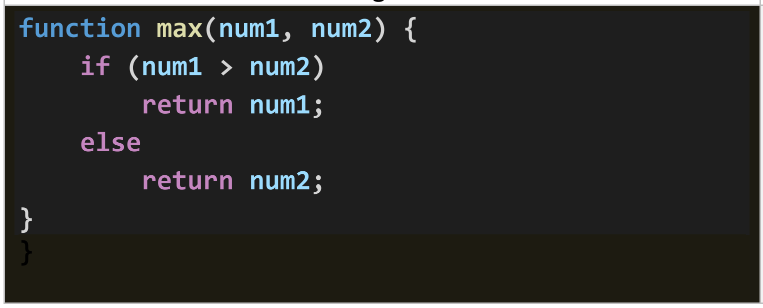
W1D5 – Lab Assignment 5 / Homework

Implement code for the following JavaScript functions, and be sure to use "use strict";

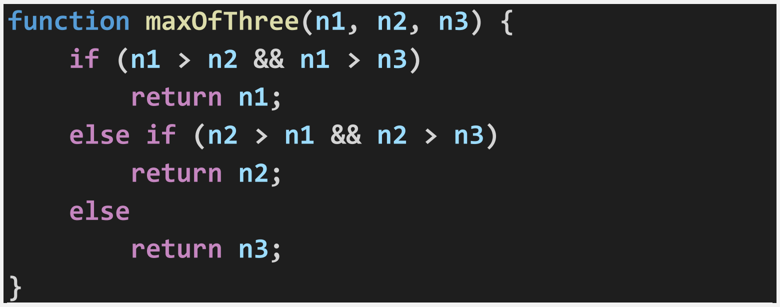
1. Define a function max() that takes two numbers as arguments and returns the largest of them. Use the if-then-else construct available in JavaScript.

Program	Output


`function max(num1, num2) {`
`    if (num1 > num2)`
`        return num1;`
`    else`
`        return num2;`
`}`
`}`
	max(5, 4);
	output: 5
	max(10, 20);
	output: 20

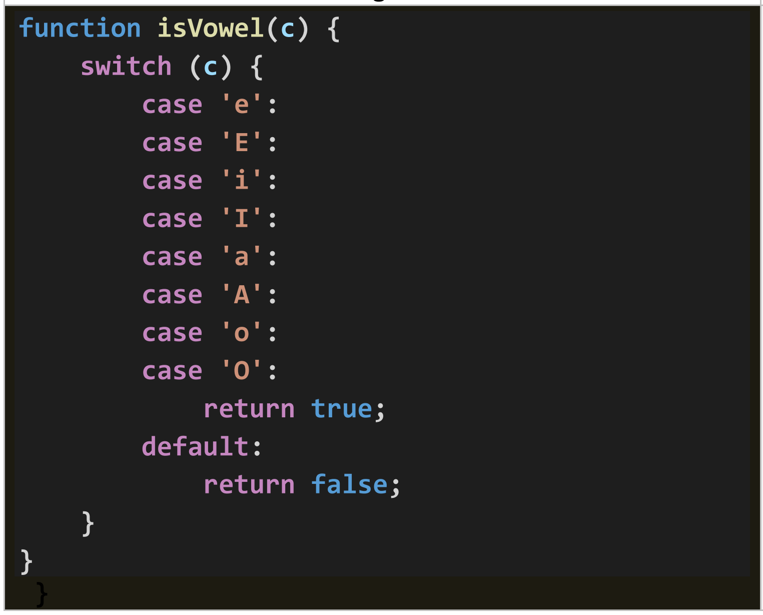
2. Define a function maxOfThree() that takes three numbers as arguments and returns the largest of them.

Program	Output


`function maxOfThree(n1, n2, n3) {`
`    if (n1 > n2 && n1 > n3)`
`        return n1;`
`    else if (n2 > n1 && n2 > n3)`
`        return n2;`
`    else`
`        return n3;`
`}`
	maxOfThree(5, 7, 4);
	output: 7
	maxOfThree(10, 5, 20);
	output: 20

3. Write a function isVowel() that takes a character (i.e. a string of length 1) and returns true if it is a vowel, false otherwise.

<table>
<tr><th>Program</th><th>Output</th></tr>
<tr><td>



```
function isVowel(c) {
    switch (c) {
        case 'e':
        case 'E':
        case 'i':
        case 'I':
        case 'a':
        case 'A':
        case 'o':
        case 'O':
            return true;
        default:
            return false;
    }
}
  }
```

</td><td>

isVowel('a');

output: True

isVowel('b');

output: false

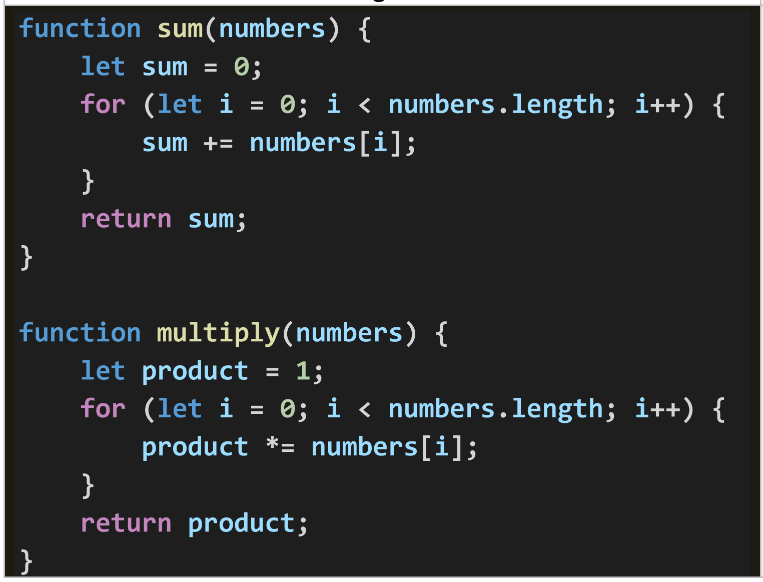
isVowel('E');

output: True

</td></tr>
</table>

4. Define a function sum() and a function multiply() that sums and multiplies (respectively) all the numbers in an input array of numbers. For example, sum([1,2,3,4]) should return 10, and multiply([1,2,3,4]) should return 24. Note/Hint: Do these using Imperative programming approach (i.e. for…loop or while…loop)

<table>
<tr><th>Program</th><th>Output</th></tr>
<tr><td>



```
function sum(numbers) {
    let sum = 0;
    for (let i = 0; i < numbers.length; i++) {
        sum += numbers[i];
    }
    return sum;
}

function multiply(numbers) {
    let product = 1;
    for (let i = 0; i < numbers.length; i++) {
        product *= numbers[i];
    }
    return product;
}
```

</td><td>

sum([1, 2, 3, 4]);

output: 10

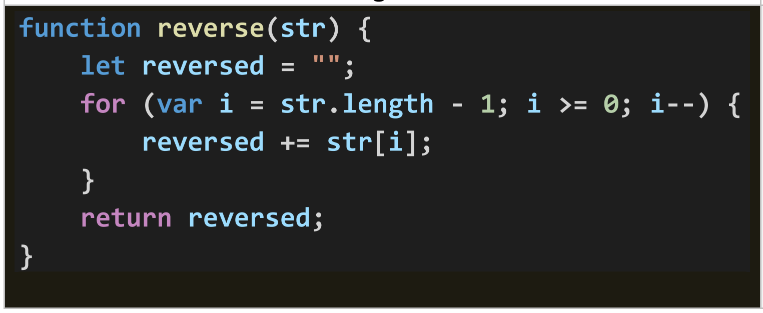
multiply ([1, 2, 3, 4]);

output: 24

</td></tr>
</table>

5. Define a function reverse() that computes the reversal of a string. For example, reverse("jag testar") should return the string "ratset gaj".

Program



```
function reverse(str) {
    let reversed = "";
    for (var i = str.length - 1; i >= 0; i--) {
        reversed += str[i];
    }
    return reversed;
}
```

Output

reverse("jag testar");

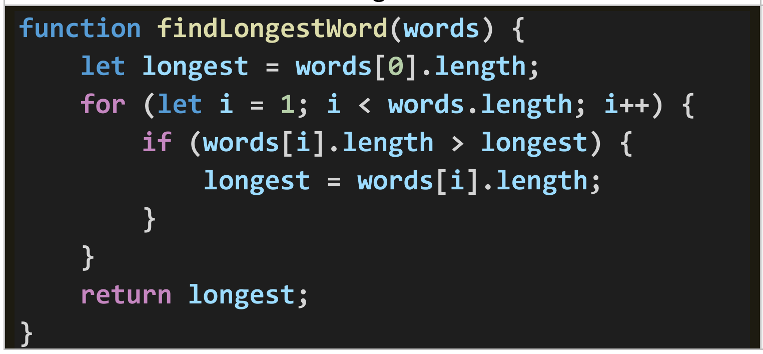
output: ratset gaj

reverse("birhane gebre ");

output: erbeg enahrib

6. Write a function findLongestWord() that takes an array of words and returns the length of the longest one.

Program



```
function findLongestWord(words) {
    let longest = words[0].length;
    for (let i = 1; i < words.length; i++) {
        if (words[i].length > longest) {
            longest = words[i].length;
        }
    }
    return longest;
}
```

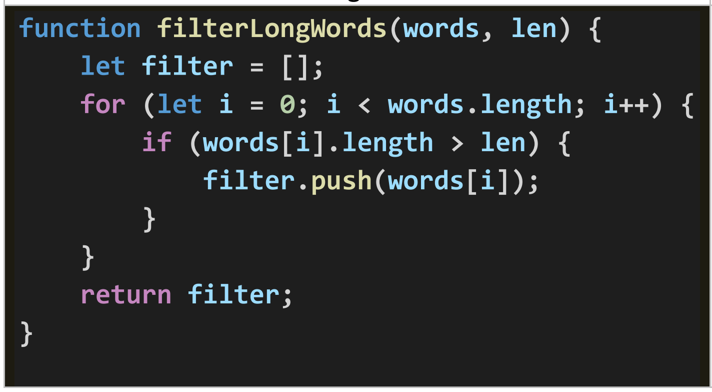
Output

findLongestWord(["hani", "welela", "Birhane", "kalu"])

output: 7

7. Write a function filterLongWords() that takes an array of words and an integer i and returns a new array containing only those words that were longer than i characters.

Program



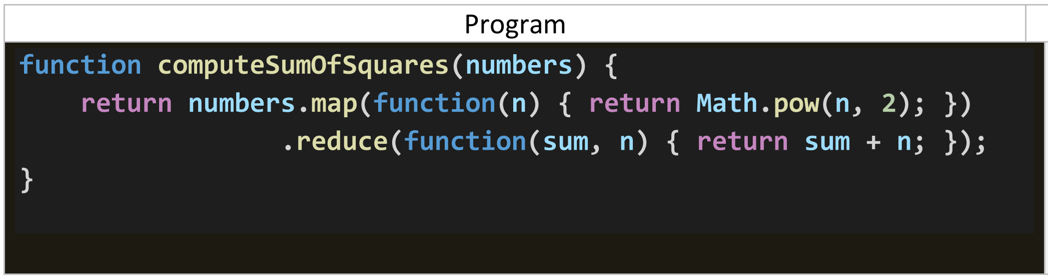
```
function filterLongWords(words, len) {
    let filter = [];
    for (let i = 0; i < words.length; i++) {
        if (words[i].length > len) {
            filter.push(words[i]);
        }
    }
    return filter;
}
```

Output

filterLongWords(["hani", "welela", "Birhane", "kalu"], 4)

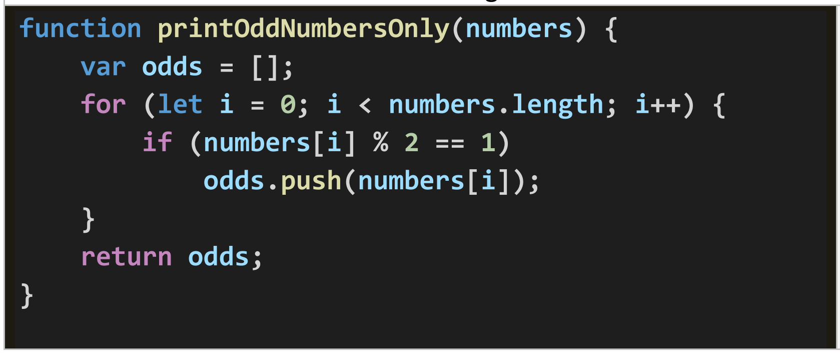
ouput: ["welela", "Birhane"]

8. Write a function named, computeSumOfSquares, that takes as input, an array of numbers and calculates and returns the sum of the squares of each number in the input array. E.g. computeSumOfSquares([1,2,3]) should be computed as $1^2 + 2^2 + 3^2$ = 14. Note: Write your Javascript code without using Imperative programming. i.e. Do NOT use any explicit looping construct; instead use functional programming style/approach.

Program	Output
	computeSumOfSquares ([1, 2, 3]) output: 14

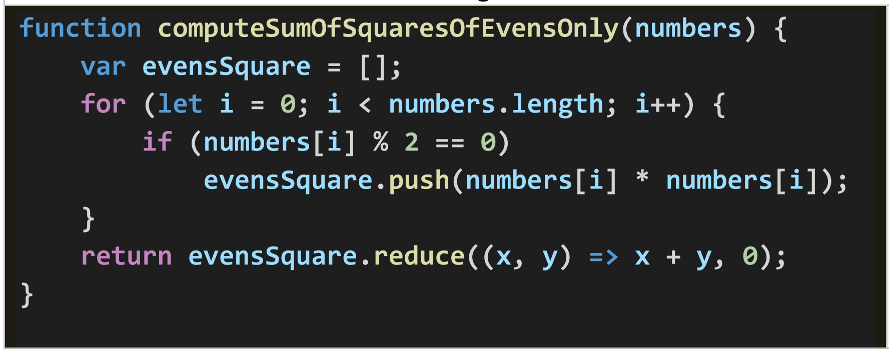
```
function computeSumOfSquares(numbers) {
    return numbers.map(function(n) { return Math.pow(n, 2); })
                  .reduce(function(sum, n) { return sum + n; });
}
```

9. Write a function named, printOddNumbersOnly, that takes as input, an array of integral numbers and it finds and prints only the numbers which are odd.

Program	Output
	printOddNumbersOnly([1, 2, 3, 4, 5, 6]); Output: [1, 3, 5]

```
function printOddNumbersOnly(numbers) {
    var odds = [];
    for (let i = 0; i < numbers.length; i++) {
        if (numbers[i] % 2 == 1)
            odds.push(numbers[i]);
    }
    return odds;
}
```

10. Write a function named, computeSumOfSquaresOfEvensOnly, that takes as input, an array of integral numbers and calculates and returns the sum of the squares of only the even numbers in the input array. E.g. computeSumOfSquaresOfEvensOnly ([1,2,3,4,5]) should be computed as $2^2 + 4^2$ = 20.

Program	Output
	computeSumOfSquaresOfEvensOnly([1, 2, 3, 4, 5]); output: 20

```
function computeSumOfSquaresOfEvensOnly(numbers) {
    var evensSquare = [];
    for (let i = 0; i < numbers.length; i++) {
        if (numbers[i] % 2 == 0)
            evensSquare.push(numbers[i] * numbers[i]);
    }
    return evensSquare.reduce((x, y) => x + y, 0);
}
```

11. Using the Array.reduce(…) function, re-implement your functions, sum(…) and multiply(…) (defined in Problem 4 above) without using Imperative programming. i.e. Do NOT use any explicit looping construct; instead use functional programming style/approach.

Program	Output

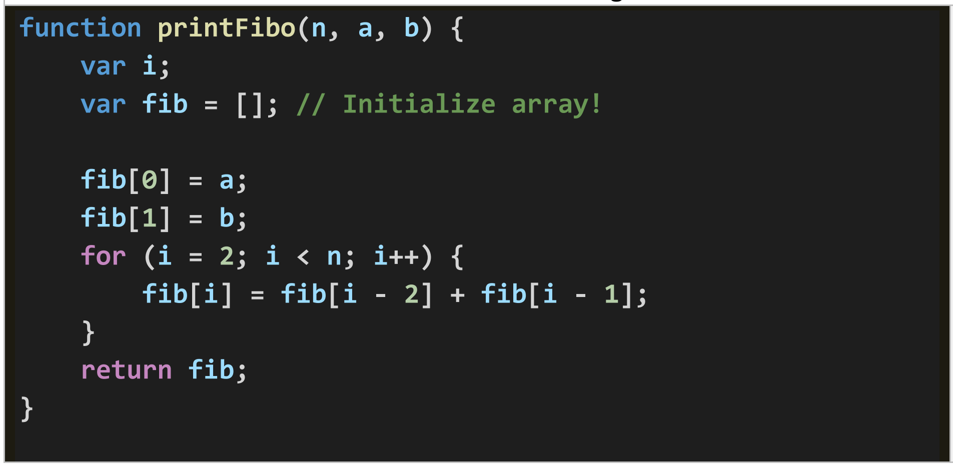
function sum(numbers) {
    return numbers.reduce((x, y) => x + y, 0);
}
 	
function multiply(numbers) {
    return numbers.reduce((x, y) => x * y, 1);
}
	sum([1, 2, 3, 4]);
	output: 10
	
	multiply([1, 2, 3, 4]);
	output: 24

12. Implement Javascript code for a function named, **findSecondBiggest**, which takes as input, an array of numbers and finds and returns the second biggest of the numbers. For example, findSecondBiggest([1,2,3,4,5]) should return 4. And findSecondBiggest([19,9,11,0,12]) should return 12. (Note: Do not use sorting!)

Program	Output

function findSecondBiggest(numbers) {
    var max1 = numbers[0];
    var max2 = -Infinity;
    for (var i = 0; i < numbers.length; i++) {
        if (numbers[i] > max1) {
            max2 = max1;
            max1 = numbers[i];
        } else if (numbers[i] > max2 && numbers[i] !== max1) {
            max2 = numbers[i];
        }
    }
    return max2;
}
	findSecondBiggest(19,9,11,0,12]);
	output: 12
	
	findSecondBiggest(1,2,3,4,5]);
	output: 4

13. Write a function named printFibo, that takes as input, a given length, n, and any two starting numbers a and b, and it prints-out the Fibonacci sequence, e.g. (0, 1, 1, 2, 3, 5, 8, 13, 21, 34,...) of the given length, beginning with a and b. (e.g. printFibo(n=1, a=0, b=1), prints-out: "0", as output; printFibo(n=2, a=0, b=1), prints-out: "0, 1", as output; printFibo(n=3, a=0, b=1), prints-out: "0, 1, 1", as output; printFibo(n=6, a=0, b=1), prints-out: "0, 1, 1, 2, 3, 5", as output; and printFibo(n=10, a=0, b=1), prints-out: "0, 1, 1, 2, 3, 5, 8, 13, 21, 34", as output).

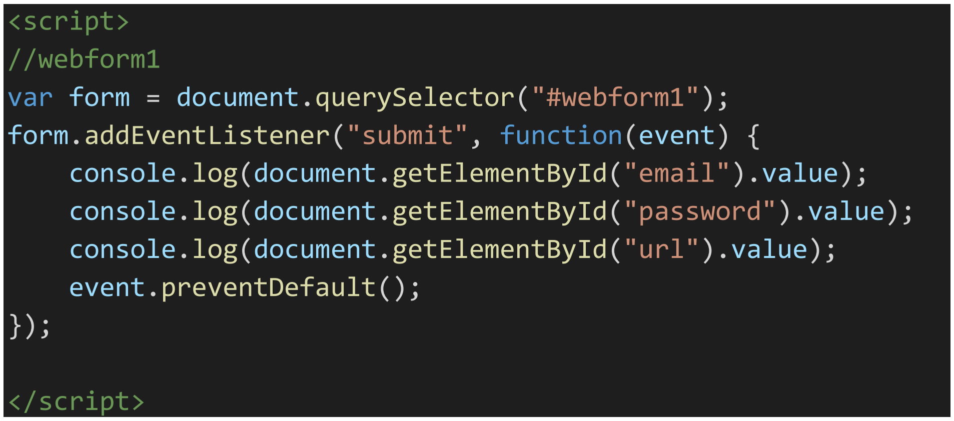
<table>
<tr><th>Program</th><th>Output</th></tr>
<tr><td>


```
function printFibo(n, a, b) {
    var i;
    var fib = []; // Initialize array!

    fib[0] = a;
    fib[1] = b;
    for (i = 2; i < n; i++) {
        fib[i] = fib[i - 2] + fib[i - 1];
    }
    return fib;
}
```
</td><td>printFibo(6, 0, 1));
output: [0, 1, 1, 2, 3, 5]
printFibo(10, 0, 1));
output: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]</td></tr>
</table>

14. Refer to your work on Lab Assignment 4. Add Javascript code to work with your 2 HTML forms as follows:

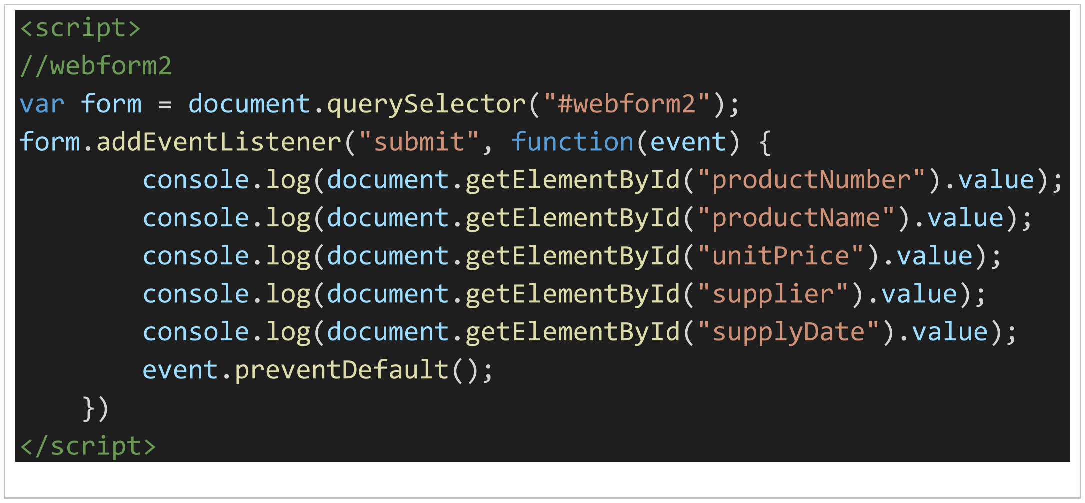
   a. Login Form: Add code such that when the Submit button is clicked, the values entered in the input fields are printed to the Console.



```
<script>
//webform1
var form = document.querySelector("#webform1");
form.addEventListener("submit", function(event) {
    console.log(document.getElementById("email").value);
    console.log(document.getElementById("password").value);
    console.log(document.getElementById("url").value);
    event.preventDefault();
});

</script>
```

b. New Product Form: Add code such that when the Submit button is clicked, the values entered in the input fields are displayed in a pop-up window.



```
<script>
//webform2
var form = document.querySelector("#webform2");
form.addEventListener("submit", function(event) {
        console.log(document.getElementById("productNumber").value);
        console.log(document.getElementById("productName").value);
        console.log(document.getElementById("unitPrice").value);
        console.log(document.getElementById("supplier").value);
        console.log(document.getElementById("supplyDate").value);
        event.preventDefault();
    })
</script>
```

15. Using JavaScript and HTML and CSS, implement a webpage that displays a working, ticking counter Clock, that counts/displays the current Date and time of the browser host, in the format: 2019-11-4 12:16:01

```
<script>
      displayTime();

      function displayTime() {
            let date = new Date();

            let year = date.getFullYear();
            let month = date.getMonth();
            let day = date.getDay();

            let hh = date.getHours();
            let mm = date.getMinutes();
            let ss = date.getSeconds();

            hh = (hh < 10) ? "0" + hh : hh;
            mm = (mm < 10) ? "0" + mm : mm;
            ss = (ss < 10) ? "0" + ss : ss;
            let timeFormated = hh + ":" + mm + ":" + ss;
            let dateFormated = year + "-" + month + "-" + day;
            document.getElementById("date").innerHTML = dateFormated;
            document.getElementById("clock").innerText = timeFormated;
            setTimeout(displayTime, 1000);
      }
</script>
```

Please submit your code as a single zip file attachment to Sakai and also push it to your github repository.

//-- Enjoy! --//