Lab 1

Due Jan 26, 2022 by 11:59pm **Points** 100 **Submitting** a file upload
File Types zip

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. The function specifications are listed in the section below.

```
const questionOne = function questionOne(arr) {
    // Implement question 1 here
}

const questionTwo = function questionTwo(num) {
    // Implement question 2 here
}

const questionThree = function questionThree(text) {
    // Implement question 3 here
}

const questionFour = function questionFour(num) {
    // Implement question 4 here
}

module.exports = {
    firstName: "YOUR FIRST NAME",
    lastName: "YOUR LAST NAME",
    studentId: "YOUR STUDENT ID",
    questionOne,
    questionTwo,
    questionThree,
    questionFour
};
```

lab1.test.js

```
const lab1 = require("./lab1");

console.log(lab1.questionOne([1, 2, 3]));
// should output 14

console.log(lab1.questionTwo(7));
// should output 13

console.log(lab1.questionThree("Mr. and Mrs. Dursley, of number four, Privet Drive, were  proud  to  say  t
// should output 196

console.log(lab1.questionFour(10));
// should output 3628800
```

Functions to implement

questionOne(arr)

For your first function, you will calculate the sum of the squares of all numbers in the array and return that result. That means that in `lab1.test.js`, running `lab1.questionOne([5, 3, 10])` would return `134`.

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

```
console.log(lab1.questionOne([5, 3, 10]));
// 134
console.log(lab1.questionOne([2, 1, 2]));
// 9
console.log(lab1.questionOne([5, 10, 9]));
// 206
```

questionTwo(index);

This function should calculate the **Fibonacci (https://en.wikipedia.org/wiki/Fibonacci_number)** that corresponds to the `index` given.

The Fibonacci value of a number is the sum of the previous two Fibonacci values; the Fibonacci of any number less than 1 is 0; the Fibonacci Value of 1 is 1; the Fibonacci value of all other numbers is the sum of the previous two Fibonacci numbers.

Index	Value	Description
0	0	Fibonacci of anything less than 1 is 0
1	1	Fibonacci of 1 is 1
2	1	Fibonacci of 2 is Fibonacci(1) + Fibonacci(0)

Index	Value	Description
3	2	Fibonacci of 3 is Fibonacci(2) + Fibonacci(1)
4	3	Fibonacci of 4 is Fibonacci(3) + Fibonacci(2)
5	5	Fibonacci of 5 is Fibonacci(4) + Fibonacci(3)
6	8	Fibonacci of 6 is Fibonacci(5) + Fibonacci(4)
7	13	Fibonacci of 7 is Fibonacci(6) + Fibonacci(5)
8	21	Fibonacci of 8 is Fibonacci(7) + Fibonacci(6)
9	34	Fibonacci of 9 is Fibonacci(8) + Fibonacci(7)
10	55	Fibonacci of 10 is Fibonacci(9) + Fibonacci(8)
11	89	Fibonacci of 11 is Fibonacci(10) + Fibonacci(9)

And so on.

questionThree(str)

This function will return the number of vowels contained in the value `str`. For the purposes of this exercise, we are not counting `y` as a vowel.

questionFour(num)

This function will return the factorial of the number `num` provided.

The factorial of a number is a simple formula:

```
factorial(n) = n * (n - 1) * (n - 2) ... * 1
```

The factorial of 0 is 1. If `num` is less than `0`, then return `NaN`.

Number	Factorial
-1	NaN
0	1
1	1
2	2
3	6
4	24
5	120

Requirements

1. You will have to write each function
2. You must submit all files, zipped up, not contained in any folders
3. You must not use any npm dependenices in this lab