**Task 5**

1. Do the below programs in anonymous function & IIFE
   1. Print odd numbers in an array

|  |
| --- |
| anonymous : function(array){ |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | if(array[i]%2!=0){ |
|  | console.log(array[i]) |
|  | } |
|  | } |
|  | } |
|  | IIFE : (function(array){ |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | if(array[i]%2!=0){ |
|  | console.log(array[i]) |
|  | } |
|  | } |
|  | })([10,20,30,40]) |

* 1. Convert all the strings to title caps in a string array

|  |
| --- |
|  |
| anonymous: function (str) { |
|  | str = str.toLowerCase().split(' '); |
|  | for (var i = 0; i < str.length; i++) { |
|  | str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1); |
|  | } |
|  | return str.join(' '); |
|  | } |
|  | IIFE : (function (str) { |
|  | str = str.toLowerCase().split(' '); |
|  | for (var i = 0; i < str.length; i++) { |
|  | str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1); |
|  | } |
|  | return str.join(' '); |
|  | })("JAVASCRIPT"); |
|  |  |

* 1. Sum of all numbers in an array

|  |
| --- |
| anonymous : function(array){ |
|  | var sum = 0; |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | sum = sum + array[i]; |
|  | } |
|  | return sum; |
|  | } |
|  | IIFE : (function(array){ |
|  | var sum = 0; |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | sum = sum + array[i]; |
|  | } |
|  | return sum; |
|  | })([1,2,3,4]) |

* 1. Return all the prime numbers in an array

|  |
| --- |
| Anonymous: |
|  | function(numArray){ |
|  | numArray = numArray.filter((number) => { |
|  | for (var i = 2; i <= Math.sqrt(number); i++) { |
|  | if (number % i === 0) return false; |
|  | } |
|  | return true; |
|  | }); |
|  | console.log(numArray); |
|  | } |
|  | IIFE: |
|  | ( |
|  | function(numArray){ |
|  | numArray = numArray.filter((number) => { |
|  | for (var i = 2; i <= Math.sqrt(number); i++) { |
|  | if (number % i === 0) return false; |
|  | } |
|  | return true; |
|  | }); |
|  | console.log(numArray); |
|  | })([1,2,3,4]) |

* 1. Return all the palindromes in an array

|  |
| --- |
| Anonymous: function (arr, n) |
|  | { |
|  |
|  |
|  | for (let i = 0; i < n; i++) |
|  | { |
|  | let ans = isPalindrome(arr[i]); |
|  | if (ans == false) |
|  | return false; |
|  | } |
|  | return true; |
|  | } |
|  |  |
|  | IIFE : |
|  |  |
|  | ( function (arr, n) |
|  | { |
|  |
|  |
|  | for (let i = 0; i < n; i++) |
|  | { |
|  | let ans = isPalindrome(arr[i]); |
|  | if (ans == false) |
|  | return false; |
|  | } |
|  | return true; |
|  | })([1,2,3] , 3) |

* 1. Return median of two sorted arrays of the same size.

Anonymous function:

const medianOfTwoSortedArrays = (arr1, arr2) =>

{ let arr3 = arr1.concat(arr2).sort(function(a,b)

{

return a - b;

});

let middleIndex = Math.floor(arr3.length / 2);

if (arr3.length % 2 === 0)

{

return (arr3[middleIndex] + arr3[middleIndex - 1]) / 2;

}

Else

{

return arr3[middleIndex];

} }

let arr1 = [1, 3, 5, 7];

let arr2 = [2, 4, 6, 8];

console.log(medianOfTwoSortedArrays(arr1, arr2)); // 4.5

IIFE:

(function(arr1, arr2)

{

let median;

let combinedArr = arr1.concat(arr2);

combinedArr.sort(function(a, b){return a-b});

if (combinedArr.length % 2 === 0)

{

median = (combinedArr[combinedArr.length/2] + combinedArr[combinedArr.length/2 - 1])/2;

}

else

{

median = combinedArr[Math.floor(combinedArr.length/2)];

}

return median;

})([1,2,3,4], [5,6,7,8]); //4.5

* 1. Remove duplicates from an array

|  |
| --- |
| Anonymous: function(array){ |
|  | let dup = [...new Set(array)]; |
|  | console.log(dup); |
|  | } |
|  | IIFE:  (function(array){ |
|  | let dup = [...new Set(array)]; |
|  | console.log(dup); |
|  | })([1,1,2,3,4]) |
|  |  |

* 1. Rotate an array by k times

|  |
| --- |
| Anonymous: function(array,k){ |
|  | k = k % a.length; |
|  | if(k < 0){ |
|  | k += a.length; |
|  | } |
|  |  |
|  | reverse(a, 0, a.length - k - 1); |
|  | reverse(a, a.length - k, a.length - 1); |
|  | reverse(a, 0, a.length - 1); |
|  | } |
|  |  |
|  | IIFE : (function(array , k){ |
|  | k = k % a.length; |
|  | if(k < 0){ |
|  | k += a.length; |
|  | } |
|  |  |
|  | reverse(a, 0, a.length - k - 1); |
|  | reverse(a, a.length - k, a.length - 1); |
|  | reverse(a, 0, a.length - 1); |
|  | })([1,2,3,4] , 2) |