Engineering 13300 HW 8 PY 4
Python 4: Python File I/O – Fa21

Individual Tasks

Guidelines for Tasks 6–7:

- Tasks 6–7 are individual tasks. You may seek help from classmates, the instructional team or others but the work you submit should be your own. If you collaborate with others and use information developed together or by someone else, ALWAYS document and reference that material.
- Each individual is responsible to submit their own assignment to Gradescope

Task 6 (of 7) [Individual]

Learning Objectives: Utilize file I/O functions in Python; Construct files using formatted output using Python; Read in input data from a user at the keyboard in Python.

Background:

For many reasons, engineers write formal reports in the real world. In a documentation of experiments or designs, if each line matches a number, like line numbering in Python programs, reviewing and discussing these reports with co-workers or clients can be easier.

Task:

Develop a flow diagram and write a program in Python that will ask the user for an input file name, and then for an output file name. For both files, user must include a proper extension such as ".txt" for text files. The input file contains a guideline with an unknown number of lines, each line representing a step that can be performed for all workplaces to take to reduce risk of exposure to coronavirus. Your job is to create a new file with the given output file name, and then write the file such that each line of the input file is written into it with step numbers appended to the beginning of the line. Note that each line may contain several sentences.

Example:

Input:

```
Enter the filename of the input file: Py4_Task6_input.txt
Enter the filename of the output file: Py4_Task6_output.txt
```

Contents of `Py4_Task6_input.txt`:

```
Encourage workers to stay home if sick.
Encourage respiratory etiquette, including covering coughs and sneezes.
Provide a place to wash hands or alcohol-based hand rubs containing at
least 60% alcohol.
Limit worksite access to only essential workers, if possible.
Establish flexible worksites (e.g., telecommuting) and flexible work
hours (e.g., staggered shifts), if feasible.
Discourage workers from using other workers' phones, desks, or other
work tools and equipment.
Regularly clean and disinfect surfaces, equipment, and other elements of
the work environment.
```

```
Use Environmental Protection Agency (EPA)-approved cleaning chemicals
with label claims against the coronavirus.
Follow the manufacturer’s instructions for use of all cleaning and
disinfection products.
Encourage workers to report any safety and health concerns.
```

Sample output of `Py4_Task6_output.txt`:

```
Step 1: Encourage workers to stay home if sick.
Step 2: Encourage respiratory etiquette, including covering coughs and
sneezes.
Step 3: Provide a place to wash hands or alcohol-based hand rubs
containing at least 60% alcohol.
Step 4: Limit worksite access to only essential workers, if possible.
Step 5: Establish flexible worksites (e.g., telecommuting) and flexible
work hours (e.g., staggered shifts), if feasible.
Step 6: Discourage workers from using other workers' phones, desks, or
other work tools and equipment.
Step 7: Regularly clean and disinfect surfaces, equipment, and other
elements of the work environment.
Step 8: Use Environmental Protection Agency (EPA)-approved cleaning
chemicals with label claims against the coronavirus.
Step 9: Follow the manufacturer's instructions for use of all cleaning
and disinfection products.
Step 10: Encourage workers to report any safety and health concerns.
```

Hint: Lookup `enumerate` function

Task 6 Files:

1. `Py4_Ind_`*`username`*`.pdf`
2. `Py4_Task6_`*`username`*`.py`
3. `Py4_Task6_output.txt`

Task 7 (of 7) [Individual] - Autograded

Learning Objectives: Utilize file I/O functions in Python.

Background:

Repeating the same word within a short chunk may lead to a less engaging read. To avoid overusing words while writing paper, calculating occurrences of the same word is a good way to check.

Task:

Write a flow chart and a python program that will ask the user for a search word and the name of text file. Assume that the input file is named `Py4_Task7_text.txt` and that it will contain a large amount of words, likely in the form of an excerpt from a text. The program is supposed to search the input file for the search word and compute the percentage of times the word occurs in the document, e.g. if there are 200 words and the word occurs 3 times, the percentage would be 1.50%. Note that this percentage should be truncated to the hundredth place.

It is important to note that capitalization does not matter for this search, e.g. “Butterfly” is equivalent to “butterfly”. Also, words may be followed by characters including but not limited to commas, periods, and semi-colons, e.g. “butterfly,” will count as “butterfly”. The search word will always be a single word, which means occurrences of the search word should not be counted if they appear inside another word. For example, “butterfly” should not be counted for “butter” or “fly”.

Hint:

The following functions may be useful: `lower()`, `isalnum()`, `isalpha()`

Example:

Input 1:

```
Enter the filename of the text file: Py4_Task7_text.txt
Enter the search word: GARDEN
```

Output 1:

```
The word “GARDEN” appears 1 out of 304 words or 0.33% of the time.
```

Input 2:

```
Enter the filename of the text file: Py4_Task7_text.txt
Enter the search word: free
```

Output 2:

```
The word “free” appears 2 out of 304 words or 0.66% of the time.
```

Contents of Py4_Task7_text.txt

Ode to a butterfly
Thou spark of life that wavest wings of gold,
Thou songless wanderer mid the songful birds,
With Nature's secrets in thy tints unrolled
Through gorgeous cipher, past the reach of words,
Yet dear to every child
In glad pursuit beguiled,
Living his unspoiled days mid flowers and flocks and herds!
Thou winged blossom, liberated thing,
What secret tie binds thee to other flowers,
Still held within the garden's fostering?
Will they too soar with the completed hours,
Take flight, and be like thee
Irrevocably free,
Hovering at will o'er their parental bowers?
Or is thy lustre drawn from heavenly hues,
A sumptuous drifting fragment of the sky,
Caught when the sunset its last glance imbues
With sudden splendor, and the tree-tops high
Grasp that swift blazonry,
Then lend those tints to thee,
On thee to float a few short hours, and die?
Birds have their nests; they rear their eager young,
And flit on errands all the livelong day;
Each fieldmouse keeps the homestead whence it sprung;
But thou art Nature's freeman,—free to stray Unfettered through the wood,
Seeking thine airy food,
The sweetness spiced on every blossomed spray.
The garden one wide banquet spreads for thee,
O daintiest reveller of the joyous earth!
One drop of honey gives satiety;
A second draught would drug thee past all mirth.
Thy feast no orgy shows;
Thy calm eyes never close,
Thou soberest sprite to which the sun gives birth.
And yet the soul of man upon thy wings
Forever soars in aspiration; thou
His emblem of the new career that springs
When death's arrest bids all his spirit bow.
He seeks his hope in thee
Of immortality.
Symbol of life, me with such faith endow!
By Thomas Wentworth Higginson. Edmund Clarence Stedman, ed. (1833–1908). An American Anthology, 1787–1900. 1900.

Note: highlighting is used to clarify the example – there is no highlighting in the text file itself. The word "garden's" is a possessive and therefore does not count.

Task 7 Files:

1. Py4_Ind_*username*.pdf
2. Py4_Task7_*username*.py

Summary of Files to submit

1. Py4_Team_*teamnumber*.pdf
2. Py4_Task1_teamnumber.py
3. Py4_Task1_output.txt
4. Py4_Task2_*teamnumber*.py
5. Py4_Task2_input.txt
6. Py4_Task2_output.txt
7. Py4_Task3_*teamnumber*.py
8. Py4_Task3_input.txt
9. Py4_Task3_output.txt
10. Py4_Task4_*teamnumber*.py
11. Py4_Task4_input.txt
12. Py4_Task4_output.txt
13. Py4_Task5_*teamnumber*.py
14. Py4_Ind_*username*.pdf
15. Py4_Task6_*username*.py
16. Py4_Task6_output.txt
17. Py4_Task7_*username*.py