CS 242
Python Practice Exercise

(Programs are based on simple logic and are aimed to test your understanding of Python syntax and semantics. Only refer internet for Hints.)

1. Start Python interpreter and use it as a calculator. Initialize *radius* with some integer value. Calculate the volume of a sphere. (**To check:** enclosing the operations with and without different combinations of parentheses)

2. A function object is a value that can be assigned to a variable or can be passed as an argument. Write a function *calc* that takes 3 arguments, *FUNCTION NAME (ONE OF THE ARITHMETIC OPERATIONS), INTEGER1,* and *INTEGER2*. And returns the output to a *print* function that prints in a formatted way, where it displays the result along with the input. For example, *calc(add, x, y)*.

3. If you are given three sticks, you may or may not be able to arrange them in a triangle. Write a program that prompts the user to input 3 stick lengths and calls a function *is_triangle* to check whether the sticks can form a triangle or not.

4. Write a recursive function that prints fibonacci series of length n. *Hint:* Fibonacci Series is a series of numbers in which each number (*Fibonacci number*) is the sum of the two preceding numbers. The simplest is the series 1, 1, 2, 3, 5, 8, etc.

5. The mathematician S Ramanujan found an infinite series that can be used to generate a numerical approximation of π. Write a function *estimate_pi* that uses this formula to compute and return an estimate of π. Compute summation until the last term is smaller than 10^{-15}. And check the result by comparing it to *math.pi*.

$$\frac{1}{\Pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!\,(1103 + 26390k)}{(k!)^4\, 396^{4k}}$$

6. Write a function *my_reverse* that takes string as an input, reverse it, without using in-built function and returns the reversed string.
7. Write a function *find_count* that takes *string/list* as first argument, *index* as second argument, and *elem* to search as a third argument. It returns the number of times and position of all the occurence of the *elem* following the parameter *index*.
8. Write a function that takes a list of numbers and returns the cumulative sum; that is, a new list where the ith element is the sum of first *i+1* elements from the original list. For example
   a. Input list : [1,2,3]
   b. Output : [1,3,6]
9. Two words are anagrams if you can rearrange the letters from one to spell the other. Write a function called *is_anagram* that takes two strings and returns True if they are anagrams.
10.   Write a function that accepts a string, store that string as a key in a dictionary, and it's value is another dictionary, where key is character present in the word and it's value is the frequency of that character in the word. For example, string : *mississippi* dct = { 'mississippi' : {'m':1, 'i':4, 's':4, 'p':2}.
   Hint: Use *get()* function for character-frequency dictionary.
11.   Read the file words.txt, call the function created in 10th problem to generate word-character-frequency dictionary. Now dump this dictionary into a *pickle* format. So that it can be loaded again directly as dictionary.
   Hint: Explore *pickle* module
   (https://docs.python.org/3/library/pickle.html).
12.   Read the file words.txt, randomly select 200 words. And sort the file based on the length of the word in decreasing order. In case of a tie, sort, following the chronological sequence of words. Write the sorted list in a file.
   Hint: Use *random.sample* for selecting 200 words. Check *sample* function from *random* module.

13.   The *os* module provides a function called *walk*. Read the documentation (https://docs.python.org/3/library/os.html) and write a function *get_files_list* using it to print the names of files, in a given directory and its subdirectories.

14.   Write a function called *sed* that takes as argument a pattern string, a replacement string, and two filenames. It should read the contents from the first file and writes the contents into the second file. If pattern string appears anywhere in the file, it should be replaced with the replacement string.

Note: All the arguments to the function *sed* should be accepted from command line.