

Make-up Exam

CmpE 150, Introduction to Computing, Spring 2023

Instructor: Mehmet Turan
TAs: Kutay Altıntaş, Mehmet Özer

Description

For each question, you are expected to write a Python script. Script naming should follow the convention `answer_<<question_no>>.py` (e.g., solution script for the 1st question should be named as `answer_1.py`). Each solution will be tested using several test inputs. A solution is graded according to the tests it successfully passes.

If a question ask you to take an input from the user, you may use `input()` function. When a question asks you to return something, define a function for that.

Grading

Question	1	2	3	4	5	6	7
Grade	10	20	15	15	15	15	10

Questions

- 1) Write a function that takes three integers as input and returns True if those integers don't share a common divisor other than 1, False otherwise.
You may assume that the input will always be positive integers (No need to implement negative checks).

Input	Output
4, 9, 15	True
21, 34, 22	True
5, 15, 370	False
60, 144, 96	False

- 2) Write a Python program that prompts the user to enter an integer n and prints out a list of all Pythagorean triplets with simplified values where hypotenuse is less than n .

A Pythagorean triplet is a triplet of integers, a , b , and c such that $a^2 + b^2 = c^2$. Some of the examples are (3, 4, 5) and (5, 12, 13).

The printed list must consist of Pythagorean triplets represented with tuples, and the numbers of the triplet must be in non-decreasing order in the tuple.

The printed list can be in any order.

The Pythagorean triplets must be simplified, meaning that a , b , and c must not share a common divisor other than 1. Ex: (3, 4, 5) is valid while (6, 8, 10), (9, 12, 15)... are invalid.

Input	Output
30	[(3, 4, 5), (5, 12, 13), (8, 15, 17), (7, 24, 25), (20, 21, 29)]

- 3) The Caesar cipher, also known as the shift cipher, is one of the simplest and most well-known encryption techniques. It was named after Julius Caesar, who is believed to have used this cipher to protect his military messages.

In this method, each letter is shifted n number of places in the alphabet. Example:

PYTHON, $n = 3$

P -> S (P, Q, R, S)

Y -> B (Y, Z, A, B)

T -> W (T, U, V, W)

H -> K (H, I, J, K)

O -> R (O, P, Q, R)

N -> Q (N, O, P, Q)

PYTHON becomes SBWKRQ when $n = 3$.

Write a program that prompts the user to enter a string then an integer, then Caesar ciphers the string by shifting it by the input number. Return the encoded message.

You may assume that the given string will consist of only spaces and lowercase English letters.

Input	Output
"python", 3	"sbwkrq"

- 4) Write a program that takes a list of tuples of length 2 ([(1,2),(3,4)...]) where each tuple represents a person and the elements of a tuple is that person's height and weight in order.
Sort this list by height in non-increasing order. However, people sharing the same height must be sorted by weight in non-increasing order as well. Print the sorted list.

Input	Output
[(10, 20), (40, 30), (30, 15), (40, 35), (50, 20), (40, 25), (50, 50)]	[(50, 50), (50, 20), (40, 35), (40, 30), (40, 25), (30, 15), (10, 20)]

- 5) Write a function that takes two strings and a float. The strings denote temperature units and they can only be "C", "F", "K".

This function must be able to convert given temperature (float) from the first string's unit to second string's unit. You should return the converted temperature.

The conversion formula is as below:

$$C = (F - 32) * 5/9 = K - 273.15$$

Input	Output
"C", "K", 50	323.15
"F", "K", 126	325.372222
"F", "C", 12	-11.111111
"K", "C", 42	-231.14999999

- 6) You have a stock that is increasing in value insanely.

Your stocks increases in value by 1% everyday, meaning that:

If your stock is £100 on day 0, it values $100 * 1.01 = 101$ on day 1, and it values $101 * 1.01 = 102.01$ on day 2.

Your stock increases in value 2% every weekend day.

Your stock increases in value 5% every month.

Your stock increases in value 10% every year.

Write a program that takes a float (initial value of your stock) and an integer (day).

The program must calculate the value of your stock at the given day.

When increasing in value on a day, the stock can increase by only one type of rule, and the maximum increment is applied.

You start from day zero on 31th December night which is coincidentally Sunday too.

Weeks pass every 7 days; months pass every 30 days and years pass every 365 days.

Input	Output
100, 2	102.01
100, 365	17739.706955739555
100, 3650	2.996594789347392e+24

- 7) Write a program that prompts the user to enter a string, print a list of all substrings of the given string.

The printed list can be in any order but it must not contain any duplicate substrings.

Input	Output
"abc"	['a', 'ab', 'abc', 'b', 'bc', 'c']
"abcdefg"	['a', 'ab', 'abc', 'abcd', 'abcde', 'abcdef', 'abcdefg', 'b', 'bc', 'bcd', 'bcde', 'bcdef', 'bcdefg', 'c', 'cd', 'cde', 'cdef', 'cdefg', 'd', 'de', 'def', 'defg', 'e', 'ef', 'efg', 'f', 'fg', 'g']

Submission

Your code will be graded automatically. Therefore, it's important that you follow the submission instructions. You will lose points if your submission does not comply with the submission rules.

First, all of your Python scripts should be collected under a folder name "midterm". Then, you should zip the "midterm" folder and rename it to "<<student id>>". This zip file will be submitted through Moodle. Naming of your solution must follow the convention: answer_<<question no>>.py (e.g., solution script for the 1st question should be named as answer_1.py).

Warnings

This is individual Midterm. All scripts are checked automatically for similarity with other submissions and exercises from previous years. Do not copy codes from the internet or your friends. Make sure you write and submit your own code. Any sign of cheating will be penalized, and you will get -50 points for the midterm, and you will get F grade in case of recurrence in any other work.