Instructions

This assignment is composed of 10 “questions”.

There are two types of questions:

- Type 1 questions for which you have to write a script.
- Type 2 questions for which you have to explain or modify a given script.

There are 5 type 1 questions and 5 type 2 questions.

You are expected to use the default packages coming with python 3. It should not be necessary to install an additional package to run your script.

Criteria:

Type 1:

- Questions will describe what the script should do
- Questions will have specific requirements on how to write the script.

You should write a script that behaves as expected and fulfills the requirements.
In addition, you should comment your script to explain your design choice.
(Point allocation: 50% behavior 30% requirements 20% comments)

A requirement for all questions is to add safeguards to avoid as much as possible that an error occurs at runtime.

Type 2:

- The script modification parts will be evaluated like type 1
  (Point allocation: 50% behavior 30% requirements 20% comments)

- The questions about script understanding will be evaluated according to how well you understood/explained the problem.

All the questions (Type 1 and Type 2) are graded on 10 points (10x10 = 100 points in all).

How to submit your work:

For each question, you should submit a single file and the name of the file to submit is indicated in the question title.

Part 1: 5 files

- Question #1: Q1.py
- Question #2: Q2.py
- Question #3: Q3.py
- Question #4: Q4.py
- Question #5: Q5.py

In this part, you re-use the script you wrote as a base.
Q2.py is built starting from Q1.py
Q3.py is built starting from Q2.py
⋮
Q5.py is built starting from Q4.py

Part 2: 5 files

- Question #6: Q6.pdf
- Question #7: Q7.pdf

create pdf files to write your explanations
(you can copy paste the part of code you explain)

- Question #8: Q8.py
- Question #9: Q9.py
- Question #10: Q10.py

copy and modify the provided moonscape.py (provided in sources.zip)
to add functionalities

You should group all your files together in a zip archive and submit it through PandA.
The name should be: <student_id>_assignment_2.zip
If your id is 123456, the zip file will be 123456_assignment_2.zip

Part I: 5 type 1 questions

Question #1

Write a script that:

1. reads the file `book_chapter.txt` (provided in sources.zip)
2. creates a list of words
   When creating the list:
   - replace upper case letter by smaller case
   - remove the empty word from the list
3. displays the total number of words (not in a GUI)

Question #2

Building on top of the previous script.

1. Create a dictionary `words_dict` that has the words as keys and the counts of the words as values.

2. Add a function that filters the dictionary to keep only the words that have a count higher than a given threshold.

   That function takes the full dictionary of words and the threshold as arguments:
   ```
   def filter_dict(words_dict, threshold):
   ```

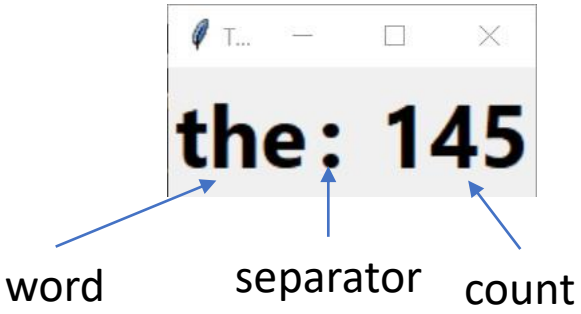
   That functions returns the filtered dictionary of words:
   ```
   return filtered_words_dict
   ```

3. After filtering `words_dict` using a threshold of 20, display all the words and their count in the filtered dictionary (not in a GUI)

Question #3

Building on top of the previous script.

Add a GUI that displays a word along its count as follows:



(You can choose to display any of the words)

- Your GUI should be an object that is a subclass of Frame
- You have to use `StringVar` and `IntVar` for the elements to display
- This object should get the dictionary of word as one of its arguments

Question #4

Building on top of the previous script.

Add **binding** so that the left and right arrow keys can be used to change the displayed word and count:

- pressing right arrow on the keyboard displays the next word and count
- pressing left arrow on the keyboard displays the previous word and count

Hint:
You can create a list of keys of the words dictionary by using the statement:

```
list_of_keys = list(word_dict.keys())
```

(this works for values too)
And use and a single index to keep the position in these lists.

Key bindings:

Next word

Previous word

(If you have some trouble on a laptop with arrow keys you can use other keys)

Question #5

Building on top of the previous script.

Modify the script such that the GUI displays the words and counts from a dictionary that was filtered using a threshold.

Add binding so that the up and down arrow keys can be used to change the value of the threshold and update the display:

- pressing up arrow increases the threshold by one unit
- pressing down arrow decreases the threshold by one unit

(Do not forget to update the filtered dictionary and the display after pressing the arrows)

Key bindings:

Next word

Previous word

increases threshold

decreases threshold

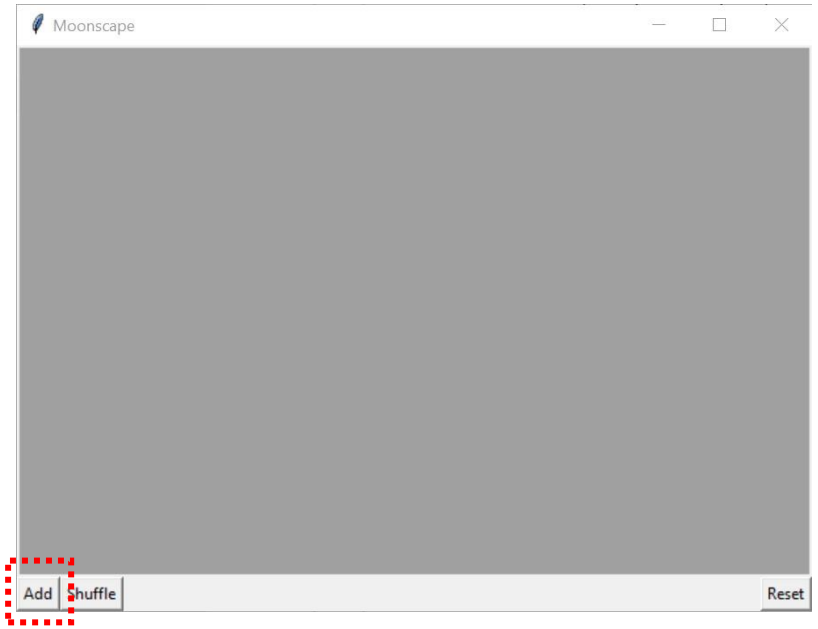
(If you have some trouble on a laptop with arrow keys you can use other keys)

Part II: 5 type 2 questions

Question #6

Explain what is happening when pressing the “Add” button in the GUI.

- Follow the code to explain the important steps.
- Show where things are taking place in the code.

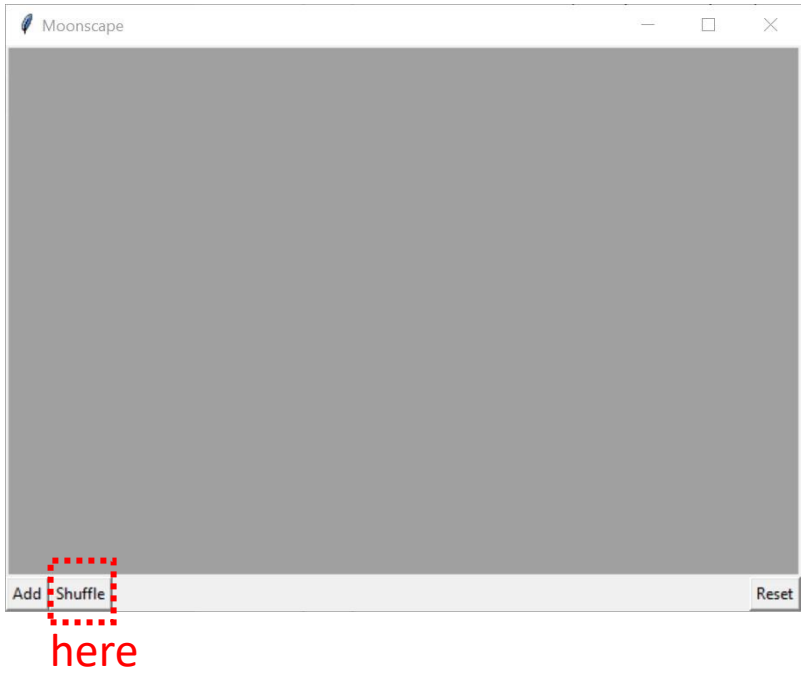


here

Question #7

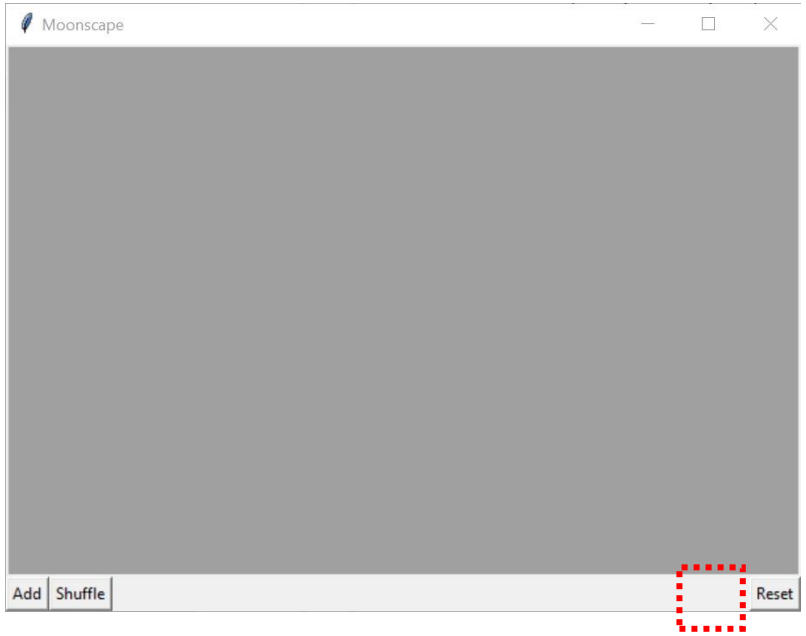
Explain what is happening when pressing the “Shuffle” button in the GUI.

- Follow the code to explain the important steps.
- Show where things are taking place in the code.



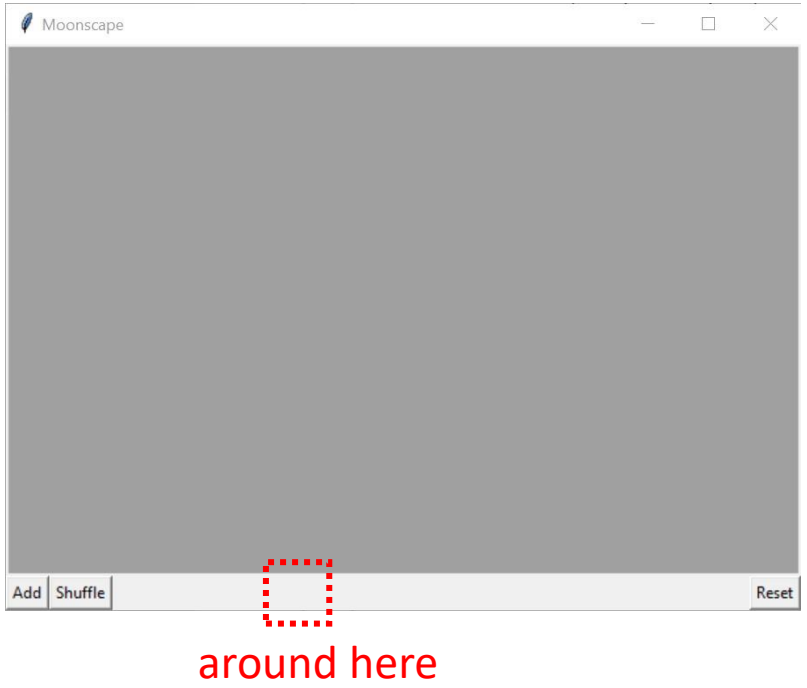
Question #8

Add a button that only erases the objects that are not gray.
(hint: you can add tags)



Question #9

Add a button for bringing the objects that are not gray in front of the objects that are gray (hint: you can add tags)



Question #10

Modify the script such that when pushing the “Add” button, in addition to the two existing types of object, a third type of object is created: green squares.

The green squares are such that:

- The color should be a random green
- The scale should be 0.15
- There should be around 3% of such objects (the split is 92%, 5% and 3%)