1)Remove duplicate value in array :

let arr=[5,4,8,9,5,2,3,6]

function Removeduplicate(arr) {

let result = [];

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

if(result.indexOf(arr[i]) ===-1) {

result.push(arr[i])

}

}

return result;

}

console.log(Removeduplicate(arr));

3) Reversing letters and word :

let a ="string"

let b =a.split("").reverse().join("")

console.log(b)

4) string contains only digits :

let string=("14321")

function onlyDigits(s) {

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

const d = s.charCodeAt(i);

if (d < 48 || d > 57) return false

}

return true

}

console.log(onlyDigits(string));

5) Counting Vowels and Consonants :

function VowcountsAndConscounts(str) {

let vowCount = 0;

let consCount = 0;

str = str.toLowerCase();

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

let y = str[i];

if (y == "a" || y == "e" || y == "i" || y == "o" || y == "u") {

vowCount++;

} else if (y >= "a" && y <= "z") {

consCount++;

}

}

return { vowels: vowCount, consonants: consCount };

}

let result = VowcountsAndConscounts("Technical class");

console.log(result);

6) Counting occurrences of a certain character:

let a = "Hello world";

let b = "l";

count = 0;

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

{

if(a.charAt(i) === b)

{

count++;

}

}

8) Removing white spaces from a string :

let str = " Hello World ";

str = str.split(" ").join("");

console.log(str);

9) Joining multiple strings with a delimiter :

const strings = ["data", "structure", "class"];

const symbol = "+ ";

const result = strings.join(symbol);

console.log(result);

11) Check weather the palindrome or not :

let str=prompt("Enter the string")

function is palindrome(str){

let Left=0;

let Right=str.length-1;

while(Left<Right)

{

if(str[Left]!==str[Right]){

return false;

}

Left++;

Right--;

}

return true;

}

console.log(ispalindrome(str));

15) Sorting an array of strings by length:

const strings = [

"apple",

"ball",

"cat",

"dog",

"elephant",

"fridge",

];

strings.sort((a, b) => a.length - b.length);

console.log(strings);

Mathematical exercise

1(a).Write method int calc(int, int) that multiplies two variables, m and n of type int, then divides the product by two, and outputs the remainder with respect to division by 7.

let m = prompt("Enter m Value");

let n = prompt("Enter n value");

let mul = m\*n ;

console.log(mul);

let div = mul/2;

console.log(parseInt(div));

let result = div%7;

console.log(parseInt(result));

1(b).Find the number as well as the sum of natural numbers, which are divisible by 2 or 7 upto a given maximum value (exclusive) and output it to the console. Write method void calcSumAndCountAllNumbersDivBy\_2\_Or\_7(int). Extend it so that it returns the two values instead of performing the console output.

function sum(N)

{

let ans = 0;

for (let i = 1; i <= N; i++) {

if (i % 2 == 0 || i % 7 == 0) {

ans += i;

}

}

return ans;

}

let N =prompt("enter the value");

console.log(sum(N));

1(c).Create the methods boolean isEven(n) and boolean isOdd(n) that will check if thepassed integer is even or odd, respectively.

let a = prompt("Enter a number: ");

if(a % 2 == 0) {

console.log("The number is even.");

}

else {

console.log("The number is odd.");

}

3.Write method List<Integer> calc Perfect Numbers(int) that calculates the perfect numbers up to a maximum value, say 10,000.

function Perfect(number) {

let sum = 0;

for (let i = 1; i < number; i++) {

if (number % i === 0) {

sum = sum+i;

}

}

return sum === number;

}

function findPerfectNumbers(maxNumber) {

const perfectNumbers = [];

for (let i = 1; i <= maxNumber; i++) {

if (Perfect(i)) {

perfectNumbers.push(i);

}

}

return perfectNumbers;

}

const maxNumber = prompt("Enter the max number ");

const perfectNumbers = findPerfectNumbers(maxNumber);

console.log(`Perfect numbers between 1 and ${maxNumber}: ${perfectNumbers}`);

 4.Write method List calcPrimesUpTo(int) to compute all prime numbers upto a given value. As a reminder, a prime number is a natural number greater than 1 andexclusively divisible by itself and by 1To compute a prime number, the so-called Sieveof Eratosthenes was described before.

function getPrimesBtw(n) {

n = prompt("enter the no")

const primes = [];

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

let Prime = true;

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

if (i % j === 0) {

Prime = false;

break;

}

}

if (Prime) {

primes.push(i);

}

}

return primes;

}

console.log(getPrimesBtw(n));

5. Compute all pairs of prime numbers with a distance of 2 (twin), 4 (cousin), and 6 (sexy)up to an upper bound for n. For twins then the following is true.

function isPrime(num) {

if (num <= 1) return false;

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) return false;

}

return true;

}

function primePairs(limit) {

limit = prompt("enter the number ")

const pairs = [];

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

for (let j = i + 1; j <= limit; j++) {

if (isPrime(i) && isPrime(j) && j - i === 2) {

pairs.push([i, j]);

}

}

}

return pairs;

}

console.log(primePairs(limit));

8(a).Compute all combinations of the values a, b, and c (each starting from 1 and less than100) for which the following formula holds: a 2 + b 2 = c 2.

const a = prompt("Enter first number ");

const b = prompt("Enter second number ");

const c = Math.sqrt(Math.pow(a, 2) + Math.pow(b, 2));

console.log(c);

8(b). Compute all combinations of the values a, b, c, and d (each starting from 1 and less than100) for which the following formula holds: a 2 + b 2 = c 2 + d 2.

const a = prompt("Enter first number ");

const b = prompt("Enter second number ");

const c = Math.sqrt((Math.pow(a, 2) + Math.pow(b, 2))/2 + Math.sqrt(2)\*Math.sqrt(Math.pow(a, 2)\*Math.pow(b, 2))/2);

const d = Math.sqrt((Math.pow(a, 2) + Math.pow(b, 2))/2 - Math.sqrt(2)\*Math.sqrt(Math.pow(a, 2)\*Math.pow(b, 2))/2);

console.log(c, d);