**AIM:** To write a map reduce program to find minimum number.

**PROGRAM:**

**/\*MAPPER\*/**

import java.util.\*;

import java.io.\*;

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

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

class MMapper extends Mapper<LongWritable,Text,Text,LongWritable>

{

int min=0,i,j,count=0;

public void map(LongWritable key, Text value, Context contex) throws IOException, InterruptedException

{

String line=value.toString();

String[] num\_arr=line.split(" ");

if(count==0)

{

min=Integer.parseInt(num\_arr[0]);

count=1;

}

else

{

for(i=0;i<num\_arr.length;i++)

{

j=Integer.parseInt(num\_arr[i]);

if(j<min)

min=j;

}

}

contex.write(new Text("X"),new LongWritable(min));

}

}

**/\*REDUCER\*/**

import java.util.\*;

import java.io.\*;

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

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

public class MReducer extends Reducer<Text, LongWritable, Text, LongWritable>

{

int i,count=0;

long min=0;

public void reduce(Text key, Iterable<LongWritable> values, Context context)throws IOException, InterruptedException

{

for(LongWritable val:values)

{

if(count==0)

{

min=val.get();

count=1;

}

else

{

if(val.get()<min)

min=val.get();

}

}

context.write(new Text("minimum"),new LongWritable(min));

}

}

**/\*DRIVER\*/**

import java.util.\*;

import java.io.\*;

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

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

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

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

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class MDriver

{

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

{

Configuration conf = new Configuration();

Job job = Job.getInstance(conf,"xyz");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(LongWritable.class);

job.setMapperClass(MMapper.class);

job.setReducerClass(MReducer.class);

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

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

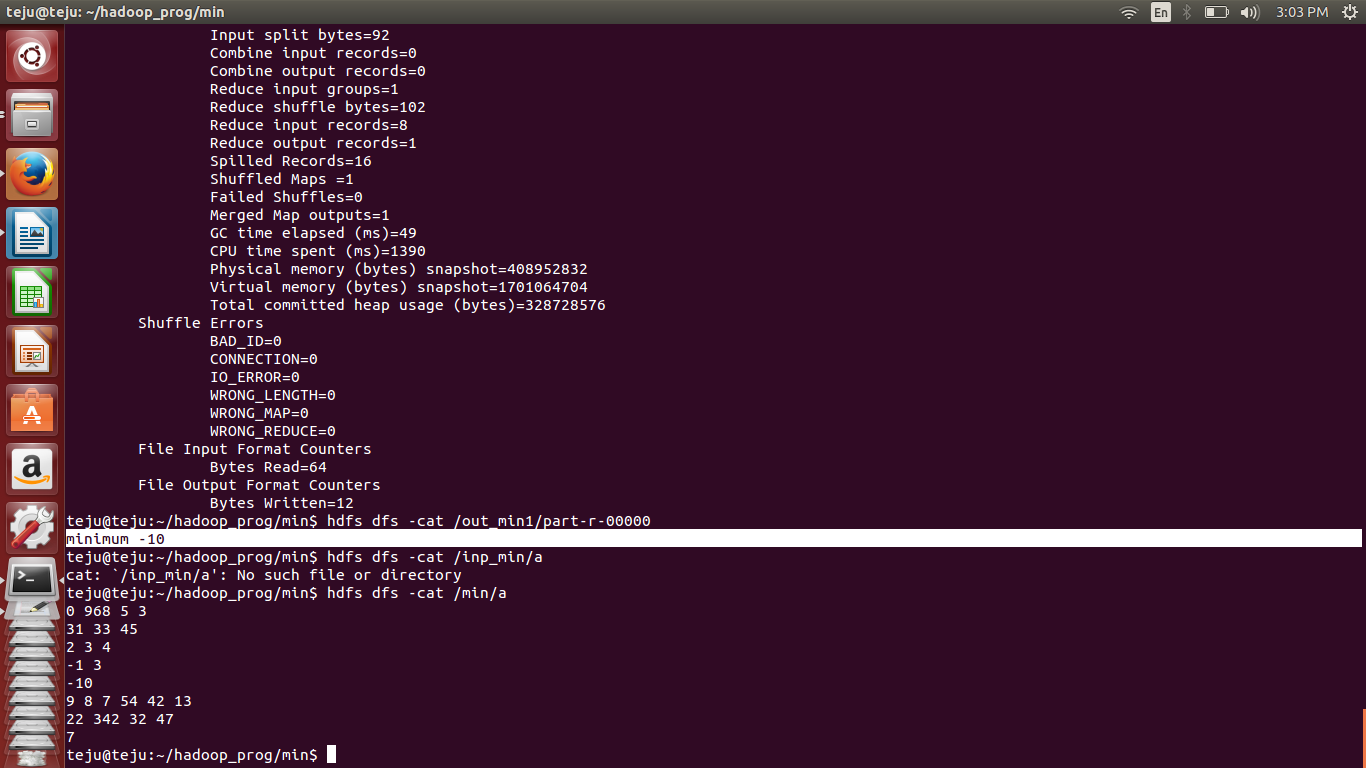
job.setJarByClass(MDriver.class);

job.waitForCompletion(true);

}

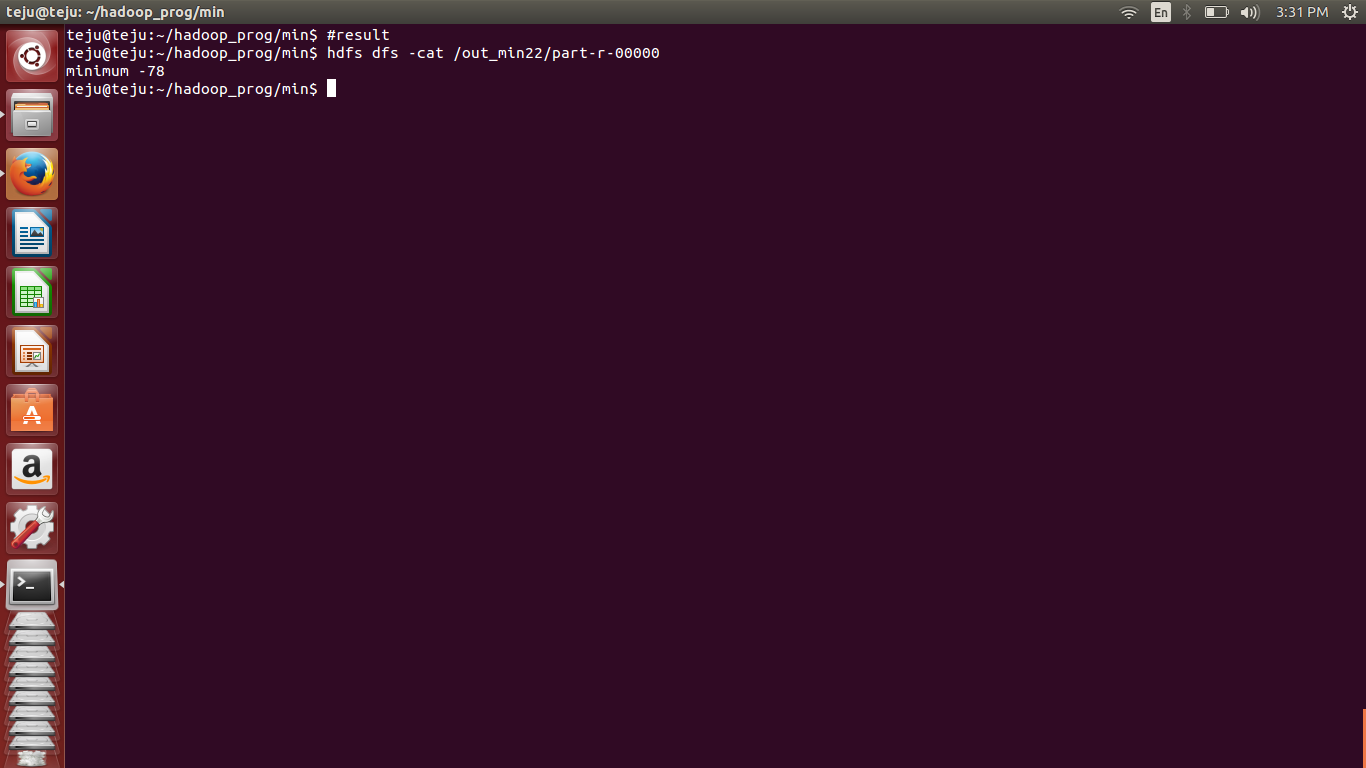
}

**OUTPUT 1:**



**OUTPUT 2:**

**Input1.txt generated by a c program is given as input.**

****