Workshop 1 Optional Exercises

1 Introduction

These are a set of practice problems that reinforce everything we covered in Workshop 1. These range from easy to quite challenging, so don't feel discouraged if you're struggling to complete one of these problems. Some of these problems are based on example problems posted to edabit.com.

Feel free to work with others on these problems.

2 Problems

2.a Unix Time Conversion and Modulo

The final basic math operation in Python that we didn't cover is called modulo. It gives you the remainder from a division operation. Below are some examples utilizing the modulo operator (%). A resource for the modulo operator can be found here.

$$5 \,\%\, 2 = 1$$

$$4 \,\%\, 2 = 0$$

$$5 \,\%\, 120 = 5$$

$$2841529 \,\%\, 9 = 3$$

Unix Time is the main standard for time keeping on computers, it is defined as the number of non-leap seconds that has passed since Midnight GMT on January 1st 1970.

Because Unix Time is defined based on the number of seconds that have passed since the 1970s it is quite unreadable to humans. In this problem you will write functions to convert a time given in Unix Time to the number of days, hours, minutes, and seconds that have passed since Jan. 1st 1970.

Given an input in seconds (unix_seconds), return how much time is passed since then in days, hours, minutes, and seconds (n_days, n_hours, n_minutes, n_seconds).

While this can be done easily with the datetime package we learned about please try to do this purely using basic math operations in python. Here are some Unix Times to work with:

1. 946710000
2. 31561200
3. 1356103399

You can also use the function np.random.randint() to generate more random unix times to practice with. The documentation on np.random.randint() can be found here

2.b Vowel Counter

Vowels are an important part of the English language. In this problem you will write a function that, given any arbitrary string, will count the number of vowels that appear in that string. For the sake of this problem consider Y a vowel, meaning your list of vowels is A, E, I, O, U, Y.

Example strings:

1. "The Quick Brown Fox Jumped Over the Lazy Dog."
2. "The Aleutian Islands are home to sea otters, sea urchins, and sea birds."
3. "The 2024 Summer Olympics were held in Paris, France and featured sports such as Breakdancing, Running, and Artistic Swimming."

2.c Check if Two Lines are Parallel or Perpendicular

Given two lines check if they are either parallel or perpendicular. The formula to get the angle between two lines is:

$$\theta = tan^{-1}(\frac{m_1 - m_2}{1 + m_1 m_2}) \tag{1}$$

Where tan^{-1} is arctan,m_1 is the slope of Line 1, and m_2 is the slope of line 2. To calculate arctan you can use the function np.arctan(). Just know that numpy works in radians, can convert between radians and degrees using the numpy functions np.deg2rad() and np.rad2deg()

Lines will be defined as two points along the line as X,Y pairs. Here are some examples to get you started.

1. Line 1: (0,0), (1,1); Line 2: (0,3), (3,6)
2. Line 1: (0,1), (1,0); Line 2: (0,0), (1,1)
3. Line 1: (2,8), (4,0); Line 2: (-3,0), (0,12)

2.d Perimeter and Area of a Triangle

Given a list containing the three points that define a triangle calculate the area and perimeter of the triangle. The points will come in a list of lists as X,Y pairs e.g. [[0,0],[1,1],[1,0]] is a triangle composed of the points (0,0),(1,1), and (1,0).

For this problem I recommend using Heron's Formula. You can read about it here from Wikipedia.

Below are a list of triangle's for you to work with:

1. $[[0, 0], [1, 0], [1, 1]]$
2. $[[5, 8], [12, 0], [0, 12]]$
3. $[[8, 8], [4, 4], [3, 3]]$
4. $[[0.5, 0.5], [-3, 3], [4, -3]]$
5. $[[3, 1], [22, 33], [108, 202]]$

2.e Number of Friday the 13ths in a Year

Friday the 13th is considered an unlucky day, in this problem we will figure out how many times Friday the 13th occurs in a given year using the datetime package we learned about.

For this problem you will write a function that takes a year as an input and outputs the number of Friday the 13ths that occurred that year.

To make this problem easier I recommend reading the documentation for the datetime package linked below. Here is a link to the datetime documentation for a function that may be useful

2.f Summing Numpy Arrays

Below is a block of code that generates a 3x3 numpy array filled with random values. Your goal is to write a function which, given a random 2-D numpy arrays returns the sum of the whole array, the sum along each column, and the sum along each row.

For this question I will not tell you which numpy function to use and instead I recommend you search the numpy documentation on your own to find the function you want to use. Here is a link to the documentation for numpy.

Your function should output three things: 1.) a float that is the sum of the entire array, 2.) a numpy array containing the sum of each column, 3.) a numpy array containing the sum of each row.

Code block begins here:

```
import numpy as np

def generate_random_3x3_array() -> np.ndarray:
    '''
        This function generates a random 3x3 array of floats
        as a numpy array, subtracts 0.5 to have the values span the
        domain above and below zero and then multiplies by five to increase
        their final value.

        Inputs:
            None

        Outputs:
            - rand_array (np.ndarray): The array of random values we
                generated and then modified.
    '''
    rand_array = (np.random.rand(3,3) - 0.5) * 5

    return rand_array
```

2.g Calculator Class

This final problem is a little harder than the others, it requires you design pieces of code and how they interact with one another on your own. I recommend you write/draw out things on a piece of paper before you start actually coding so that you can fully think things through.

In this problem your goal is to make a custom calculator class. A calculator object should be able to add, subtract, multiply, and divide. It should also store the results of the last three operations you asked it to carry out.

Your methods to add, subtract, multiply, and divide should take two numbers as inputs and output the result of the desired operation.

Your calculator class should also have a method to report the results of the last three operations you asked it to perform. This means your class needs to both store data, and modify the data it stores. Some example code of what your final calculator class should be able to do is reproduced below.

```
# Initialize a calculator object
ex_calc = calculator()
# Add two numbers
ex_calc.add(4,5)
# Multiply two numbers
ex_calc.multiply(4,5)
# Divide two numbers
ex_calc.divide(3,3)
# Report the last three operations results
ex_calc.history()
```

When I ran my code I got the output reproduced in the block below:

```
9
20
1
[1,20,9]
```