MapReduce Program:

import java.io.IOException;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

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 OlympicsOne {

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

Configuration conf = new Configuration();

if (args.length != 2) {

System.err.println("Usage: Olympics Champions <in> <out>");

System.exit(2);

}

Job job = new Job();

job.setJarByClass(OlympicsOne.class);

job.setJobName("Olympics Champions");

job.setMapperClass(TokenizerMapper.class);

job.setCombinerClass(SumReducer.class);

job.setReducerClass(SumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

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

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

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

}

public static class SumReducer extends

Reducer<Text, IntWritable, Text, IntWritable> {

private IntWritable result = new IntWritable();

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

Reducer<Text, IntWritable, Text, IntWritable>.Context context)

throws IOException, InterruptedException {

int sum = 0;

for (IntWritable val : values) {

sum += val.get();

}

this.result.set(sum);

context.write(key, this.result);

}

}

public static class TokenizerMapper extends

Mapper<Object, Text, Text, IntWritable> {

public void map(Object key, Text value,

Mapper<Object, Text, Text, IntWritable>.Context context)

throws IOException, InterruptedException {

String[] parts = value.toString().split("\t");

if (parts[OlympConst.Sport].equalsIgnoreCase("Swimming"))

{

String StrCountry=parts[OlympConst.Country];

int totalMedals=Integer.parseInt(parts[OlympConst.TotalMedals]);

context.write(new Text(StrCountry), new IntWritable(totalMedals));

}

}

}

}

class OlympConst {

public static int Athlete = 0;

public static int Age = 1;

public static int Country = 2;

public static int Year = 3;

public static int ClosingDate = 4;

public static int Sport = 5;

public static int GoldMedals = 6;

public static int SilverMedals = 7;

public static int BronzeMedals = 8;

public static int TotalMedals = 9;

}

Output

