1. **Find Missing Number**

let arr = [1, 6,2, 3, 5];

let sort=arr.sort((a,b)=>a-b);

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

if(arr.indexOf(i)==-1){

console.log("sort2",i)

}

}

console.log("sort",sort) // 4

**2 . find second Largest number**

let arr=[2,1,6,8,8,3,10];

let arr2=arr.filter((i,index)=>arr.indexOf(i)==index);

let sortArr=arr2.sort((a,b)=>a-b)

let secondLargestNo=sortArr[sortArr.length-2]

console.log("arr",secondLargestNo)//8

1. **Palindrome Number**

let str='madam'

for(let i=0;i<str.length/2;i++){

if(str[i]==str[str.length-1-i]){

console.log('yes',str.length)

}

else{

console.log('no')

}

}

1. **Factorial number**

let no=5

if(no<0){

console.log('error');

}

else{

let fact=1;

for(let i=1;i<=no;i++){

console.log("i",i)

fact=fact\*i

}

console.log("fact",fact)// 120

}

1. **Check the Prime Number or Not**

let num = 8;

let value = checkPrime(num);

console.log('value', value);

function checkPrime(no) {

let isPrime = true;

if (no == 1) {

console.log('Not Prime nor composite number');

return false;

} else if (no > 1) {

for (let i = 2; i < no; i++) {

if (no % i == 0) {

isPrime = false;

break;

}

}

}

return isPrime;

}

1. **ArmStrong Number 153=1^3+5^3+3^3**

let no=153

let armstrongNumber=getNumber(no);

console.log("armstrongNumber",armstrongNumber)

function getNumber(num){

let sum=0

let temp=num;

while(temp>0){

let re=temp%10

sum=sum+re\*\*3

temp=parseInt(temp/10);

}

return sum;

**6 Remove duplicate value**

1. let number=[1,2,3,2,4,5,5,6]

let value=[...new Set(number)];

console.log('value',value);

let removeDub=number.filter((i,index)=>number.indexOf(i)==index);

console.log('value',removeDub);

1. **find max occurance of char and count**

let str = 'kirti';

let value = findOccurance(str);

console.log('value', value);

function findOccurance(str) {

const obj = {};

let maxCount = 0;

let maxChar = '';

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

let char = str[i];

if (obj[char]) {

obj[char]++;

} else {

obj[char] = 1;

}

if (obj[char] > maxCount) {

maxCount = obj[char];

maxChar = char;

}

}

return { , maxCount,obj };

//value { maxChar: 'i', maxCount: 2, obj: { k: 1, i: 2, r: 1, t: 1 } }

**7. flattenArray want into single array**

function flattenArray(arr){

let result=[]

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

if(Array.isArray(arr[i])){

let newArray=flattenArray(arr[i]);

result.push(...newArray);

}else{

result.push(arr[i]);

}

}

return result;

}

const originalArray = [1,[8,6], 2, 3, [[4, 8,[5]]], [6]];

const flattenedArray = flattenArray(originalArray);

console.log(flattenedArray);

// [ 1, 8, 6, 2, 3,4, 8, 5, 6]

1. [**Longest Common PrefixLongest Common Prefix**](https://leetcode.com/problems/longest-common-prefix/)

**Input:** strs = ["flower","flow","flight"]

**Output:** "fl"

var longestCommonPrefix = function(strs) {

if (strs.length === 0) return "";

let result = strs[0];

let currPo=0;

let length=result.length;

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

while (strs[i].indexOf(result) !== 0) {

result = result.substring(0, result.length - 1);

if (result === "") return "";

}

}

return result;

};

1. [**Longest Substring Without Repeating Characters**](https://leetcode.com/problems/longest-substring-without-repeating-characters/)
2. **Input:** s = "abcabcbb"
3. **Output:** 3
4. **Explanation:** The answer is "abc", with the length of 3.

var lengthOfLongestSubstring = function(s) {

let map=new Map();

let start=0;

let maxLength=0;

for(let end=0; end<s.length;end++){ //end=0

if(map.has(s[end])){ //false

start=Math.max(map.get(s[end])+1,start);

}

map.set(s[end],end); //map.set(a,0)=>{a->0}

maxLength=Math.max(maxLength,end-start+1); //max(0,0-0+1)=>1

}

console.log(' return maxLength;',maxLength)

return maxLength;

};

Or

var lengthOfLongestSubstring = function (s) {  
    let currentString = [];  
    let longestStringLength = 0;

    for (let i = 0; i < s.length; i++) {  
        const currentCharacterPosition = currentString.indexOf(s[i]);  
        if (currentCharacterPosition !== -1) {  
         currentString.splice(0, currentCharacterPosition + 1);  
        }  
        currentString.push(s[i]);  
  
        longestStringLength = Math.max(  
            longestStringLength,  
            currentString.length  
        );  
    }

    return longestStringLength;  
};

console.log(lengthOfLongestSubstring('abcadabb'),'hhh')

9. [**Reverse Integer**](https://leetcode.com/problems/reverse-integer/)

**Input:** x = 123

**Output:** 321

var reverse = function(x) {

let rev=0;

let temp=0;

let remi=0

temp=Math.abs(x);

while(temp!=0){

remi=temp%10;

rev=rev\*10+remi;

temp=Math.floor(temp/10)

if(rev>2147483647){

return 0;

}

}

if (x < 0) {

rev = -rev;

}

return rev;

};

10. [**Palindrome Number**](https://leetcode.com/problems/palindrome-number/)

**Input:** x = 121

**Output:** true

**Explanation:** 121 reads as 121 from left to right and from right to left.

var isPalindrome = function(x) {

if (x < 0) return false;

let rev=0;

let rem=0;

let temp=x;

while (temp !== 0) {

let rem = temp % 10;

rev = rev \* 10 + rem;

temp = Math.floor(temp / 10);

}

// Check if the original number is equal to the reversed number

return x === rev;

};

11. **Move all zeroes to end of array**

Input : arr[] = {1, 2, 0, 4, 3, 0, 5, 0};

Output : arr[] = {1, 2, 4, 3, 5, 0, 0, 0};

function moveZeroesToEnd(arr) {

let nonZeroes = [];

let zeroes = [];

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

if (arr[i] === 0) {

zeroes.push(arr[i]);

} else {

nonZeroes.push(arr[i]);

}

}

return nonZeroes.concat(zeroes);

}

let arr = [1, 2, 0, 4, 3, 0, 5, 0];

arr = moveZeroesToEnd(arr);

console.log(arr); // Output: [1, 2, 4, 3, 5, 0, 0, 0]

1. Remove special character from string using regular expression

const str = "abc's test#s";

console.log(str.replace(/[^a-zA-Z ]/g, ""));

1. **Print Anagram a string**
2. **function** areAnagram(str1,str2)
3. {
4. // Get lengths of both strings
5. let n1 = str1.length;
6. let n2 = str2.length;
7. **if** (n1 != n2)
8. **return** **false**;
10. // Sort both strings
11. str1.sort();
12. str2.sort()
14. // Compare sorted strings
15. **for** (let i = 0; i < n1; i++)
16. **if** (str1[i] != str2[i])
17. **return** **false**;
19. **return** **true**;
20. }
21. let str1=['t', 'e', 's', 't'];
22. let str2=['t', 't', 'e', 'w'];

**13. second largest number;**

**let arr=[10,30,50,5,66,50,70,68]**

**let maxVaue=arr[0]**

**let sLar=0**

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

**if(arr[i]>maxVaue){**

**maxVaue=arr[i]**

**sLar=arr[i]**

**}else if(maxVaue>arr[i]){**

**sLar=arr[i]**

**}**

**}**

**console.log('dd',maxVaue,sLar)**