Python Boot Camp

Summer 2020
Instructor: Kelvin Ma
Assignment # 02
Due: Saturday June 21, 2020, 5:00 PM EST.

This assignment contains a few parts:

- **Part I – Programming Challenges for Everyone**
    - Every student must do these problems
- **Part II – Programming Challenges for Electrical Engineers**
    - You can do these problems if you are studying electrical engineering – or for fun.
- **Part III – Programming Challenges for Chemical Engineers**
    - You can do these problems if you are studying chemical engineering – or for fun.
- **Part IV – Programming Challenges for Mechanical and Civil Engineers**
    - You can do these problems if you are studying mechanical or civil engineering – or for fun.
- **Part V – Extra credit**
    - These problems serve as extra credit. You MUST do two of the four previous parts.

Every student is required to do Part I of the assignment + one other part depending on your major. Note that the second question in parts II to IV are always harder.

If you want to do all of it – please feel free to do so, however, extra credit is only available in Part 5.

Good luck and happy coding!

Table of Contents

Part I. Programming Challenges for Everyone

Programming Challenge #1 – The odd list (2 points)

Write a function in Python that takes a single input. The input to the function should be a list. The function should return a new list containing only the odd numbers from the input list.

Programming Challenge #2 – Multiples (2 points)

Write a function in python that has one input. The input to the function should be an integer. The function should return a list of all integers between 1 and 100 (non-inclusive) that are multiples of the input.

Programming Challenge #3 – Sorted or not (3 points)

Write a function that has one input. The input to the function should be a list of numbers (either integers or floating-point numbers). Your function should determine if the list is sorted in ascending order or not.

Programming Challenge #4 – Slot Machine (2 points)

You are a software engineer in Las Vegas, writing a small script to simulate a slot machine. The slot machine has three columns much like the one shown on the right. Each column has the possibilities of:

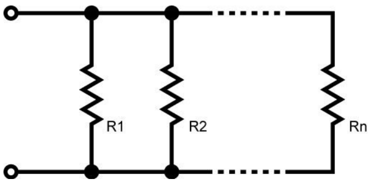
['Cherry', '7', 'J', '10', 'Apple', 'A', 'Q', 'M']

Write a function that picks three random items from the list. Put these three items into a new list and return it.

PART II. PROGRAMMING CHALLENGES FOR ELECTRICAL ENGINEERS

You may recognize the first problem …

EE Programming Challenge #1 – Parallel Circuit (1 points)

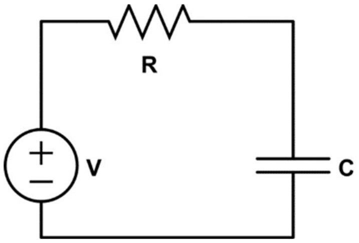


The circuit shown on the right shows a set of resistors in parallel configuration. The formula to finding the equivalent resistance is:

$$R_{eq} = \frac{1}{\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \cdots + \frac{1}{R_n}\right)}$$

Write a function that takes in one input: a list containing all the resistor values. The output of the function should be the equivalent resistance as a floating-point number.

EE Programming Challenge #2 – RC Discharge (2 points)



The image on the right shows a simple RC circuit. When the input voltage is turned off, the voltage across the capacitor is defined by the following discharge rate:

$$V_C = V_S \times e^{-\frac{t}{RC}}$$

Where V_C is the voltage across the capacitor, V_S is the source voltage, e is Euler's number, t is time in seconds, R is the resistance in Ω and C is the capacitance in Farads.

Write a function that takes in 4 inputs: Vs, R, C, and t as a list. Your output should be a list of capacitor voltages. Note that to perform an exponential in Python, you will need to use the math module.

$$e^x \rightarrow \text{math.exp(x)}$$

EE Programming Challenge #3 – Wireless Multipath Fading (3 points)

Multipath fading of a wireless signal (cellular, Wi-Fi, etc.) occurs due to a receiver (or transmitter) in motion. This fading is due to the doppler effect. A quick way to estimate the number of fades that occur per second is:

$$\frac{\#\ of\ fades}{second} = \frac{v}{\lambda}$$

Where v is the velocity of the object in motion in m/s, and λ is the wavelength of the wireless signal. Write a function in Python that takes two inputs. The first input is a list containing different velocities in m/s and the second input is a floating-point number representing the frequency of wireless transmission. Remember that:

$$\lambda = \frac{c}{f}$$

Where c, the speed of light, is 3×10^8 m/s. The output of the function should be a list containing the fades per second of each frequency. Note that when testing this function, you may want to try frequencies above 28 GHz – if you are curious about the multipath fading of 5G networks.

Part III. Programming Challenges for Chemical Engineers

ChemE Programming Challenge #1 – Charles' Law (2 points)

Charles' Law is a special case of the ideal gas law where the pressure of a gas remains constant. Charles' Law can be written as:

$$\frac{V_i}{T_i} = \frac{V_f}{T_f}$$

Where V_i is the initial volume, T_i is the initial temperature in Kelvin, V_f is the original final volume and $T_f$is the final temperature in Kelvin.

Write a function in Python that takes in 3 inputs: V_i, T_i and T_f. Assume that the units for temperature is given in *Celsius* (your function must handle conversions internally). Your output should be the final volume.

ChemE Programming Challenge #2 – Average Atomic Mass (4 points)

Average atomic mass can be calculated using the following formula:

$$Average\ Atom\ Mass = \sum(Fractional\ Abundance)(Isotope\ atomic\ mass)$$

For example, Chlorine has two isotopes. Chlorine-35 has an isotope mass of 34.9689 u and chlorine-37 has an isotope atomic mass of 36.9659 u. In any sample of chlorine atoms, 75.771% will be chlorine-35 and 24.229% will be chlorine 37. The average atomic mass is:

$$Avg\ chlorine\ atomic\ mass = (0.75711)(34.9689) + (0.24229)(36.9659) = 35.43177\ \text{u}$$

Write a function in Python that takes in 2 inputs: a list of isotope atomic masses, and a list of fractional abundances. Your output should be a floating-point number that represents the average atomic mass. 😊

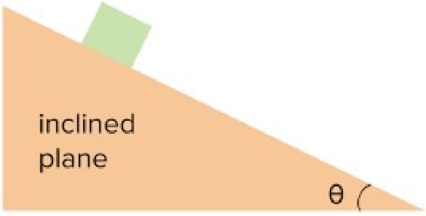
Chemical Engineers – please feel free to give me feedback on these problems, my chemistry is really weak.

PART IV. PROGRAMMING CHALLENGES FOR MECHANICAL AND CIVIL ENGINEERS

Honestly, I could not find many good problems for you guys – so this may just feel like Physics I, but hopefully it helps you improve your Python skills, nonetheless.

MechE & CE Programming Challenge #1 – Inclined Plane (2 points)

A box of mass m is released from the top of an inclined ramp with angle θ. Friction is negligible on this ramp. To find the time it takes this box to reach the bottom of the ramp, we can solve the following equation:



$$x = \frac{1}{2}a_x t^2$$

$$t = \sqrt{\frac{2x}{a_x}}$$

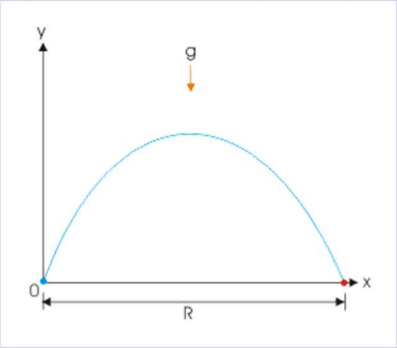
Where a_x is the acceleration due to gravity in the x-direction. $a_x = mg \cdot sin(\theta)$.

Write a function in Python that takes in 3 inputs. The first input is a list of different ramp inclinations in degrees. The second input is the mass of the box. The last input is the length of the ramp. The output of the function should be the time it takes to reach the end of the ramp.

Note that to use trigonometric functions in Python, you will need the math module:

$sin(x)$ → math.sin(x)

MechE & CE Programming Challenge #2 – Projectile Motion (4 points)



Projectile motion is a form of motion where an object is thrown or otherwise projected into the area. The object is only subject to acceleration from gravity. The image on the left is a trace of the trajectory. The distance that the object travels is given by:

$$R = v \times \frac{\sin(2\theta)}{g}$$

Where R is the distance in meters, v is the initial magnitude of the velocity, θ is the angle in which the object was thrown in degrees, and g is the acceleration of gravity.

Write a function in Python that takes in 3 inputs. The first input is the velocity in which the object was thrown as a floating-point number. The second input is a list of different angles for which the object was thrown (in degrees). The third input the acceleration due to gravity.

If you would like to have some fun, here are the gravities on different planetary bodies in our solar system:

Planetary Body Name	Gravity (m/s^2)
Sun	274
Mercury	3.7
Venus	8.9
Earth	9.8

Mars	3.7
Jupiter	24.9
Saturn	10.4
Uranus	8.9
Pluto – Yes, I still consider it a planet, I don't care what IAU says.	0.58

The output of the function should be the distance in which the object travelled. Try playing around with different gravities 😊.

Part V. Extra Credit Problems

Extra Credit Challenge #1– Playing slots (2 points)

Call the slot machine function from the previous challenge (with a loop) 1000 times, count how many times a jackpot occurs. A jackpot, for the sake of this assignment, is when all three elements are the same. For example,

['7','7','7'] is a jackpot, and ['Cherry', 'Cherry', 'Cherry'] is also jackpot, etc.

Hint: You may want to write a function to check if all three elements of a list are the same.

Extra Credit Challenge #2– Uno deck (3 points)

There are 108 cards in an Uno deck. There are four suits, Red, Green, Yellow and Blue, each consisting of one **0** card, two **1** cards, two **2**s, **3**s, **4**s, **5**s, **6**s, **7**s, **8**s and **9**s; two **Draw Two** cards; two **Skip** cards; and two **Reverse** cards. In addition, there are four **Wild** cards and four **Wild Draw Four** cards.

You may need to refer to the standard playing card deck example to do this problem. Good luck!