**ADVANCED JAVASCRIPT PRACTICAL ASSIGNMENTS**

1. Write a JavaScript program to convert a given string into an array of words.

* string\_to\_array = function (str) {

return str.trim().split(" ");

};

console.log(string\_to\_array("Dhruv khambhata"));

1. Write a JavaScript program to test a value, x, against a predicate function. If true, return fn(x). Else, return x.

* <script>

// Function that we have to return

function fn(x) {

let fact = 1;

for (i = 1; i <= x; i++) {

fact \*= i;

}

return fact;

}

// Predicate function checking even value

function isEven(x) {

return (x % 2 === 0);

}

// Function returning fn(x) if predicate

// function return true else return x

function test(x) {

let e = isEven(x);

if (e) {

return fn(x);

} else

return x;

}

// Printing return value

let ans = test(6);

console.log(`This function returns ${ans}`);

</script>

1. Write a JavaScript program to reduce a given Array-like into a value hash (keyed data store).

* <script>

// An array of keys

var keys = [1, 2, 3];

// An array of values

var values = ["GeeksforGeeks", "Computer", "Science"];

// Object created

var obj = {};

// Using loop to insert key

// value in Object

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

obj[keys[i]] = values[i];

}

// Printing object

for (var key of Object.keys(obj)) {

document.write(key + " => " + obj[key] + "</br>")

}

</script>

1. Write a JavaScript program to convert a float-point arithmetic to the Decimal mark form and It will make a comma separated string from a number.

* const toDecimalMark = num => num.toLocaleString('en-US');

console.log(toDecimalMark(12305030388.9087));

console.log(toDecimalMark(123.2264))

console.log(toDecimalMark(-100.10))

1. **Write a JavaScript program to create an object with the same keys as the provided object and values generated by running the provided function for each value.**

* // javascript code demonstrating a simple object

let school = {

name: 'Vivekananda School',

location : 'Delhi',

established : '1971',

displayInfo : function(){

console.log(`${school.name} was established

in ${school.established} at ${school.location}`);

}

}

school.displayInfo();

1. **Write a JavaScript program to replace all but the last number of characters with the specified mask character.**

* <script>

// Function for masking the character

function MaskCharacter(str, mask, n = 1) {

// Slice the string and replace with

// mask then add remaining string

return ('' + str).slice(0, -n)

.replace(/./g, mask)

+ ('' + str).slice(-n);

}

// Number that we want to mask

var num = 12345678;

// Convert number into string

var str = num.toString();

// Call function without giving value of n

console.log(MaskCharacter(str, '#'));

// Call function with value of n

console.log(MaskCharacter(str, '#', 2));

</script>

1. **Write a JavaScript program to filter out all values from an array for which the comparator function does not return true.**

* **const differenceWith = (arr, val, comp) => arr.filter(a => val.findIndex(b => comp(a, b)) === -1);**

**console.log(differenceWith([1, 1.2, 1.5, 3, 0], [1.9, 3, 0], (a, b) => Math.round(a) === Math.round(b)));**

1. **Write a JavaScript program to compute the new ratings between two or more opponents using the Elo rating system. It takes an array of pre-ratings and returns an array containing post-ratings. The array should be ordered from best performer to worst performer (winner -> loser).**

1. **Write a JavaScript program to split values into two groups according to a predicate function, which specifies which group an element in the input collection belongs to.**

* **const bifurcateBy = (arr, fn) =>**

**arr.reduce((acc, val, i) => (acc[fn(val, i) ? 0 : 1].push(val), acc), [[], []]);**

**console.log(bifurcateBy(['beep', 'boop', 'foo', 'bar'], x => x[0] === 'b'));**

1. **Write a JavaScript program to create a function that invokes fn with a given context, optionally adding any additional supplied parameters to the beginning of the arguments**

* **const bind = (fn, context, ...args) =>**

**function() {**

**return fn.apply(context, args.concat(...arguments));**

**};**

**function greet(greeting, punctuation) {**

**return greeting + ' ' + this.user + punctuation;**

**}**

**const freddy = { user: 'Morning' };**

**const freddyBound = bind(greet, freddy);**

**console.log(freddyBound('Good', '!'));**

1. **Write a JavaScript program to perform a deep comparison between two values to determine if they are equivalent.**

* **const equals = (a, b) => {**

**if (a === b) return true;**

**if (a instanceof Date && b instanceof Date) return a.getTime() === b.getTime();**

**if (!a || !b || (typeof a !== 'object' && typeof b !== 'object')) return a === b;**

**if (a === null || a === undefined || b === null || b === undefined) return false;**

**if (a.prototype !== b.prototype) return false;**

**let keys = Object.keys(a);**

**if (keys.length !== Object.keys(b).length) return false;**

**return keys.every(k => equals(a[k], b[k]));**

**};**

**console.log(equals({ a: [2, { e: 3 }], b: [4], c: 'foo' }, { a: [2, { e: 3 }], b: [4], c: 'foo' }));**

1. **Write a JavaScript program to get an array of function property names from own (and optionally inherited) enumerable properties of an object.**

* **const functions = (obj, inherited = false) =>**

**(inherited**

**? [...Object.keys(obj), ...Object.keys(Object.getPrototypeOf(obj))]**

**: Object.keys(obj)**

**).filter(key => typeof obj[key] === 'function');**

**function Foo() {**

**this.a = () => 1;**

**this.b = () => 2;**

**}**

**Foo.prototype.c = () => 3;**

**console.log(functions(new Foo()));**

**console.log(functions(new Foo(), true));**

1. **Write a JavaScript program to get the sum of a given array, after mapping each element to a value using the provided function.**

* **const sumBy = (arr, fn) =>**

**arr.map(typeof fn === 'function' ? fn : val => val[fn]).reduce((acc, val) => acc + val, 0);**

**console.log(sumBy([{ n: 4 }, { n: 2 }, { n: 8 }, { n: 6 }], o => o.n));**

**console.log(sumBy([{ n: -4 }, { n: -2 }, { n: 8 }, { n: 6 }], 'n'));**

1. **Write a JavaScript program to get a random number in the specified range.**

* **const randomIntegerInRange = (min, max) => Math.floor(Math.random() \* (max - min + 1)) + min;**

**console.log(randomIntegerInRange(0, 5));**

**console.log(randomIntegerInRange(2, 5));**

**console.log(randomIntegerInRange(5, -5));**

**console.log(randomIntegerInRange(-2, -7));**

1. **Write a JavaScript program to get a random integer in the specified range.**

* **const randomIntegerInRange = (min, max) => Math.floor(Math.random() \* (max - min + 1)) + min;**

**console.log(randomIntegerInRange(0, 5));**

**console.log(randomIntegerInRange(2, 5));**

**console.log(randomIntegerInRange(5, -5));**

**console.log(randomIntegerInRange(-2, -7));**

1. **Write a JavaScript program to create a function that accepts up to n arguments, ignoring any additional arguments.**

* **const uncurry = (fn, n = 1) => (...args) => {**

**const next = acc => args => args.reduce((x, y) => x(y), acc);**

**if (n > args.length) throw new RangeError('Arguments too few!');**

**return next(fn)(args.slice(0, n));**

**};**

**const add = x => y => z => x + y + z;**

**const uncurriedAdd = uncurry(add, 3);**

**console.log(uncurriedAdd(1, 2, 3));**

1. **Write a JavaScript program to decode a string of data which has been encoded using base-64 encoding.**

* **<!DOCTYPE html>**

**<html>**

**<head>**

**<title>**

**JavaScript |**

**encode/decode a string to Base64.**

**</title>**

**</head>**

**<body style="text-align:center;">**

**<h1 style="color:green;">**

**Valorant </h1>**

**<p id="GFG\_UP">**

**</p>**

**<button onclick="valo()">**

**Encode**

**</button>**

**<p id="GFG\_DOWN"**

**style="color:green;">**

**</p>**

**<script>**

**var str = "This is valorant";**

**var up = document.getElementById("GFG\_UP");**

**var down = document.getElementById("GFG\_DOWN");**

**up.innerHTML = "Str = '" + str + "'";**

**Function Valo() {**

**down.innerHTML = window.btoa(str);**

**}**

**</script>**

**</body>**

**</html>**

1. **Write a JavaScript program to remove elements in an array until the passed function returns true. Returns the remaining elements in the array.**

* **const dropWhile = (arr, func) => {**

**while (arr.length > 0 && !func(arr[0])) arr = arr.slice(1);**

**return arr;**

**};**

**console.log(dropWhile([1, 2, 3, 4], n => n >= 3));**

1. **Write a JavaScript program to remove elements from the end of an array until the passed function returns true. Returns the remaining elements in the array.**

* **const dropWhile = (arr, func) => {**

**while (arr.length > 0 && !func(arr[0])) arr = arr.slice(1);**

**return arr;**

**};**

**console.log(dropWhile([1, 2, 3, 4], n => n >= 3));**

1. **Write a JavaScript program to remove elements from an array for which the given function returns false.**

* **const remove = (arr, func) =>**

**Array.isArray(arr)**

**? arr.filter(func).reduce((acc, val) => {**

**arr.splice(arr.indexOf(val), 1);**

**return acc.concat(val);**

**}, [])**

**: [];**

**console.log(remove([1, 2, 3, 4], n => n % 2 === 0));**

1. **Write a JavaScript program to log the name of a function.**

* **function abc() {**

**console.log( arguments.callee.name );**

**}**

**abc();**

1. **Write a JavaScript program to get a boolean determining if the passed value is primitive or not.**

* **const isPrimitive = val => !['object', 'function'].includes(typeof val) || val === null;**

**console.log(isPrimitive(null));**

**console.log(isPrimitive(50));**

**console.log(isPrimitive('Hello!'));**

**console.log(isPrimitive(false));**

**console.log(isPrimitive(Symbol()));**

**console.log(isPrimitive([]));**

1. **Write a JavaScript program to check if the provided integer is a prime number is not.**

* **const isPrime = num => {**

**const boundary = Math.floor(Math.sqrt(num));**

**for (var i = 2; i <= boundary; i++) if (num % i === 0) return false;**

**return num >= 2;**

**};**

**console.log(isPrime(11));**

**console.log(isPrime(17));**

**console.log(isPrime(8));**

1. **Write a JavaScript program to group the elements into two arrays, depending on the provided function's truthiness for each element.**

* **const partition = (arr, fn) =>**

**arr.reduce(**

**(acc, val, i, arr) => {**

**acc[fn(val, i, arr) ? 0 : 1].push(val);**

**return acc;**

**},**

**[[], []]**

**);**

**const users = [{ user: 'barney', age: 36, active: false }, { user: 'fred', age: 40, active: true }];**

**partition(users, o => o.active);**

**console.log(users);**

1. **Write a JavaScript program to create a function that invokes fn with partials appended to the arguments it receives.**

* **const partialRight = (fn, ...partials) => (...args) => fn(...args, ...partials);**

**const greet = (greeting, name) => greeting + ' ' + name + '!';**

**const greetJohn = partialRight(greet, 'John');**

**console.log(greetJohn('Hello'));**

1. **Write a JavaScript program to chunk an array into smaller arrays of a specified size.**

* **const chunk = (arr, size) =>**

**Array.from({ length: Math.ceil(arr.length / size) }, (v, i) =>**

**arr.slice(i \* size, i \* size + size)**

**);**

**console.log(chunk([1, 2, 3, 4, 5], 2));**

1. **Write a JavaScript program to clamp number within the inclusive range specified by the given boundary values a and b.**

* **const clampNumber = (num, a, b) => Math.max(Math.min(num, Math.max(a, b)), Math.min(a, b));**

**console.log(clampNumber(2, 3, 5));**

**console.log(clampNumber(1, -1, -5));**

1. **Write a JavaScript program that Assigns default values for all properties in an object that are undefined.**

* **const defaults = (obj, ...defs) => Object.assign({}, obj, ...defs.reverse(), obj);**

**console.log(defaults({ a: 1 }, { b: 2 }, { b: 6 }, { a: 3 }));**

1. **Write a JavaScript program to deep flatten an array.**

* **var flatten = function(a, shallow,r){**

**if(!r){ r = []}**

**if (shallow) {**

**return r.concat.apply(r,a);**

**}**

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

**if(a[i].constructor == Array){**

**flatten(a[i],shallow,r);**

**}else{**

**r.push(a[i]);**

**}**

**}**

**return r;**

**}**

**console.log(flatten([1, [2], [3, [[4]]],[5,6]]));**

**console.log(flatten([1, [2], [3, [[4]]],[5,6]], true));**

1. **Write a JavaScript program to run a function in a separate thread by using a Web Worker, allowing long running functions to not block the UI.**

* **const runAsync = fn => {**

**const worker = new Worker(**

**URL.createObjectURL(newBlob([`postMessage((${fn})());`]), {**

**type: 'application/javascript; charset=utf-8'**

**})**

**);**

**return new Promise((res, rej) => {**

**worker.onmessage = ({ data }) => {**

**res(data), worker.terminate();**

**};**

**worker.onerror = err => {**

**rej(err), worker.terminate();**

**};**

**});**

**};**

**const longRunningFunction = () => {**

**let result = 0;**

**for (let i = 0; i < 1000; i++) {**

**for (let j = 0; j < 700; j++) {**

**for (let k = 0; k < 300; k++) {**

**result = result + i + j + k;**

**}**

**}**

**}**

**return result;**

**};**

**/\***

**\*/**

**runAsync(longRunningFunction).then(console.log); // 209685000000**

**runAsync(() => 10 \*\* 3).then(console.log); // 1000**

**let outsideVariable = 50;**

**runAsync(() => typeof outsideVariable).then(console.log); // 'undefined'**

1. **Write a JavaScript program to display the current day and time in the following format.**

* **var today = new Date();**

**var day = today.getDay();**

**var daylist = ["Sunday","Monday","Tuesday","Wednesday ","Thursday","Friday","Saturday"];**

**console.log("Today is : " + daylist[day] + ".");**

**var hour = today.getHours();**

**var minute = today.getMinutes();**

**var second = today.getSeconds();**

**var prepand = (hour >= 12)? " PM ":" AM ";**

**hour = (hour >= 12)? hour - 12: hour;**

**if (hour===0 && prepand===' PM ')**

**{**

**if (minute===0 && second===0)**

**{**

**hour=12;**

**prepand=' Noon';**

**}**

**else**

**{**

**hour=12;**

**prepand=' PM';**

**}**

**}**

**if (hour===0 && prepand===' AM ')**

**{**

**if (minute===0 && second===0)**

**{**

**hour=12;**

**prepand=' Midnight';**

**}**

**else**

**{**

**hour=12;**

**prepand=' AM';**

**}**

**}**

**console.log("Current Time : "+hour + prepand + " : " + minute + " : " + second);**

1. **Write a JavaScript program to print the contents of the current window.**

* **function print\_current\_page()**

**{**

**window.print();**

**}**

1. **Write a JavaScript program to get the current date. Expected Output : mm-dd-yyyy, mm/dd/yyyy or dd-mm-yyyy, dd/mm/yyyy**

* **var today = new Date();**

**var dd = today.getDate();**

**var mm = today.getMonth()+1;**

**var yyyy = today.getFullYear();**

**if(dd<10)**

**{**

**dd='0'+dd;**

**}**

**if(mm<10)**

**{**

**mm='0'+mm;**

**}**

**today = mm+'-'+dd+'-'+yyyy;**

**console.log(today);**

**today = mm+'/'+dd+'/'+yyyy;**

**console.log(today);**

1. **Write a JavaScript program to find the area of a triangle where lengths of the three of its sides are 5, 6, 7.**

* **var side1 = 5;**

**var side2 = 6;**

**var side3 = 7;**

**var perimeter = (side1 + side2 + side3)/2;**

**var area = Math.sqrt(perimeter\*((perimeter-side1)\*(perimeter-side2)(perimeter-side3)));**

**console.log(area);**

1. **Write a JavaScript program to determine whether a given year is a leap year in the Gregorian calendar**.

* **function leapyear(year)**

**{**

**return (year % 100 === 0) ? (year % 400 === 0) : (year %4 === 0);**

**}**

**console.log(leapyear(2016));**

**console.log(leapyear(2000));**

**console.log(leapyear(1700));**

**console.log(leapyear(1800));**

**console.log(leapyear(100));**

1. **Write a JavaScript program to find 1st January is being a Sunday between 2014 and 2050.**

* **console.log('--------------------');**

**for (var year = 2014; year <= 2050; year++)**

**{**

**var d = new Date(year, 0, 1);**

**if ( d.getDay() === 0 )**

**console.log("1st January is being a Sunday "+year);**

**}**

**console.log('--------------------');**

1. **Write a JavaScript program where the program takes a random integer between 1 to 10, the user is then prompted to input a guess number. If the user input matches with guess number, the program will display a message "Good Work" otherwise display a message "Not matched".**

* **// Get a random integer from 1 to 10 inclusive**

**const num = Math.ceil(Math.random() \* 10);**

**console.log(num);**

**const gnum = prompt('Guess the number between 1 and 10inclusive');**

**if (gnum == num)**

**console.log('Matched');**

**else**

**console.log('Not matched, the number was '+gnum);**

1. **Write a JavaScript program to calculate days left until next Christmas.**

* **today=new Date();**

**var cmas=new Date(today.getFullYear(), 11, 25);**

**if (today.getMonth()==11 && today.getDate()>25)**

**{**

**cmas.setFullYear(cmas.getFullYear()+1);**

**}**

**var one\_day=1000\*60\*60\*24;**

**console.log(Math.ceil((cmas.getTime()-today.getTime())/(ne\_day))+**

**" days left until Christmas!");**

1. **Write a JavaScript program to calculate multiplication and division of two numbers (input from user).**

* **function multiplyBy()**

**{**

**num1 = document.getElementById("firstNumber").value;**

**num2 = document.getElementById("secondNumber").value;**

**document.getElementById("result").innerHTML =**

**num1 \* num2;**

**}**

**function divideBy()**

**{**

**num1 = document.getElementById("firstNumber").value;**

**num2 = document.getElementById("secondNumber").value;**

**document.getElementById("result").innerHTML = num1 /**

**num2;**

**}**

1. **Write a JavaScript program that accept two integers and display the larger.**

* **var num1, num2;**

**num1 = window.prompt("Input the First integer", "0");**

**num2 = window.prompt("Input the second integer", "0");**

**if(parseInt(num1, 10) > parseInt(num2, 10))**

**{**

**console.log("The larger of "+ num1+ " and "+ num2+ " is "+ num1+ ".");**

**}**

**else**

**if(parseInt(num2, 10) > parseInt(num1, 10))**

**{**

**console.log("The larger of "+ num1+" and "+ num2+ " is "+ num2+ ".");**

**}**

**else**

**{**

**console.log("The values "+ num1+ " and "+num2+ " are equal.");**

**}**

1. **Write a JavaScript conditional statement to find the sign of product of three numbers. Display an alert box with the specified sign. Sample numbers : 3, -7, 2 Output : The sign is -**

* **var x=3;**

**var y=-7;**

**var z=2;**

**if (x>0 && y>0 && z>0)**

**{**

**alert("The sign is +");**

**}**

**else if (x<0 && y<0 && z<0)**

**{**

**console.log("The sign is -");**

**}**

**else if (x>0 && y<0 && z<0)**

**{**

**console.log("The sign is +");**

**}**

**else if (x<0 && y>0 && z<0)**

**{**

**console.log("The sign is +");**

**}**

**else**

**{**

**console.log("The sign is -");**

**}**

1. **Write a JavaScript conditional statement to sort three numbers. Display an alert box to show the result. Sample numbers : 0, -1, 4 Output : 4, 0, -1**

* **var x= 0;**

**var y=-1;**

**var z= 4;**

**if (x>y && x>z)**

**{**

**if (y>z)**

**{**

**console.log(x + ", " + y + ", " +z);**

**}**

**else**

**{**

**console.log(x + ", " + z + ", " +y);**

**}**

**}**

**else if (y>x && y >z)**

**{**

**if (x>z)**

**{**

**console.log(y + ", " + x + ", " +z);**

**}**

**else**

**{**

**console.log(y + ", " + z + ", " +x);**

**}**

**}**

**else if (z>x && z>y)**

**{**

**if (x>y)**

**{**

**console.log(z + ", " + x + ", " +y);**

**}**

**else**

**{**

**console.log(z + ", " + y + ", " +x);**

**}**

**}**

1. **Write a JavaScript conditional statement to find the largest of five numbers. Display an alert box to show the result. Sample numbers : -5, -2, -6, 0, -1 Output : 0**

* **a=-5;**

**b=-2;**

**c=-6;**

**d= 0;**

**f=-1;**

**if (a>b && a>c && a>d && a>f)**

**{**

**console.log(a);**

**}**

**else if (b>a && b>c && b>d && b>f)**

**{**

**console.log(b);**

**}**

**else if (c>a && c>b && c>d && c>f)**

**{**

**console.log(c);**

**}**

**else if (d>a && d>c && d>b && d>f)**

**{**

**console.log(d);**

**}**

**else**

**{**

**console.log(f);**

**}**

1. **Write a JavaScript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen. Sample Output : "0 is even" "1 is odd" "2 is even"**

* **for (var x=0; x<=15; x++) {**

**if (x === 0) {**

**console.log(x + " is even");**

**}**

**else if (x % 2 === 0) {**

**console.log(x + " is even");**

**}**

**else {**

**console.log(x + " is odd");**

**}**

**}**

1. **Write a JavaScript program which iterates the integers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".**

* **for ( var i = 1; i <= 100; i++ )**

**{**

**if ( i%3 === 0 && i%5 === 0 )**

**{**

**console.log( i + " FizzBuzz" );**

**}**

**else if ( i%3 === 0 )**

**{**

**console.log(i+ " Fizz" );**

**}**

**else if ( i%5 === 0 )**

**{**

**console.log(i+ " Buzz" );**

**}**

**else**

**{**

**console.log(i);**

**}**

**}**

1. **According to Wikipedia a happy number is defined by the following process : "Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers (or sad numbers)". Write a JavaScript program to find and print the first 5 happy numbers.**

* **function happy\_number(num)**

**{**

**var m, n ;**

**var c = [] ;**

**while(num != 1 && c[num] !== true)**

**{**

**c[num] = true ;**

**m = 0 ;**

**while (num > 0) {**

**n = num % 10 ;**

**m += n \* n ;**

**num = (num - n) / 10 ;**

**}**

**num = m ;**

**}**

**return (num == 1) ;**

**}**

**var cnt = 5;**

**var num = 1;**

**var f5 = '';**

**while(cnt-- > 0)**

**{**

**while(!happy\_number(num))**

**num++ ;**

**f5 = f5+(num + ", ") ;**

**num++ ;**

**}**

**console.log('First 5 happy numbers are : '+f5);**

1. **Write a JavaScript program to find the Armstrong numbers of 3 digits. Note : An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since 3\*3 + 73 + 1\*3 = 371.**

* **function three\_digit\_armstrong\_number()**

**{**

**for (var i = 1; i < 10; ++i)**

**{**

**for (var j = 0; j < 10; ++j)**

**{**

**for (var k = 0; k < 10; ++k)**

**{**

**var pow = (Math.pow(i,3) + Math.pow(j,3) +Math.pow(k,3));**

**var plus = (i \* 100 + j \* 10 + k);**

**if (pow == plus)**

**{**

**console.log(pow);**

**}**

**}**

**}**

**}**

**}**

**three\_digit\_armstrong\_number();**

1. **Write a JavaScript program to construct the following pattern, using a nested for loop.**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

* **var x,y,chr;**

**for(x=1; x <=6; x++)**

**{**

**for (y=1; y < x; y++)**

**{**

**chr=chr+("\*");**

**}**

**console.log(chr);**

**chr='';**

**}**

1. **Write a JavaScript program to compute the greatest common divisor (GCD) of two positive integers.**

* **var a = 2154; //First number**

**var b = 458; //Second number**

**var gcd;**

**while (a!=b)**

**{**

**if (a>b)**

**{**

**a = a -b;**

**}**

**else**

**{**

**b = b - a;**

**}**

**}**

**gcd = a;**

**console.log(gcd);**

1. **Write a JavaScript program to sum the multiples of 3 and 5 under 1000.**

* **var sum = 0;**

**for (var x = 0; x < 1000; x++)**

**{**

**if (x % 3 === 0 || x % 5 === 0)**

**{**

**sum += x;**

**}**

**}**

**console.log(sum);**

1. **Write a JavaScript function to find longest substring in a given a string without repeating characters.**

**function longest\_substring\_without\_repeating\_characters(input) {**

**var chars = input.split('');**

**var curr\_char;**

**var str = "";**

**var longest\_string = "";**

**var hash = {};**

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

**curr\_char = chars[i];**

**if (!hash[chars[i]])**

**{**

**str += curr\_char;**

**hash[chars[i]] = {index:i};**

**}**

**else**

**{**

**if(longest\_string.length <= str.length)**

**{**

**longest\_string = str;**

**}**

**var prev\_dupeIndex = hash[curr\_char].index;**

**var str\_FromPrevDupe = input.substring(prev\_dupeIndex + 1, i);**

**str = str\_FromPrevDupe + curr\_char;**

**hash = {};**

**for (var j = prev\_dupeIndex + 1; j <= i; j++) {**

**hash[input.charAt(j)] = {index:j};**

**}**

**}**

**}**

**return longest\_string.length > str.length ? longest\_string : str;**

**}**

**console.log(longest\_substring\_without\_repeating\_characters("google.com"));**

**console.log(longest\_substring\_without\_repeating\_characters("example.com"));**

1. **Write a JavaScript function which will take an array of numbers stored and find the second lowest and second greatest numbers, respectively. Sample array : [1,2,3,4,5] Expected Output : 2,4**

**unction Second\_Greatest\_Lowest(arr\_num)**

**{**

**arr\_num.sort(function(x,y)**

**{**

**return x-y;**

**});**

**var uniqa = [arr\_num[0]];**

**var result = [];**

**for(var j=1; j < arr\_num.length; j++)**

**{**

**if(arr\_num[j-1] !== arr\_num[j])**

**{**

**uniqa.push(arr\_num[j]);**

**}**

**}**

**result.push(uniqa[1],uniqa[uniqa.length-2]);**

**return result.join(',');**

**}**

**console.log(Second\_Greatest\_Lowest([1,2,3,4,5]));**

1. **Write a JavaScript function to get the number of occurrences of each letter in specified string.**

**function Char\_Counts(str1) {**

**var uchars = {};**

**str1.replace(/\S/g, function(l){uchars[l] = (isNaN(uchars[l]) ? 1 : uchars[l] + 1);});**

**return uchars;**

**}**

**console.log(Char\_Counts("The quick brown fox jumps over the lazy dog"));**