The following guidelines should be followed and will be used to grade your homework:
- All code to be implemented and submitted as a jupyter notebook (.ipynb) file.
- This is an individual homework assignment, no group submissions will be accepted. If you discuss in groups, please write your code individually and submit.
- Sample runs shown in the question should be used as a guide for implementation. However extensive testing needs to be done on your code to deal with all test cases that might possibly be executed.
- The high level algorithm, instructions for running of each cell and the expected results should be documented in the cell preceding the code using markdown language.
- Every code segment in the jupyter notebook cells should be well documented with comments. Use # in the code to provide comments and they should explain the algorithm and what the code segment is doing.
- Error checking in your code is very important and differentiates a high quality programmer from a low quality one. Hence you should account for invalid user inputs, infinite loops, out of range results, missing files, etc. and resolve them by appropriate error messages. The homework will be graded for robustness of your code.

1. (10 points) Write a program that rolls a 7 faced die 100 times and then tells you the number of 1s, 2s, 3s,..and 7s. It should also provide a graphical display of the frequencies. Sample runs are shown below:

First run:
Frequencies:
Number of 1s: 13
Number of 2s: 21
Number of 3s: 19
Number of 4s: 16
Number of 5s: 16
Number of 6s: 15
Number of 7s: 17
*1   *************
*2   **********************
*3   *******************
*4   *****************
*5   *****************
*6   ****************

*7   *****************

Second run:
Frequencies:
Number of 1s: 13
Number of 2s: 23
Number of 3s: 12
Number of 4s: 21
Number of 5s: 14
Number of 6s: 17
Number of 7s: 16
*1   *************
*2   ***********************
*3   ************
*4   *********************
*5   **************
*6   *****************
*7   ****************

2. (10 points) Write a program where the user and the program/computer play a number guessing game. The program should prompt the user for an integer (between 1 and 100, inclusive) then the program/computer has to guess what the user entered. Keep track of the number of iterations it takes for the computer to guess the number. Sample runs are shown below:

    Enter number to be guessed: *88*
    You entered 88, and it took the program 3 iterations to guess

    Enter number to be guessed: *55*
    You entered 55, and it took the program 19 iterations to guess

3. (10 points) Write a function that accepts a line of text and a single letter as input (case insensitive) and returns the number of times the letter is the first character of a word
    Sample runs are given below:
    Enter your line of text: *All the world's a stage, and all the men and women merely players. They have their exits and their entrances; And one man in his time plays many parts.*
    Enter your letter to use: *m*
    Your letter h occurs as the first letter: 3 times

4. (10 points) A Pangram is a sentence that contains every letter in the English alphabet. For example "The quick brown fox jumps over the lazy dog" is a pangram. Write a Python program to check if a string given by the user is a pangram or not. In addition, if the sentence is not a pangram, your program

should output the letters that were missing to make it a pangram. Sample runs are given below:

Enter a string: The quick brown fox jumps over the lazy dog
The given string is a pangram

Enter a string: The quick brown fox jumps over the lazy cat
The given string is not a pangram, you are missing the letters: d, g

Some more pangrams to check your program are:
> Pack my box with five dozen liquor jugs.
> The five boxing wizards jump quickly.

5. (10 points) Write a program to request a file name from the user and calculate the following statistics of the contents of the file:
   - Number of lines
   - Number of words
   - Number of characters
   - Average length of a word

   In this problem use the following definitions:
   A line is a sequence of characters that end with a newline (\n) character
   A word bounded by one or more spaces (or \n) on either side of it (or both sides)
   A character is any single length string, e.g. 'a', '-', etc. but not a space (or white space)

   An example file called Shakespeare.txt is included in the homework files
   Sample Run
   What is the filename: Shakespeare.txt
   Number of lines:
   Number of words:
   Number of characters:
   Average length of a word:

   Note: If your file statistics are different from the answer I have given above, please explain in your notes/markdown script how you arrived at your answers. For example if you use the readlines() function then it will count the last line which does not end with a newline (\n) as a line, that is fine as long as you understand it and are able to explain.