**W1D2**

**Question 1**

**This part of the assignment starts where W1D1 left off. Yesterday you created the Mapper Class. Today you are going to create the Reducer Class.**

Write a Java program to accomplish the following tasks.

1. Create a GroupByPair with two data members. First member is Key. Second member is a list of Values.
2. Iterating through the sorted list of Key-Value Pair, create a new list of GroupByPair so that all values associated with the same key are grouped together and kept in one GroupByPair.
3. Print the GroupByPair list
4. Iterate through GroupBy list “sum” the values and print out (key, sum) pairs.
5. **Include additional print statements so that the output produced matches the sample output file attached.**

Note: 1. Use the test data from W1D1.

Note: 2. Observe that you just simulated word count MapReduce with one Mapper and one Reducer.

**Question 2**

Define an array to be nice if **whenever it contains 4, it also contains 3.**See examples below:

[7, 6, 2, 3, 1]   //nice

[7, 6, 2, 4, 1]   // NOT nice. 4 is there, but 3 is not there

[3, 6, 2, 3, 4]   //nice

[3, 4, 2, 3, 4, 7, 4]   //nice

[1, 6, 2, 8, 2, 9]   //nice

Explain how map and fold (of functional programming) can be used to determine whether or not **an array is nice**.  In other words, write pseudo-code for functions f and g, specify initial value and interpretation for final value.

Answers:

*map is used to return x if x = 3 or x = 4 otherwise return 0.*

*reduce is used to return x if x = 7 otherwise return x +y;*

Part 1: pseudo code for f.

int f(x) {

if (x == 3 || x == 4)

return x;

else

return 0;

}

Part 2: pseudo code for g.

int g(x, y) {

if (x == 7)

return x;

else

return x + y;

}

Part 3: Specification of initial value.

Initial value = 0

Part 4: Interpretation of final value.

+ integer: 0, 7, 3 => array is nice, 4 not nice