

CICS-160 Spring 2023

Assignment 1: Basic Python

Due on Sunday February 19th 2023, by 11:59 PM

Learning Goals:

This assignment is designed to give you an opportunity to check your proficiency with commonly-used Python programming, before we start working with classes and objects. This includes input and output operations, conditionals, loops, functions, and lists. Each of the tasks below require you to write functions. **All programs need to be written in Python 3.x.** You do not need to write a main program, but you can, and you are encouraged to do so in order to test your functions. However, for the Gradescope autograder to be able to correctly process your work, by the time you upload your submission your main program has to be either completely commented out, or completely enclosed in an `"if __name__ == '__main__':"` block, as discussed in class.

Overview

For this project, you will be checking a couple of properties for numbers. The properties both have to do with integers being prime, and with integers being divisible by other integers. This is a common operation at the center of encrypting information as it gets transmitted from one place to another. As such, it is at the center of many communications that have come to be standard in Internet communication.

First, a couple of definitions:

- An integer number is **prime** if it is greater than 1, and if it can only be divided by 1 and itself. For example, 2, 3, 5, and 7 are all prime, but 4, 6, 8, and 9 are not (4, 6 and 8 can all be divided by 2; 9 can be divided by 3).
- Two integers are relatively prime if they do not have any common divisor other than 1. For example, 2 and 9 are **relatively prime**, since 2 can only be divided by 1 and 2, and 9 can only be divided by 1, 3, and 9, so the only common divisor is 1.
 - Notice that for two numbers to be relatively prime it is NOT necessary for either of them to be prime. 4 and 9 are relatively prime, even though neither of them is prime.
 - When you establish a secure connection between two machines, they use relatively prime numbers to encrypt and decrypt messages. So it is really important to know if two numbers are relatively prime.

- As an added effect of these definitions, two prime numbers are always relatively prime, since neither has a divisor other than 1.
 - But if you have a prime number and a number that is not prime, they might or might not be relatively prime when taken together. For example, 3 is prime, but 3 and 9 are not relatively prime, since 3 is a divisor of both 3 and 9.

Program Development- Your Tasks.

Please submit the four functions described below in a single file called assignment1.py. If you want, you can make use of `math.sqrt`, but do not use any other functions from the math library without asking first.

Task 1

Write a function called *is_prime* that receives a positive integer greater than 1 as input, and returns true if the integer is prime, and false otherwise. You do not have to worry about efficiency. The only requirement is that the function returns the correct value. You also do not have to perform error checking on the input.

(25 points)

Task 2

Write a function called *are_relatively_prime* that receives two integers as input, and returns true if the two integers are relatively prime, and false otherwise. This function does not necessarily need to use the function you created in task 1. You do not have to perform error checking on the inputs.

(30 points)

Task 3

Write a function called *primes* that receives a positive integer as input, and returns a list of all prime numbers smaller than or equal to the input number. This function must use the function that you created for task 1. You do not have to perform error checking on the input.

(30 points)

Task 4

Write a function called *prime_decomposition*, that receives as input a positive integer, and returns a list with the prime number decomposition of the number received as input. For example,

`prime_decomposition(2)` returns `[2]`, `prime_decomposition(10)` returns `[2,5]`, and `prime_decomposition(12)` returns `[2,2,3]`. You do not have to perform error checking on the input. (15 points)