Mastery Check 3 Assignment Overview:

Objective: You are creating a program that analyzes a given string and performs various operations on it using user-defined functions.

Goal: Your program should incorporate functions to analyze text input provided by the user. The program will incorporate user-defined functions and string manipulation techniques to perform various tasks such as counting characters, words, vowels, and sentences, determining the frequency of each letter in the text, and so much more! There are over a dozen options for you to pick from.

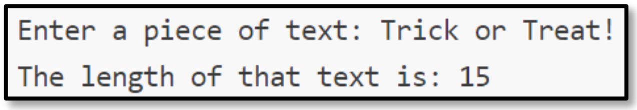
Remember to put all your functions definitions at the top of your program and your function calls in the main of the project. Also, do not use global variables. We want a value to be sent to a function parameter and for the function to return a value.

Tasks:

Part 1: Input Text: Prompt the user to enter a piece of text in the main of the project. (6 points)

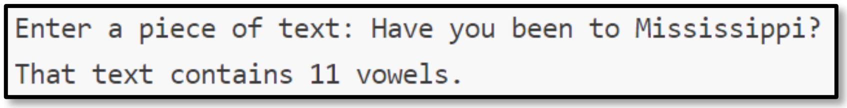
Part 2: Processing: **Select 6 user-defined functions** from the list of below and write them into your program. Be sure to leave comments. (60 pts, 10 points for each function)

- Option 1: String Length
  - Write a function called **calculate_length(text)** that takes a string as input and returns its length.



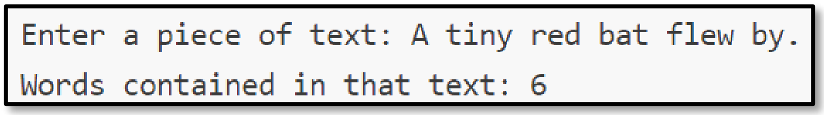
```
Enter a piece of text: Trick or Treat!
The length of that text is: 15
```

- Option 2: Number of Vowels
  - Write a function called **count_vowels(text)** that takes a string as input and returns the count of vowels in the string. For this exercise, vowels are defined as a,e,i,o,u



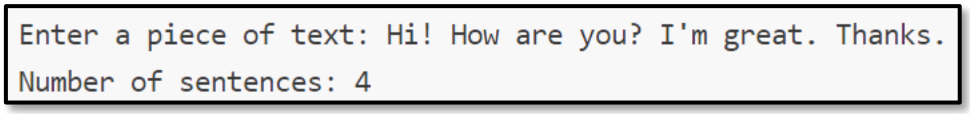
```
Enter a piece of text: Have you been to Mississippi?
That text contains 11 vowels.
```

- Option 3: Word Count
  - Write a function called **count_words(text)** that takes the input text as a parameter and returns the total number of words in the text. Words are separated by spaces.



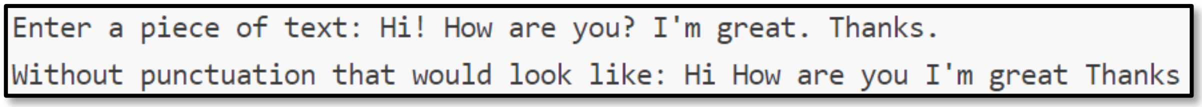
```
Enter a piece of text: A tiny red bat flew by.
Words contained in that text: 6
```

- Option 4: Sentence Count
  - Write a function **count_sentences(text)** that takes the input text as a parameter and returns the total number of sentences in the text. Sentences end with periods, exclamation marks, or question marks.



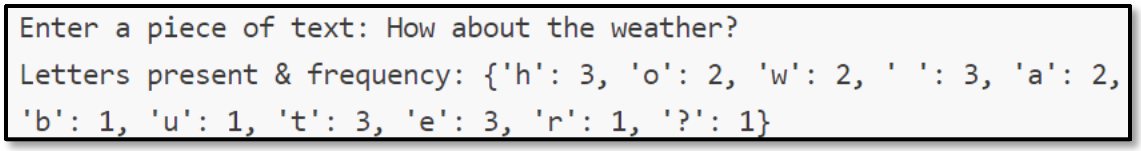
```
Enter a piece of text: Hi! How are you? I'm great. Thanks.
Number of sentences: 4
```

- Option 5: Punctuation Remover
    - Write a function **punctuation_remover(text)** that takes a string as input and returns a new sentence with all punctuation marks removed being replaced by spaces. Utilize the .replace() method to remove punctuation marks.



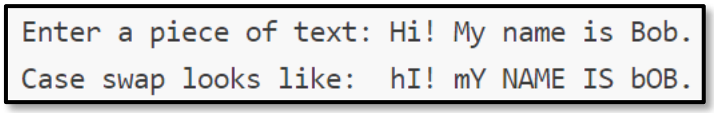
```
Enter a piece of text: Hi! How are you? I'm great. Thanks.
Without punctuation that would look like: Hi How are you I'm great Thanks
```

- Option 6: Letter Frequency
    - Write a function **letter_frequency(text)** that takes the input text as a parameter and returns a dictionary containing the frequency of each letter in the text. Ignore case sensitivity (treat 'A' and 'a' as the same). It is ok to include spaces and punctuation if you want to.



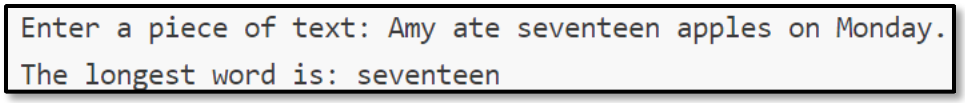
```
Enter a piece of text: How about the weather?
Letters present & frequency: {'h': 3, 'o': 2, 'w': 2, ' ': 3, 'a': 2,
'b': 1, 'u': 1, 't': 3, 'e': 3, 'r': 1, '?': 1}
```

- Option 7: Upper and Lower Conversion
    - Write a function called **convert_case(text)** that takes a string as input and returns the string converted to uppercase and lowercase alternatively. Utilize .isupper() and .islower() to determine case.



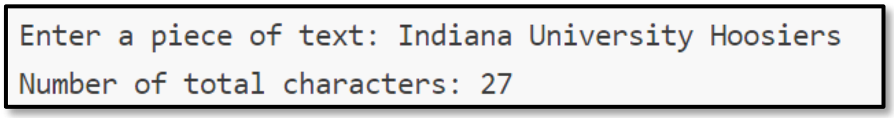
```
Enter a piece of text: Hi! My name is Bob.
Case swap looks like:  hI! mY NAME IS bOB.
```

- Option 8: Longest word
    - Write a function called **longest_word(text)** that takes a string as input and returns the longest word in the text.



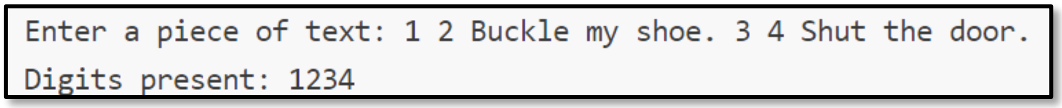
```
Enter a piece of text: Amy ate seventeen apples on Monday.
The longest word is: seventeen
```

- Option 9: Character Count
    - Write a function called **count_characters(text)** that takes the input text as a parameter, uses a count variable and for loop, returns the total number of characters (including spaces and punctuation marks) in the text.



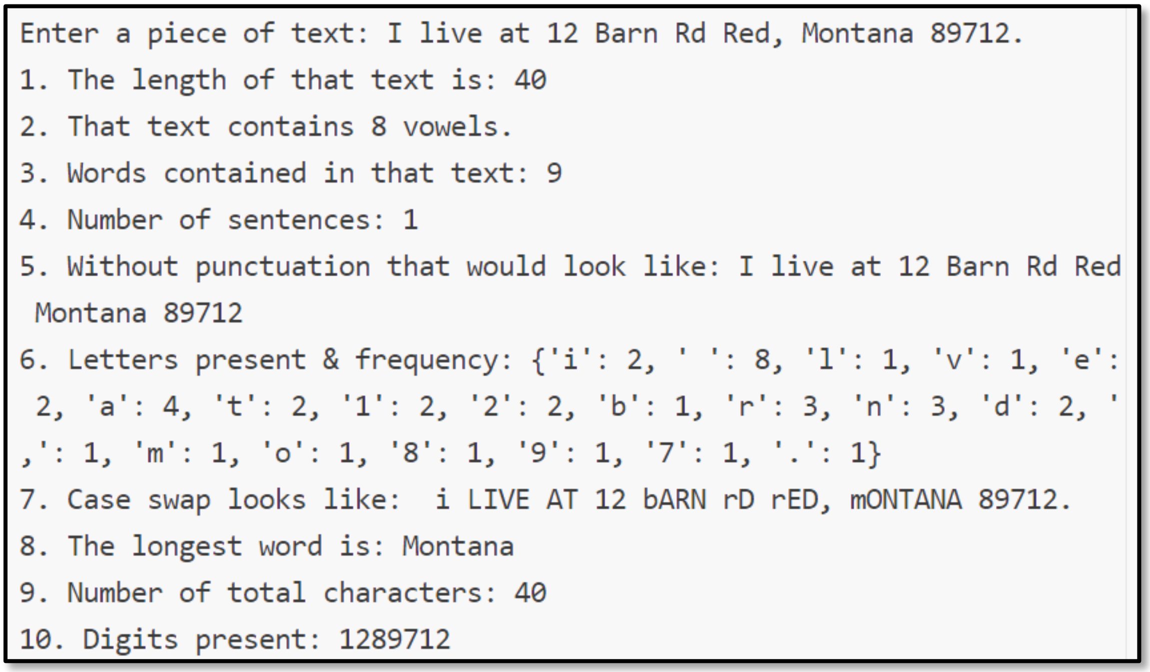
```
Enter a piece of text: Indiana University Hoosiers
Number of total characters: 27
```

- Option 10: Digit Extractor
    - Write a function **extract_digits(text)** that takes the input sentence as a parameter and returns a string containing only the digits found in the sentence. Utilize the .isdigit() method to check if a character is a digit.



```
Enter a piece of text: 1 2 Buckle my shoe. 3 4 Shut the door.
Digits present: 1234
```

Part 3: Output: Call each of the six functions you selected from the list above with the user-input string and print the results of each analysis. (24 pts, 4 points for each of the six function calls).
1 point is for including relevant text to explain the output given.



```
Enter a piece of text: I live at 12 Barn Rd Red, Montana 89712.
1. The length of that text is: 40
2. That text contains 8 vowels.
3. Words contained in that text: 9
4. Number of sentences: 1
5. Without punctuation that would look like: I live at 12 Barn Rd Red
 Montana 89712
6. Letters present & frequency: {'i': 2, ' ': 8, 'l': 1, 'v': 1, 'e':
 2, 'a': 4, 't': 2, '1': 2, '2': 2, 'b': 1, 'r': 3, 'n': 3, 'd': 2, '
,': 1, 'm': 1, 'o': 1, '8': 1, '9': 1, '7': 1, '.': 1}
7. Case swap looks like:  i LIVE AT 12 bARN rD rED, mONTANA 89712.
8. The longest word is: Montana
9. Number of total characters: 40
10. Digits present: 1289712
```

Part 4: Comments and Code Structure: Go through this coding file and explain what is happening. Leave code telling what different variables, functions and functions calls are used for. Write code explaining different code blocks. Using comments when coding is a critical skill to develop. Verify that your functions are at the top and your function calls are in the main. (10 points)

Part BONUS:

- Option 1 – Create additional function definitions and function calls from the list of options in Part 2. Five bonus points for each correctly executed function beyond the six required for the assignment. Must create the function and call the function.
- Option 2 – Create a menu that allows the user to choose which operations to perform on the input text. Up to 10 bonus points available. *Shown in Bonus Example Image Below*
- Option 3 – Move the functions into a separate .py file and import the module. Ensure you can still run your code without errors.

Assessment / Grading:

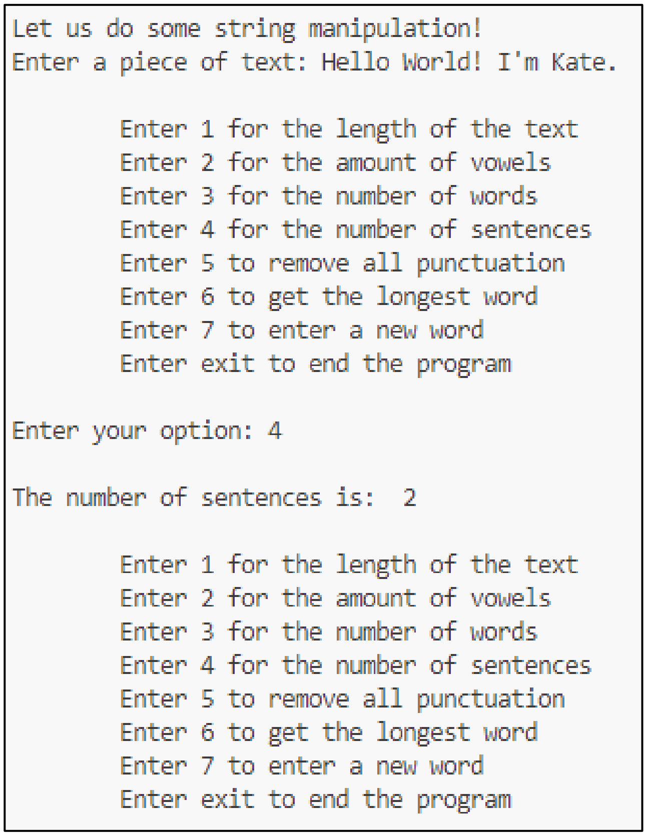
100 point assignment

- 6 points for user input
- 60 points for defining six functions from list
- 24 points for calling six user-defined functions
- 10 points for commenting / explaining code and correct code placement

Maximum score available on the assignment is **110 points** out of a possible 100 points.

Bonus Example Output

With Bonus Option #2 your program might look something like the images below. Ensure your options match your selected functions.



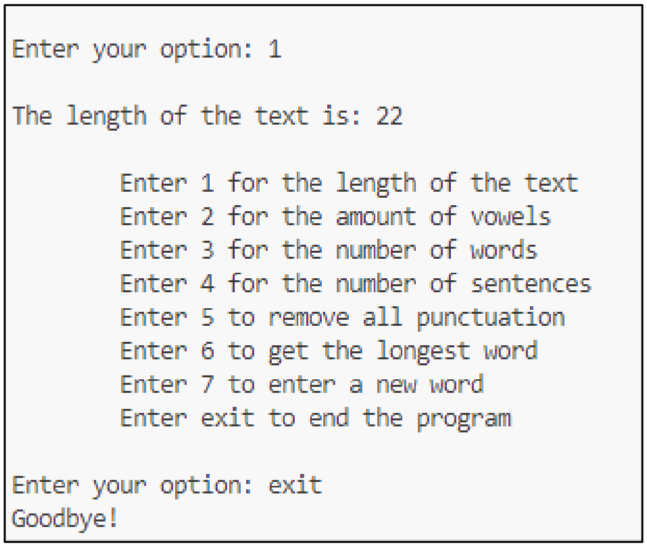
```
Let us do some string manipulation!
Enter a piece of text: Hello World! I'm Kate.

        Enter 1 for the length of the text
        Enter 2 for the amount of vowels
        Enter 3 for the number of words
        Enter 4 for the number of sentences
        Enter 5 to remove all punctuation
        Enter 6 to get the longest word
        Enter 7 to enter a new word
        Enter exit to end the program

Enter your option: 4

The number of sentences is:  2

        Enter 1 for the length of the text
        Enter 2 for the amount of vowels
        Enter 3 for the number of words
        Enter 4 for the number of sentences
        Enter 5 to remove all punctuation
        Enter 6 to get the longest word
        Enter 7 to enter a new word
        Enter exit to end the program
```



```

Enter your option: 1

The length of the text is: 22

        Enter 1 for the length of the text
        Enter 2 for the amount of vowels
        Enter 3 for the number of words
        Enter 4 for the number of sentences
        Enter 5 to remove all punctuation
        Enter 6 to get the longest word
        Enter 7 to enter a new word
        Enter exit to end the program

Enter your option: exit
Goodbye!
```