UNIVERSITY OF COLORADO - BOULDER
Robotics Program

COEN 5830 (ROBO 5000) - Intro to Robotics

Homework #1 (Assigned: 8/30, Due: 9/17 11:59pm on *Canvas*)
Python Software Development

Instructions

This homework assignment has no written component and all questions should be completed through Python code. Be sure to comment code so the grading staff can easily follow your logic. **You are encouraged to discuss your work with other students, but submitted work must be your own. Please indicate on the first page of your answer sheet who your collaborators for this homework are.**

Please complete all programming questions in a single .py file with the naming convention "LastName_FirstName.py". For example, my submission file would be called "Beuken_Leopold.py".

1. This question asks you to set up Python and VSCode on your local machine. [5]

    (a) The course will utilize Visual Studio Code (VSCode) as the primary development environment for code. This is the most popular editor and has become an industry standard in recent years. If you do not have VSCode and/or Python installed on your machine, follow the instructions in the link below to install them. **Do not complete all the steps listed, only steps 1 and 3 are needed.**
    `https://www.mooc.fi/en/installation/vscode/#run-source-code`

    (b) SciPy ("sigh pie") is a Python library used for scientific computing and is very useful for engineers. The following link has instructions for installing the library.
    `https://scipy.org/install/`

    (c) NumPy is a Python library which supports the use of large arrays and related linear algebra functions. The following link has instructions for installing the library.
    `https://numpy.org/install/`

    (d) Matplotlib is a Python library which is very useful for generating figures and plots. The following link has instructions for installing the library.
    `https://matplotlib.org/stable/install/index.html`

2. For this course, you are expected to be comfortable with fundamental programming concepts (variables, conditionals, iteration, etc.). The following questions are designed to test these basic principles and it is expected that you spend no more than 15 minutes per question. If you require more time (or need excessive online resources) to complete the questions then it is suggested that you invest some time to sharpen your Python skills. Programming fundamentals will not be taught explicitly in this course and it is your responsibility to fill any gaps in your knowledge. The University of Helsinki has a free introductory course to Python programming, which will serve as an excellent introduction:

    `https://programming-24.mooc.fi`

(a) [5 points] Consider the following summation,

$$\sum_{k=1}^{N}(k^2+1) = 2+5+10+\dots$$

.

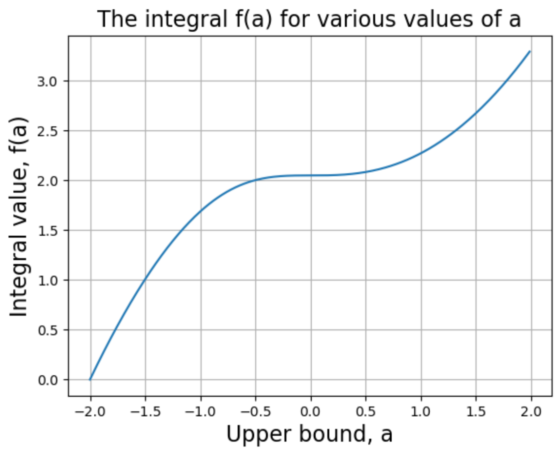
Write a Python function *sequence* that takes one input, namely the maximum sum S_m. This function must compute the minimum number of terms such that the sum of terms is greater than S_m. These terms must be returned in an array (or list). Example: The input 22 should return a list: [2, 5, 10, 17].

(b) [5 points] Write a Python function *smallest_multiple* that takes a list of positive whole numbers as input. This function must compute and return the smallest whole number that can be divided by each of the numbers in the input array without a remainder. Example: The input [2, 4, 7] should return 28.

3. [10 points] Consider the following integral:

$$f(a) = \int_{-2}^{a} \frac{x^2}{(1+x+x^2)^{0.5}} dx$$

Write a Python program that reproduces the figure below with all annotations included (axis labels, title, tick marks, etc.). The Figure is produced by plotting $f(a)$ as a function of the upper bound a of the integral. The upper bound a is discretized between -2 and 2 using increments of 0.01. The function *quad* in the *scipy.integrate* subpackage may be useful for this problem.



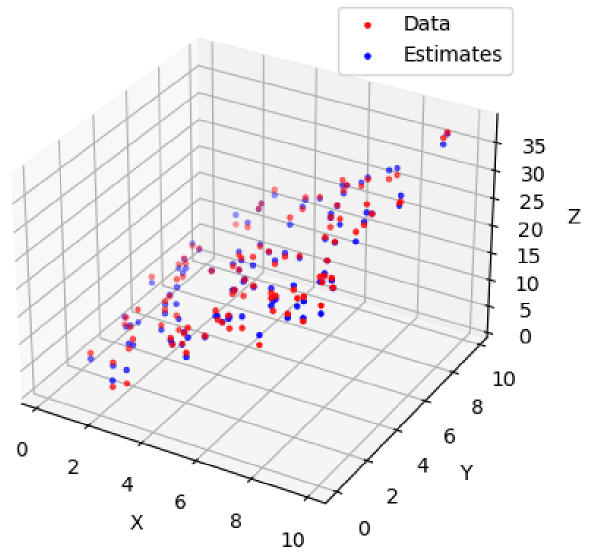
4. [10 points] A circular prime number is a prime number of which all permutations of its digits are also prime. For example, 199 is a circular prime number as all permutations of digits [199]: 199, 919, 991 are also prime numbers.

Write a function *cprimes_list* which has one argument, namely the integer N. The functon must compute and return a numpy array of the first N circular prime numbers. The array of circular prime numbers must contain unique entries in ascending order.

For example, the call *cprimes_list*(9) must return the first 9 circular prime numbers in a numpy array: [2, 3, 5, 7, 11, 13, 17, 31, 37]. Hints: The function *permutations* in the *itertools* module may be useful for this problem. Also consider using a "set" data structure to remove repeated numbers.

5. [10 points] Linear regression helps us find a mathematical model from data in the form of $y = mx + b$ where x represents the independent variable (predictor), y represents the dependent variable (target) and m and b are model parameters determined through the regression process. Multiple linear regression is an extension of linear regression where multiple independent variables are present creating a model of the form $z = ax + by + c$. Notice that there are now two independent variables, x and y and a single dependent variable z.

   For this problem, import data from the file *q5.csv* to determine a multiple linear regression model fitting the x and y independent variables to the z dependent variable (the data columns in the file are ordered x, y, z). Your code should print out the relevant coefficients in the order [a, b, c]. You are also required to generate a 3-D scatter plot similar to the one shown in the figure below. Hint: The module *LinearRegression* from *sklearn.linear_model* may be useful for this problem. Use the *loadtext* function from *Numpy* to load the .csv file.



6. [5 points] Define a class named Square which inherits the Rectangle class. The sides of a square are all of equal length, which makes the square a special case of the rectangle. The new class should not contain any new attributes. See file *q6.py* file for skeleton code. The client code in the .py file is expected to produce the output:

   square 4x4
   area: 16

7. [10 points] The file *q7.py* contains code for a two-player word game. The parent class **WordGames** contains attributes that keeps track of how many games each player has won and also the number of rounds a particular game will last. The method *round_winner* determines which player won the previously played round based on their input to the console. This input is generated from the *play* function that handles the gameplay for each round. The nominal *round_winner* method determines the round winner by chance.

   Define a class **RockPaperScissors** which allows you to play a game of rock-paper-scissors. The rules are as follows:

   - rock beats scissors (the rock can break the scissors but the scissors can't cut the rock)
   - paper beats rock (the paper can cover the rock)
   - scissors beats paper (the scissors can cut the paper)

   The class **RockPaperScissors** is required to inherit from the **WordGame** class already defined and will override the *round_winner* method. If the input from either player is invalid, they lose the round. If both players type in something other than rock, paper or scissors, the result for that round is a tie.