

CALIFORNIA STATE UNIVERSITY, FRESNO
COMPUTER SCIENCE DEPARTMENT

Simulation (CSCI 154)

Instructor: A. A. Panagopoulos

Assignment I

Release date: 02-21-2019

Due date: 03-05-2019

Instructions:

- ❖ This is a group assignment.
- ❖ If you haven't done it already, please register your group through Canvas. New groups can be registered until the due date of the first group assignment. If the due day has passed and you have not registered your group yet please contact the instructor.
- ❖ Groups are limited to up to three students (≤ 3) and generally remain the same for all group assignments and projects (except due to extraordinary circumstances and after instructor approval).

Each group should have a zip file with the .py and a readme file (if necessary) submitted through Canvas by the due date of the assignment. Please make sure that the submitted file is not corrupted. Please, also make sure to include the names and ID# of the group members.

1) [50/100] Write a python script that repeatedly prompts the user to select one of the following menu options:

- A. Display a Pascal's triangle of height H.
- B. Display the value of the factorial of N.
- C. Approximate and display the Euler's number.
- D. Approximate and display the value of the sine of X.
- M. Display these menu options.
- Q. Exit from the program.

The menu of options should be displayed once the execution begins, whenever the Option M is selected, and whenever an invalid menu option is given. The program should accept both uppercase and lowercase letters for the menu options (e.g., a and A). Option Q exits the program, while the rest of the menu options are detailed below:

- **Option A** Write a python script that generates and appropriately displays a Pascal's triangle. The Pascal's triangle is a triangular array where, starting from the top, each cell is populated by adding the contents of the 2 cells directly above it, for instance a Pascal's triangle with height 7 is the following:

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1

```

The program asks the user for the height of the triangle until it gets as input a non-negative integer (i.e., {0,1,2,3...}).

- **Option B** will ask the user to enter an integer value N. If N is a non-negative integer (i.e., {0,1,2,3...}), it will calculate and display the factorial of N. If the input is not a non-negative integer, the program should ask for a non-negative integer until it is given. You are requested to not use existing function to calculate the factorial.
- **Option C** approximates and displays the Euler's number (*e*). *e* is a mathematical constant and equals the base of the natural logarithm; i.e. the unique number whose natural logarithm is equal to one. You are requested to not use existing functions to calculate the Euler's number. The Euler's number is approximately equal to 2.71828 and can also be calculated as the sum of the infinite series:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{1} + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \dots$$

Approximate the Euler's number by expanding the power series until the absolute value of the next term in the series is less than 10e-100.

- **Option D** will ask the user to enter a real value X (measured in radians), and approximate and display the value of the sine of X. If the input is not a real number the program should persist until a real value is given. You are requested to not use an existing function to calculate the sine of X but use a loop. In more detail, the sine of a real number x can be calculated as the power series:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

Approximate X by expanding the power series until the absolute value of the next term in the series is less than 10e-8.

Make sure to write docstrings for each class and method you implement to support the help() function usage. Also follow the PEP8 naming conventions and do not forget to comment thoroughly and licence your code. Also try to write in a Pythonic way ☺

- 2) [40/100] One claim that was popular back in the days, stated that the order of the letters of each word doesn't matter, except for the first and the last one, when reading a sentence. This claim was using the following paragraph to back its claims:

"Aoccdrnig to rscheearch at an Elingsh uinervtisy, it deosn't mtttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer is at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit a porbelm. Tihs is bcuseae we do not raed ervey lteter by itslef but the wrod as a wlohe."

Let's evaluate this claim, is it a hoax? You will write a script that asks the user to enter one sentence at a time, until the user gives you an empty sentence (presses just enter as the input). Then the program should scramble each word of each sentence and print the scrambled sentence to the console. Ask your friend to read it 😊

You are requested to write the script using the following specification: Create a class named Scrambler that will be given the paragraph and scramble it. The interface will include the following two methods:

- Scrambler(): The constructor will be asking from the user to give the paragraph and would store it to a instance attribute of your choice.
- scramble_it(): This method will return the scrabbled paragraph to be printed by the main body of the program.

Make sure to write docstrings for each class and method you implement to support the help() function usage. Also follow the PEP8 naming conventions and do not forget to comment thoroughly and licence your code. Also try to write in a Pythonic way 😊

- 3) [10/100] Understand and comment the following code snippet and write a single line of code that enables the class Hamburger to store as instance arguments the meat choice and as many extras as given at the instantiation call.

```
class Hamburger:
    def __init__(self, **kwargs):
        *** your code goes here ***

    def __str__(self, sep=' '):
        return(', '.join(list(self.__dict__.values())))
```

```
burger1 = Hamburger(meat="chicken", extra1="cheese", extra2="ketchup")
print(burger1)
```

For instance, the above code snippet should print: "chicken, cheese, ketchup".

Resources/Extra reading:

- ❖ http://en.wikipedia.org/wiki/Pascal's_triangle
- ❖ <https://en.wikipedia.org/wiki/Factorial>
- ❖ https://en.wikipedia.org/wiki/Power_series