd earthquake**

**gedit driver.java**

**gedit mapper.java**

**gedit reducer.java**

**driver.java**

package earthquake;

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 earthquake;

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().split(",");

Double longi=Double.parseDouble(line[7]);

output.collect(new Text(line[11]), new DoubleWritable(longi));

}

}

**reducer.java**

package earthquake;

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;

while(value.hasNext())

{

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

max=Math.max(max,temp);

}

output.collect(new Text(key), new DoubleWritable(max));

}

}

javac -d . \*.java

echo Main-Class: earthquake.driver > Manifest.txt

jar cfm earthquake.jar Manifest.txt earthquake/\*.class

hadoop jar earthquake.jar input.csv output

cat output/\*

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

**mkdir oddeven**

**cd oddeven**

**gedit driver.java**

**gedit mapper.java**

**gedit reducer.java**

**gedit input.txt**

**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));

}

}

**input.txt**

1 2 3 4 5 6 7 8 9 10

javac -d . \*.java

echo Main-Class: oddeven.driver > Manifest.txt

jar cfm oddeven.jar Manifest.txt oddeven/\*.class

echo 1 2 3 4 5 6 7 8 9 10 > input.txt

hadoop jar oddeven.jar input.txt output

cat output/\*

**Write a MapReduce program using Java, to analyze the given Insurance Data and generate a statistics report with the construction building name and the count of building/ county name and its frequency.**

**mkdir insurance**

**cd insurance**

**gedit driver.java**

**gedit mapper.java**

**gedit reducer.java**

**driver.java**

package insurance;

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 insurance;

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(",");

output.collect(new Text(line[2]),new IntWritable(1));

}

}

**reducer.java**

package insurance;

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;

while(value.hasNext())

{

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

}

output.collect(key,new IntWritable(sum));

}

}

javac -d . \*.java

echo Main-Class: insurance.driver > Manifest.txt

jar cfm insurance.jar Manifest.txt insurance/\*.class

hadoop jar insurance.jar input-insurance.csv output

cat output/\*

**Write a MapReduce program using Java, to analyze the given employee record data and generate a statistics report with the total number of Female and Male Employees and their average salary.**

**mkdir employee**

**cd employee**

**gedit driver.java**

**gedit mapper.java**

**gedit reducer.java**

**gedit input.csv**

**driver.java**

package employee;

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(DoubleWritable.class);

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

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

JobClient.runJob(conf);

}

}

**mapper.java**

package employee;

import java.io.\*;

import java.util.\*;

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

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

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().split("\\t");

Double salary=Double.parseDouble(line[8]);

output.collect(new Text(line[3]), new DoubleWritable(salary));

}

}

**reducer.java**

package employee;

import java.io.\*;

import java.util.\*;

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

{

int count=0;

Double sum=0.0;

while(value.hasNext()){

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

count+=1;

}

output.collect(new Text(key+" Average"), new DoubleWritable(sum/count));

output.collect(new Text(key+" Count"), new DoubleWritable(count));

}

}

javac -d . \*.java

echo Main-Class: employee.driver > Manifest.txt

jar cfm employee.jar Manifest.txt employee/\*.class

hadoop jar employee.jar input.csv output

cat output/\*

**Write a MapReduce program using Java, to analyze the given Sales Records over a period of time and generate data about the country’s total sales, and the total number of the products. / Country’s total sales and the frequency of the payment mode.**

**mkdir sales**

**cd sales**

**gedit driver.java**

**gedit mapper.java**

**gedit reducer.java**

**driver.java**

package sales;

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 sales;

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(",");

int price=Integer.parseInt(line[2]);

String cardtype=line[3];

String Country=line[7];

output.collect(new Text("Country "+Country),new IntWritable(price));

output.collect(new Text("CardType "+cardtype),new IntWritable(1));

}

}

**reducer.java**

package sales;

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;

while(value.hasNext())

{

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

}

output.collect(new Text(key),new IntWritable(sum));

}

}

javac -d . \*.java

echo Main-Class: sales.driver > Manifest.txt

jar cfm sales.jar Manifest.txt sales/\*.class

hadoop jar sales.jar sales.txt output

cat output/\*