**1.** Write a JavaScript function to check whether an `input` is an array or not.   
*Test Data* :  
console.log(is\_array('w3resource'));  
console.log(is\_array([1, 2, 4, 0]));  
false  
true

**2.** Write a JavaScript function to clone an array.   
*Test Data* :  
console.log(array\_Clone([1, 2, 4, 0]));  
console.log(array\_Clone([1, 2, [4, 0]]));  
[1, 2, 4, 0]  
[1, 2, [4, 0]]

**3.** Write a JavaScript function to get the first element of an array. Passing a parameter 'n' will return the first 'n' elements of the array.   
*Test Data*:  
console.log(first([7, 9, 0, -2]));  
console.log(first([],3));  
console.log(first([7, 9, 0, -2],3));  
console.log(first([7, 9, 0, -2],6));  
console.log(first([7, 9, 0, -2],-3));  
*Expected Output*:  
7  
[]  
[7, 9, 0]  
[7, 9, 0, -2]  
[]

**4.** Write a JavaScript function to get the last element of an array. Passing a parameter 'n' will return the last 'n' elements of the array.   
*Test Data*:  
console.log(last([7, 9, 0, -2]));  
console.log(last([7, 9, 0, -2],3));  
console.log(last([7, 9, 0, -2],6));  
*Expected Output*:  
-2  
[9, 0, -2]  
[7, 9, 0, -2]

**5.** Write a simple JavaScript program to join all elements of the following array into a string.   
*Sample array*: myColor = ["Red", "Green", "White", "Black"];  
*Expected Output*:  
"Red,Green,White,Black"  
"Red,Green,White,Black"  
"Red+Green+White+Black"

**6.** Write a JavaScript program which accept a number as input and insert dashes (-) between each two even numbers. For example if you accept 025468 the output should be 0-254-6-8. 

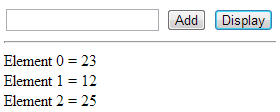
**7.** Write a JavaScript program to sort the items of an array.   
*Sample array* : var arr1 = [ 3, 8, 7, 6, 5, -4, 3, 2, 1 ];  
*Sample Output* : -4,-3,1,2,3,5,6,7,8  
  
  
**8.** Write a JavaScript program to find the most frequent item of an array.   
*Sample array* : var arr1=[3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3];  
*Sample Output* : a ( 5 times )

**9.** Write a JavaScript program which accept a string as input and swap the case of each character. For example if you input 'The Quick Brown Fox' the output should be 'tHE qUICK bROWN fOX'. 

**10.** Write a JavaScript program which prints the elements of the following array.   
Note : Use nested for loops.  
Sample array : var a = [[1, 2, 1, 24], [8, 11, 9, 4], [7, 0, 7, 27], [7, 4, 28, 14], [3, 10, 26, 7]];  
*Sample Output* :  
"row 0"  
" 1"  
" 2"  
" 1"  
" 24"  
"row 1"  
------  
------

**11.** Write a JavaScript program to find the sum of squares of a numeric vector. 

**12.** Write a JavaScript program to compute the sum and product of an array of integers. 

**13.** Write a JavaScript program to add items in an blank array and display the items.   
*Sample Screen* :  


**14.** Write a JavaScript program to remove duplicate items from an array (ignore case sensitivity). 

**15.** We have the following arrays :   
color = ["Blue ", "Green", "Red", "Orange", "Violet", "Indigo", "Yellow "];  
o = ["th","st","nd","rd"]  
Write a JavaScript program to display the colors in the following way :  
"1st choice is Blue ."  
"2nd choice is Green."  
"3rd choice is Red."  
- - - - - - - - - - - - -  
Note : Use ordinal numbers to tell their position.

**16.** Write a JavaScript program to find the leap years in a given range of years. 

**17.** Write a JavaScript program to shuffle an array. 

**18.** Write a JavaScript program to perform a binary search.   
Note : A binary search or half-interval search algorithm finds the position of a specified input value within an array sorted by key value.  
Sample array :  
var items = [1, 2, 3, 4, 5, 7, 8, 9];  
Expected Output :  
console.log(binary\_Search(items, 1)); //0  
console.log(binary\_Search(items, 5)); //4

**19.** There are two arrays with individual values, write a JavaScript program to compute the sum of each individual index value from the given arrays.   
Sample array :  
array1 = [1,0,2,3,4];  
array2 = [3,5,6,7,8,13];  
Expected Output :  
[4, 5, 8, 10, 12, 13]

**20.** Write a JavaScript program to find duplicate values in a JavaScript array. 

**21.** Write a JavaScript program to flatten a nested (any depth) array. If you pass shallow, the array will only be flattened a single level.   
Sample Data :  
console.log(flatten([1, [2], [3, [[4]]],[5,6]]));  
[1, 2, 3, 4, 5, 6]  
console.log(flatten([1, [2], [3, [[4]]],[5,6]], true));  
[1, 2, 3, [[4]], 5, 6]

**22.** Write a JavaScript program to compute the union of two arrays.   
Sample Data :  
console.log(union([1, 2, 3], [100, 2, 1, 10]));  
[1, 2, 3, 10, 100]

**23.** Write a JavaScript function to find the difference of two arrays.  *Test Data* :  
console.log(difference([1, 2, 3], [100, 2, 1, 10]));  
["3", "10", "100"]  
console.log(difference([1, 2, 3, 4, 5], [1, [2], [3, [[4]]],[5,6]]));  
["6"]  
console.log(difference([1, 2, 3], [100, 2, 1, 10]));  
["3", "10", "100"]

**24.** Write a JavaScript function to remove. 'null', '0', '""', 'false', 'undefined' and 'NaN' values from an array.   
Sample array : [NaN, 0, 15, false, -22, '',undefined, 47, null]  
Expected result : [15, -22, 47]

**25.** Write a JavaScript function to sort the following array of objects by title value.   
Sample object :

var library = [

{ author: 'Bill Gates', title: 'The Road Ahead', libraryID: 1254},

{ author: 'Steve Jobs', title: 'Walter Isaacson', libraryID: 4264},

{ author: 'Suzanne Collins', title: 'Mockingjay: The Final Book of The Hunger Games', libraryID: 3245}

];

Expected result :

[[object Object] {

author: "Suzanne Collins",

libraryID: 3245,

title:"Mockingjay:The Final Book of The Hunger Games"

}, [object Object] {

author: "Bill Gates",

libraryID: 1254,

title: "The Road Ahead"

}, [object Object] {

author: "Steve Jobs",

libraryID: 4264,

title: "Walter Isaacson"

}]

**26.** Write a JavaScript program to find a pair of elements (indices of the two numbers) from an given array whose sum equals a specific target number. 

Input: numbers= [10,20,10,40,50,60,70], target=50  
Output: 2, 3

**27.** Write a JavaScript function to retrieve the value of a given property from all elements in an array.   
Sample array : [NaN, 0, 15, false, -22, '',undefined, 47, null]  
Expected result : [15, -22, 47]

**28.** Write a JavaScript function to find the longest common starting substring in a set of strings.

Sample array : console.log(longest\_common\_starting\_substring(['go', 'google']));  
Expected result : "go"

**29.** Write a JavaScript function to fill an array with values (numeric, string with one character) on supplied bounds.

Test Data :  
console.log(num\_string\_range('a', "z", 2));  
["a", "c", "e", "g", "i", "k", "m", "o", "q", "s", "u", "w", "y"]

**30.** Write a JavaScript function to merge two arrays and removes all duplicates elements. 

Test data :  
var array1 = [1, 2, 3];  
var array2 = [2, 30, 1];  
console.log(merge\_array(array1, array2));  
[3, 2, 30, 1]

**31.** Write a JavaScript function to remove a specific element from an array. 

Test data :  
console.log(remove\_array\_element([2, 5, 9, 6], 5));  
[2, 9, 6]

**32.** Write a JavaScript function to find an array contains a specific element. 

Test data :  
arr = [2, 5, 9, 6];  
console.log(contains(arr, 5));  
[True]

**33.** Write a JavaScript script to empty an array keeping the original. 

.

**34.** Write a JavaScript function to get nth largest element from an unsorted array. 

Test Data :  
console.log(nthlargest([ 43, 56, 23, 89, 88, 90, 99, 652], 4));  
89

**35.** Write a JavaScript function to get a random item from an array. 

**36.** Write a JavaScript function to create a specified number of elements with pre-filled numeric value array. 

Test Data :  
console.log(array\_filled(6, 0));  
[0, 0, 0, 0, 0, 0]  
console.log(array\_filled(4, 11));  
[11, 11, 11, 11]

**37.** Write a JavaScript function to create a specified number of elements with pre-filled string value array. 

Test Data :  
console.log(array\_filled(3, 'default value'));  
["default value", "default value", "default value"]  
console.log(array\_filled(4, 'password'));  
["password", "password", "password", "password"]

**38.** Write a JavaScript function to move an array element from one position to another. 

Test Data :  
console.log(move([10, 20, 30, 40, 50], 0, 2));  
[20, 30, 10, 40, 50]  
console.log(move([10, 20, 30, 40, 50], -1, -2));  
[10, 20, 30, 50, 40]

**39.** Write a JavaScript function to filter false, null, 0 and blank values from an array. 

*Test Data* :  
console.log(filter\_array\_values([58, '', 'abcd', true, null, false, 0]));  
[58, "abcd", true]

**40.** Write a JavaScript function to generate an array of specified length, filled with integer numbers, increase by one from starting position. 

*Test Data* :  
console.log(array\_range(1, 4));  
[1, 2, 3, 4]  
console.log(array\_range(-6, 4));  
[-6, -5, -4, -3]

**41.**Write a JavaScript function to generate an array between two integers of 1 step length. 

*Test Data* :  
console.log(rangeBetwee(4, 7));  
[4, 5, 6, 7]  
console.log(rangeBetwee(-4, 7));  
[-4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7]

**42.** Write a JavaScript function to find the unique elements from two arrays. 

*Test Data* :  
console.log(difference([1, 2, 3], [100, 2, 1, 10]));  
["1", "2", "3", "10", "100"]  
console.log(difference([1, 2, 3, 4, 5], [1, [2], [3, [[4]]],[5,6]]));  
["1", "2", "3", "4", "5", "6"]  
console.log(difference([1, 2, 3], [100, 2, 1, 10]));  
["1", "2", "3", "10", "100"]

**43.** Write a JavaScript function to create an array of arrays, ungrouping the elements in an array produced by zip. 

*Test Data* :  
unzip([['a', 1, true], ['b', 2, false]])  
unzip([['a', 1, true], ['b', 2]])  
Expected Output:  
[["a","b"],[1,2],[true,false]]  
[["a","b"],[1,2],[true]]

**44.** Write a JavaScript function to create an object from an array, using the specified key and excluding it from each value. 

*Test Data* :  
indexOn([ { id: 10, name: 'apple' }, { id: 20, name: 'orange' } ], x => x.id)  
Expected Output:  
{"undefined":{"id":20,"name":"orange"}}