**Instruction:**

* Use Object Oriented Python or NodeJS to solve the following problem.
* Develop your own algorithm and do not use external modules in Python/NodeJS.
* Make sure code is legible/maintainable and includes appropriate documentation, error handling, etc.
* Write code that is typical of something our Data Application team would be proud to have in production.
* You will need to independently overcome any challenges you face in delivery, be prepared to answer any challenge questions for your code design.
* List any assumptions if applicable.

**Problem to solve:**

* Read strings in the file inputTestCases.txt.
* Add more test cases to the file if applicable.
* Starting from the outermost to the innermost, each curly bracket represents one layer. Add no hyphen to the output of the elements in the outermost curly bracket. Add one hyphen to the output of the elements in the 2nd outermost curly bracket. Add two hyphens to the output of the elements in the 3rd outermost curly bracket, and so forth.
* Convert  
  {id,operator,collectedWhen{id,dateTag,collectedWhen{id}, uploadedWhen},data}

to the following output – unsorted:

id

operator

collectedWhen

- id

- dateTag

- collectedWhen

-- id

- uploadedWhen

data

and output in alphabetical sorted order:

collectedWhen

- collectedWhen

data

- dateTag

id

- id

-- id

operator

- uploadedWhen

**Deliverables:**

* Source code.
* README file with instruction how to run.