EEE 419/591
Fall 2025
Homework 1

Solve the following three problems. Put them all into a single code named hw1.py and submit it to the designated link on Canvas. If you upload a second attempt to Canvas and it renames your file, ignore the renaming.

If you are a beginner in programming, you are encouraged to write the code yourself without the help of AI (except for limited help such as understanding what a greatest common divisor is, how to find it, different methods of checking if a number is prime and similar hints).

Problem 1: Greatest Common Divisor

Write a code that takes a list of integer numbers and gives the greatest common divisor of them all. Do not use the "greatest common divisor" function built-in in any of the python packages.

Your code should expect a list of numbers as an input and print "The GCD is: " followed by the greatest common divisor of the list passed to it by the user. An example of output of your code is as follows.

```
Input a list of integers: [24, 60, 120, 30]

The GCD is: 6
```

Hint: start writing a code that works on a list of 2 numbers.

Problem 2: Prime Number Checker

Write a code that takes a list of integers and outputs, for each number, whether it is prime. Do not use the "prime number checker" function built-in in any of the python packages.

Your code should expect a single list of integers as an input and prints if each number is prime as the following example.

```
Input a list of integers: [31, 17, 81, 28]

31 is a prime

17 is a prime

81 is not a prime

28 is not a prime
```

Problem 3: Reimann-Zeta Function

Write a code that calculates an approximation to the Reimann-Zeta function for a specific value of integer. Your code takes a single integer and outputs an approximate value of the corresponding Reimann-Zeta function of that integer as well as the number of terms your code depends on to calculate that function. Only print up to the first 4 decimal places of the function's value at the given input. Do not use the "Reimann-Zeta" function built-in in any of the python packages.

Recall that the Reimann-Zeta function is given by

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

Since we cannot calculate an infinite sum of terms, you will need to use only a finite summation which is why your code will give an approximate value, no matter how many terms your code computes. It is up to you to decide how many terms your code computes to give good approximation. It is up to you to define how good is "good". Your code should print something as the following example.

```
Input an integer: 2

Zeta(2) = 1.6449 based on 10,000 terms
```