Koç University
COMP 125: Programming with Python
Homework #1

Deadline: March 14, Sunday at 23:59pm
Submission through: Blackboard
Make sure you read and understand every part of this document

This homework assignment contains 4 programming questions.

Download Hw1.zip from Blackboard and unzip the contents to a convenient location on your computer. Each file in Hw1.zip contains the starter code for one programming question.

Solve each question in its own file. **DO NOT CHANGE THE NAMES OF THE FILES. DO NOT CHANGE THE HEADERS OF THE GIVEN FUNCTIONS (FUNCTION NAMES, FUNCTION PARAMETERS). DO NOT USE TURKISH CHARACTERS.**

When you are finished, compress your Hw1 folder containing all of your answers. The result should be a SINGLE compressed file (file extension must be .zip, or .rar). Upload this compressed file to Blackboard.

Q1: Class Grade - 25 pts

Throughout the semester you have taken three midterms in your science class, however your instructor could not decide how to assign the students' course grades. She is considering two different methods and she wants to choose the method which will output the higher course grade for the students. Open ClassGrade.py and help her to design a program through the following steps:

- First method she is considering is to drop the minimum midterm grade and calculate the average grade using the remaining two midterm grades (each will now weigh 50%). Implement drop_minimum(grade1, grade2, grade3) accordingly and **return** the average grade. You **cannot** use the built-in **min** function.

- As the exams tend to become harder, second method she is considering is to adjust the weights of the exams such that midterm1 will have a weight of 45%, midterm2 will have a weight of 30% and midterm3 will have a weight of 25%. Implement change_weights(grade1, grade2, grade3) accordingly and **return** the resulting grade.

- Now it is time to decide which method is better for the students. Implement compare_results(average1, average2) such that it will **return** the average which is higher than the other.

Q2: Taylor Series - 25 pts

Consider the Taylor series expansion of ln(1+x) that converges for -1 < x ≤ 1:

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \dots$$

Write a function taylor_series_ln(x, terms) that computes the value of ln(1+x) according to this Taylor series expansion. Your function has two inputs:

- x: this is the value for which we want to compute ln(1+x)
- terms: denotes how many terms should be considered in the Taylor series expansion. In the above example, terms=5. There may be fewer terms or more terms.

Open TaylorSeries.py, implement your code inside the function: taylor_series_ln(x, terms). Do not forget to **return** your result.

Use the example function calls under main() to test your taylor_series_ln function.

HINT: To check whether your solution is correct, compare the return values of your function against the real value of ln(1+x), which you can compute using a calculator. Note that the Taylor series converges for -1 < x ≤ 1, so the x values you pick should be within that range. While your function may be imprecise when # of terms is low, as you increase the # of terms, the output of your function should converge to the real value of ln(1+x).

Q3: Prime Numbers - 25 pts

A prime number is a number that is only evenly divisible by itself and 1. For example, 3 is prime because it can only be evenly divided by 1 and 3. However 8 is not prime because it can be divided evenly by 1, 2, 4, and 8.

Note that 1 is not a prime number and the smallest prime number is 2.

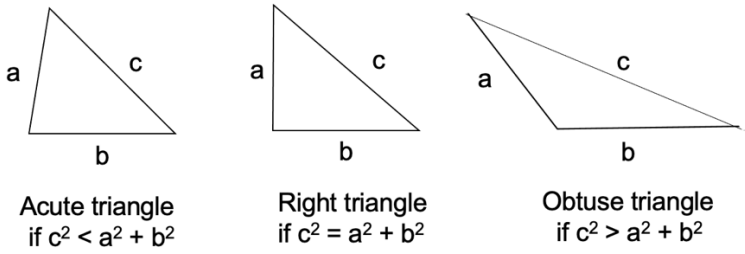
Open PrimeNumbers.py and implement the function prime_numbers(n). **Your function should print all of the prime numbers from 1 to n (included).** Use the example function calls under main() to test your prime_numbers(n) function.

e.g.: When you call prime_numbers(20), your console should look like this:

```
2
3
5
7
11
13
17
19
```

Q4: Triangle Types - 25 pts

In geometry, the following rule is employed to determine whether a given triangle is right, acute or obtuse:



Acute triangle if $c^2 < a^2 + b^2$

Right triangle if $c^2 = a^2 + b^2$

Obtuse triangle if $c^2 > a^2 + b^2$

where c is the largest side length

In this question, you are asked to write a program which determines whether a given triangle is a right, acute or obtuse triangle. Open TriangleTypes.py and implement your code inside triangle_type(side1, side2, side3) function accordingly. You **cannot** use the built-in **max** function. As the final step:

- If your triangle is acute, **return** 1.
- If your triangle is right, **return** 2.
- If your triangle is obtuse, **return** 3.

Submission and Grading

Solve each question in its own file. **DO NOT CHANGE THE NAMES OF THE FILES. DO NOT CHANGE THE HEADERS OF THE GIVEN FUNCTIONS (FUNCTION NAMES, FUNCTION PARAMETERS). DO NOT USE TURKISH CHARACTERS.**

When you are finished, compress your Hw1 folder containing all of your answers. The result should be a SINGLE compressed file (file extension must be .zip, or .rar). Upload this compressed file to Blackboard.

Follow instructions, print messages, input-output formats closely. **Your code may be graded by an autograder**, which means any inconsistency will be automatically penalized.

After you submit your Hw1, download it from Blackboard to make sure it is not corrupted and it has the latest version of your code. You are only going to be graded based on your Blackboard submission. **We will not accept homework via e-mail or other means**.

Code that does not run (eg: syntax errors) or does not terminate (eg: infinite loops) will not receive any credit.

Happy coding! ☺