import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

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.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

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

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

public class sort {

public static class TokenizerMapper1

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

private Text text = new Text();

public void map(Object key, Text value, Context context

) throws IOException, InterruptedException {

String line = value.toString();

String runs=line.replaceAll("[^0-9]","");

String tuple=line.replaceAll("[^ A-Za-z,]","");

int run=Integer.parseInt(runs);

IntWritable word = new IntWritable(run);

word.set(tuple);

context.write(text,word);

}

}

public static class IntSumReducer

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

private Text res= new Text();

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

Context context

) throws IOException, InterruptedException {

for (Text val : values) {

//result = val;

res.set(val);

context.write(key,res);

}

}

}

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

Configuration config = new Configuration();

Job job1 = Job.getInstance(config, "word count");

job.setJarByClass(sort.class);

job.setMapperClass(TokenizerMapper1.class);

job.setCombinerClass(IntSumReducer.class);

job.setReducerClass(IntSumReducer.class);

job.setOutputKeyClass(IntWritable.class);

job.setOutputValueClass(Text.class);

FileInputFormat.addInputPath(job1, new Path(args[0]));

FileOutputFormat.setOutputPath(job1, new Path(args[1]));

System.exit(job1.waitForCompletion(true) ? 0 : 1);

}

}