

## Assignment 1

### Q1

A) Mapper Input: <sup>Line #</sup> IntWritable & <sup>numbers</sup> Text, Output: IntWritable, IntWritable (key=Line#)  
Reducer Input: IntWritable, IntWritable, Output: IntWritable, IntWritable

### B) Map Method for Program

... extends Mapper<IntWritable, Text, IntWritable, IntWritable>

... Map(...) & ...

Split all numbers in line, Let n = number

for each number: context.write(key, n); // key = line number

### C) Reduce Method for Program

... extends Reducer<IntWritable, IntWritable, IntWritable, IntWritable>

... reduce(IntWritable lineNumber, Iterable<IntWritable> numbers, context context)...

int count = 0;

for(IntWritable number: numbers)

if(number % 3 == 0) count++;

Context.write(lineNumber, new IntWritable(count));

### D) Combiner

I would say a combiner would not work/make sense for this program

because the map method is splitting each line into a key value pair of

line number and the value, so there isn't anything to actually combine.

Maybe if the mapper returned a list of numbers, instead, numbers not

divisible could be skipped.

In short, there is nothing to optimize



Q2 Note: Exact Similar as provided in slides

A) Mapper Input: LongWritable, Text Output: Text, IntWritable  
Reduce Input: Text Output: IntWritable

B) Map method

TempMapper extends Mapper <LongWritable, Text, Text, IntWritable> ...

map(LongWritable key, Text value, Context context) ...

String valAsString = value.toString().trim();

... Split " " and error check ...

Context.write(new Text(tokens[0]), new IntWritable(... tokens[1]));

C) Reduce method // calculate highest temp for each day

TempReducer extends Reducer <Text, IntWritable, Text, IntWritable> ...

reduce(Text date, Iterable<IntWritable> temps, Context context) ...

// return error if no temps, (edge case)

int highest = Math.MIN;

for (IntWritable temperature : temperatures)

highest = Math.max(highest, temperature);

Context.write(date, new IntWritable(highest));

D) Combiner

Yes, but it would not make sense to create one? I am having trouble

with understanding the purpose of this combiner. It seems the code

would be the same, but there would be a class object to hold the value?