1. Do the below programs in anonymous function, IIFE & Arrow Function

1. **Print odd numbers in an array**

**Anonymous**: let x = function(array) {

console.log(array.filter((element) => element%2 != 0).join(" "))

}

**IIFE**: (function(array) {

console.log(array.filter((element) => element%2 != 0).join(" "))

})();

**Arrow Function**:

(array) => {console.log(array.filter((element) => element%2 != 0).join(" "))}

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

**Anonymous**: let x = function(string) {

var sentence = string.toLowerCase().split(" ");

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

sentence[i] = sentence[i][0].toUpperCase() + sentence[i].slice(1);

}

**IIFE**: (function(string) {

var sentence = string.toLowerCase().split(" ");

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

sentence[i] = sentence[i][0].toUpperCase() + sentence[i].slice(1);

}) ();

**Arrow Function**:

(string) => { var sentence = string.toLowerCase().split(" ");

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

sentence[i] = sentence[i][0].toUpperCase() + sentence[i].slice(1); }

c.  **Sum of all numbers in an array**

**Anonymous**: let x = function (array) {

array.reduce((ini,cur) => ini+cur);

}

**IIFE**: function (array) {

console.log(array.reduce((ini,cur) => ini+cur));

}()

**Arrow Function**: (array) => array.reduce((ini,cur) => ini+cur);

d. **Return all the prime numbers in an array**

**Anonymous:**  let x = function (array) {

array.filter( (val) =>

{

var isPrime = true;

for(var i = 2; i < Math.Ceil(val/2); i++)

{

if( val % i == 0)

{

isPrime = false;

return isPrime;

}

}

return isPrime;

}

);

}

**IIFE**: (

function (array) {

array.filter( (val) =>

{

var isPrime = true;

for(var i = 2; i < Math.Ceil(val/2); i++)

{

if( val % i == 0)

{

isPrime = false;

return isPrime;

}

}

return isPrime;

}

);

})();

**Arrow Function**:

(array) => {

array.filter( (val) =>

{

var isPrime = true;

for(var i = 2; i < Math.Ceil(val/2); i++)

{

if( val % i == 0)

{

isPrime = false;

return isPrime;

}

}

return isPrime;

}

);

}

e. **Return all the palindromes in an array**

**Anonymous**: let x = function (array){

return array.filter( (val) => val.split(" ").reverse.join('') == val);

}

**IIFE**: (function (array){

return array.filter( (val) => val.split(" ").reverse.join('') == val);

})();

**Arrow Function**: (array)=> array.filter( (val) => val.split(" ").reverse.join('') == val);

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

**Anonymous**: let x = function (array) {

sortedArray = array.sort((a,b) => a-b);

if (sortedArray.length % 2 == 0)

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

else

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

return median;

}

**IIFE**: (function (array) {

sortedArray = array.sort((a,b) => a-b);

if (sortedArray.length % 2 == 0)

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

else

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

return median;

})();

**Arrow Function**:

(array) => {

sortedArray = array.sort((a,b) => a-b);

if (sortedArray.length % 2 == 0)

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

else

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

return median;

}

g.  **Remove duplicates from an array**

**Anonymous**: let x = function(array) {

uniqueArray = [];

for (var element of array)

{

if(uniqueArray.includes(element))

continue;

else

uniqueArray.push(element);

}

return uniqueArray; }

**IIFE**: (function(array) {

uniqueArray = [];

for (var element of array)

{

if(uniqueArray.includes(element))

continue;

else

uniqueArray.push(element);

}

return uniqueArray; })();

**Arrow Function**:

(array) => {

uniqueArray = [];

for (var element of array)

{

if(uniqueArray.includes(element))

continue;

else

uniqueArray.push(element);

}

return uniqueArray; }

h. **Rotate an array by k times**

**Anonymous:**

let x = function (array, k) {

rotatedArray = [];

rotationCount = k;

for (var index in array)

{

if (index + rotationCount <= array.length-1)

rotatedArray[ index + rotationCount ] = array[index];

else

rotatedArray[ index + rotationCount - rotatedArray.length ] = array[index];

}

return rotatedArray;

}

**IIFE**:

(function (array, k) {

rotatedArray = [];

rotationCount = k;

for (var index in array)

{

if (index + rotationCount <= array.length-1)

rotatedArray[ index + rotationCount ] = array[index];

else

rotatedArray[ index + rotationCount - rotatedArray.length ] = array[index];

}

return rotatedArray;

})();

**Arrow Function**:

(array, k) => {

rotatedArray = [];

rotationCount = k;

for (var index in array)

{

if (index + rotationCount <= array.length-1)

rotatedArray[ index + rotationCount ] = array[index];

else

rotatedArray[ index + rotationCount - rotatedArray.length ] = array[index];

}

return rotatedArray;

}