**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.

function max(x, y)

{ if (x > y)

return x;

else

return y; }

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

function maxOfThree(n1, n2, n3) {

if (n1 > n2 && n1 > n3)

return n1;

else if (n2 > n1 && n2 > n3)

return n2;

else return n3; }

1. 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.

function isVowel(x) {

switch (x) {

case 'e':

case 'E':

case 'i':

case 'I':

case 'a':

case 'A':

case 'o':

case 'O':

return true;

default:

return false; }

1. 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)

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; }

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

function reverse(str) {

let reversed = "";

for (let i = str.length - 1; i >= 0; i--) {

reversed += str[i]; }

return reversed; }

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

function findLongestWord(words) {

let max = words[0].length;

for (let i = 1; i < words.length; i++) {

if (words[i].length > max) {

longest = words[i].length; }

} return longest;

1. 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.

function filterLongWords(words, size) {

let newArr = [];

for (let i = 0; i < words.length; i++) {

if (words[i].length > size) {

newArr.push(words[i]); }

} return filter; }

1. 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 12 + 22 +32 = 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.

function computeSumOfSquares(numbers) {

return numbers.map(function(n) { return Math.pow(n, 2); })

.reduce(function(sum, n) { return sum + n; }); }

1. 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.

function printOddNumbersOnly(numbers) {

var oddNums = [];

for (let i = 0; i < numbers.length; i++) {

if (numbers[i] % 2)

oddNums.push(numbers[i]); }

return oddNums; }

1. 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 22 +42 = 20.

function computeSumOfSquaresOfEvensOnly(numbers) {

var evensSquare = [];

let sum=0;

for (let i = 0; i < numbers.length; i++) {

if (numbers[i] % 2 == 0)

evensSquare.push(numbers[i] \* numbers[i]); }

for (let i = 0; i < evensSquare.length; i++) {

sum+=evensSquare[i];

}

return sum;

}

1. 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.

function sum(numbers) {

return numbers.reduce((x, y) => x + y, 0); }

function multiply(numbers) {

return numbers.reduce((x, y) => x \* y, 1); }

1. 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!)

function findSecondBiggest(numbers) {

var biggeset = numbers[0];

var second = -Infinity;

for (var i = 0; i < numbers.length; i++) {

if (numbers[i] > biggeset) {

second = biggeset;

biggeset = numbers[i]; }

else if (numbers[i] > second && numbers[i] !== biggeset) {

second = numbers[i]; } }

return second; }

1. 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).

function printFibo(n,a,b){

var arr=[];

arr.push(a);

arr.push(b);

for(let i=2;i< n;i++){

arr.push(arr[i-1]+arr[i-2]);

}

return arr;

}

1. Refer to your work on Lab Assignment 4. Add Javascript code to work with your 2 HTML forms as follows:
   1. Login Form: Add code such that when the Submit button is clicked, the values entered in the input fields are printed to the Console.
   2. 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.
2. 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

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

**//-- Enjoy! --//**