CS-546 Lab 1

An Intro to Node

For this lab, you will be creating and running several functions to practice JavaScript syntax.

For this lab, you will make two files: lab1.js and lab1.test.js and submit them in a zip file that's named LastName\_FirstName.zip. For example: Hill\_Patrick.zip

You **should not** have any folders inside the zip file.

You **must** submit your files with the format specified, named as specified.

lab1.js

In this file, you will update the content of the functions and update the firstName, lastName, and studentId with the appropriate information.

Functions to implement

questionOne(arr);

For your first function, you will calculate if each and every element in the array is a prime number. Your function will return an array that has boolean values corresponding to the original index of the input.  That means that in lab1.test.js, running lab1.questionOne([5, 3, 10]) would return [true, true, false].

To test this function, you will log the result of 5 calls to lab1.questionOne([x, y, z]) with different inputs.

questionTwo(startingNumber, commonRatio, numberOfTerms);

This function will sum a [Geometric SeriesLinks to an external site.](https://www.mathsisfun.com/algebra/sequences-sums-geometric.html)

startingNumber can be positive, negative, decimal but CANNOT be 0. If 0 is passed into the function as the startingNumber your function should return 0.

commonRatio can be positive, negative, decimal but CANNOT be 0. If 0 is passed into the function as the commonRatio your function should return 0.

numberOfTerms can only be a positive whole number greater than 0. If numberOfTerms is <=0 or is a decimal, your function should return NaN

That means that in lab1.test.js, running lab1.questionTwo(2,10,4) would return 2222.

To test this function, you will log the result of 5 calls to lab1.questionTwo(x, y, z) with different inputs.

questionThree(str)

This function will return the number of consonants contained in the value str.

questionFour(fullString, substring)

 For the fourth function, you will calculate how many times a substring occurs in a given string.

For example, calling questionFour("hello world", "o"); should return 2, because the letter o appears two times in the string.

However, you must also factor in a case where there are overlaps! When you call questionFour("Helllllllo, class!", "ll"); should return 3  since "ll" appears 3 times.