Big Data Lab Answer Sheet.

Please complete this answer sheet and turn it in at the beginning of class on the due date posted in LEARN.

Part I

|  |  |
| --- | --- |
| **Part 1:** | Answer |
| **1**  (4 pts) | json organized the data by section name and value with space. It’s more readable than XML. |
| **2**  (4 pts) | XML print out the variable name with <> in front and end. The value is between them without space, which is difficult to see. |
| **3**  (4 pts) | Alabama: 2552  Alaska: 388  I think it because of Alabama has larger population than Alaska. |
| **4**  (6 pts) | I use search function from the browser to find the sucide data set in particular time.  I prefer XML. It clearly label the element name so when I can just read the element name within the <> to know what that data represent. |
| **Part 2:** | Answer |
| **5**  (2 pts) | When I clicked the single step, it first printed the calling mapper with data that contained a complete sentence. It only prints one sentence each time when I clicked the single step button. Then, the mapper classifies each word and store in key value. |
| **6**  (2 pts) | A word from a sentence. |
| **7**  (2 pts) | The MapReduce sum word counts from each line to get total for each word. When a paragraph contains duplicate word, it would store in the same row, but different value. |
| **8**  (2 pts) | The mapper only storing single letter from the sentences. |
| **9**  (2 pts) | The mapper function counts the number of each word in line of input and Emit function emit each word along with its count. |
| **10**  (2 pts) | It counts all the word from mapper function. If it finds duplicate value or word, the counter would increment. |
| **11**  (5 pts) | The code split the sentence into single word and assign value 1 into it. |
| **12**  (15 pts) | Mapper:  Dave 81  Bob 66  Alice 95  Eve 86  Carol 57  Carol 74  Dave 74  Bob 82  Eve 55  Alice 87  Alice 63  Dave 79  Bob 56  Eve 50  Carol 70  Reducer:  Alice 81.6666666667  Bob 68.0  Carol 67.0  Dave 78.0  Eve 63.6666666667 |
| **13**  (20 pts) | def mapper(key, value):      grade\_map = eval(key)      for student in grade\_map:          grade = grade\_map[student]          Wmr.emit(student, grade)  Reducer:  def reducer(key, values):      sum = 0      for value in values:          sum = sum + int(value)     if key=="Enrollments" or key =="Dropouts":          Wmr.emit(key, sum) |