**Do the below programs in anonymous function & IIFE**

1. Print odd numbers in an array in anonymous function & IIFE

* **Using anonymous function**:

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const oddNumbers = numbers.filter(function(number) {

return number % 2 !== 0;

});

console.log(oddNumbers);

* **Using IIFE**:

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

(function() {

const oddNumbers = numbers.filter(function(number) {

return number % 2 !== 0;

});

console.log(oddNumbers); // Output: [1, 3, 5, 7, 9]

})();

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

* **Using anonymous function:**

const stringArray = ["hello world", "goodbye universe", "welcome to earth"];

const titleCapsArray = stringArray.map(function(str) {

return str.split(' ').map(function(word) {

return word.charAt(0).toUpperCase() + word.slice(1).toLowerCase();

}).join(' ');

});

console.log(titleCapsArray)**;**

* **Using IIFE**:

const stringArray = ["hello world", "goodbye universe", "welcome to earth"];

(function() {

const titleCapsArray = stringArray.map(function(str) {

return str.split(' ').map(function(word) {

return word.charAt(0).toUpperCase() + word.slice(1).toLowerCase();

}).join(' ');

});

console.log(titleCapsArray);

})();

3. Sum of all numbers in an array

* **Using anonymous function:**

const numbers = [1, 2, 3, 4, 5];

const sum = function(arr) {

return arr.reduce(function(total, num) {

return total + num;

}, 0);

};

console.log(sum(numbers));

* **Using IIFE:**

const numbers = [1, 2, 3, 4, 5];

(function(arr) {

const sum = arr.reduce(function(total, num) {

return total + num;

}, 0);

console.log(sum);

})(numbers);

4. Return all the prime numbers in an array

* **Using anonymous function:**

const numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11];

const isPrime = function(num) {

if (num <= 1) return false;

if (num <= 3) return true;

if (num % 2 === 0 || num % 3 === 0) return false;

for (let i = 5; i \* i <= num; i += 6) {

if (num % i === 0 || num % (i + 2) === 0) return false;

}

return true;

};

const primeNumbers = numbers.filter(function(number) {

return isPrime(number);

});

console.log(primeNumbers);

* **Using IIFE:**

const numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11];

(function(arr) {

const isPrime = function(num) {

if (num <= 1) return false;

if (num <= 3) return true;

if (num % 2 === 0 || num % 3 === 0) return false;

for (let i = 5; i \* i <= num; i += 6) {

if (num % i === 0 || num % (i + 2) === 0) return false;

}

return true;

};

const primeNumbers = arr.filter(function(number) {

return isPrime(number);

});

console.log(primeNumbers);

})(numbers);

5. Return all the palindromes in an array

* **Using anonymous function:**

const words = ["radar", "hello", "level", "world", "deified"];

const isPalindrome = function(str) {

const reversed = str.split('').reverse().join('');

return str === reversed;

};

const palindromeWords = words.filter(function(word) {

return isPalindrome(word);

});

console.log(palindromeWords);

* **Using IIFE:**

const words = ["radar", "hello", "level", "world", "deified"];

(function(arr) {

const isPalindrome = function(str) {

const reversed = str.split('').reverse().join('');

return str === reversed;

};

const palindromeWords = arr.filter(function(word) {

return isPalindrome(word);

});

console.log(palindromeWords);

})(words);

6. Return median of two sorted arrays of the same size.

* **Using anonymous function:**

const arr1 = [1, 3, 5];

const arr2 = [2, 4, 6];

const findMedian = function(arr1, arr2) {

const merged = arr1.concat(arr2).sort((a, b) => a - b);

const midIndex = Math.floor(merged.length / 2);

if (merged.length % 2 === 0) {

return (merged[midIndex - 1] + merged[midIndex]) / 2;

} else {

return merged[midIndex];

}

};

console.log(findMedian(arr1, arr2));

* **Using IIFE:**

const arr1 = [1, 3, 5];

const arr2 = [2, 4, 6];

(function(arr1, arr2) {

const merged = arr1.concat(arr2).sort((a, b) => a - b);

const midIndex = Math.floor(merged.length / 2);

if (merged.length % 2 === 0) {

console.log((merged[midIndex - 1] + merged[midIndex]) / 2);

} else {

console.log(merged[midIndex]);

}

})(arr1, arr2);

7.Remove duplicates from an array

* **Using anonymous function:**

const numbers = [1, 2, 2, 3, 4, 4, 5, 5];

const removeDuplicates = function(arr) {

return arr.filter(function(value, index, self) {

return self.indexOf(value) === index;

});

};

const uniqueNumbers = removeDuplicates(numbers);

console.log(uniqueNumbers);

* **Using IIFE:**

const numbers = [1, 2, 2, 3, 4, 4, 5, 5];

(function(arr) {

const uniqueArray = arr.filter(function(value, index, self) {

return self.indexOf(value) === index;

});

console.log(uniqueArray);

})(numbers);

8. Rotate an array by k times

* **Using anonymous function:**

const array = [1, 2, 3, 4, 5];

const k = 2;

const rotateArray = function(arr, rotations) {

const len = arr.length;

const normalizedRotations = rotations % len;

const rotated = arr.slice(-normalizedRotations).concat(arr.slice(0, len - normalizedRotations));

return rotated;

};

const rotatedArray = rotateArray(array, k);

console.log(rotatedArray);

* **Using IIFE:**

const array = [1, 2, 3, 4, 5];

const k = 2;

(function(arr, rotations) {

const len = arr.length;

const normalizedRotations = rotations % len;

const rotatedArray = arr.slice(-normalizedRotations).concat(arr.slice(0, len - normalizedRotations));

console.log(rotatedArray);

})(array, k);

**Do the below programs in arrow functions**

1. Print odd numbers in an array:

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const oddNumbers = numbers.filter(number => number % 2 !== 0);

oddNumbers.forEach(oddNumber => console.log(oddNumber));

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

const stringArray = ["hello world", "goodbye universe", "welcome to earth"];

const titleCapsArray = stringArray.map(str => str.split(' ')

.map(word => word.charAt(0).toUpperCase() + word.slice(1).toLowerCase())

.join(' ')

);

titleCapsArray.forEach(titleCaps => console.log(titleCaps));

1. Sum of all numbers in an array:

const numbers = [1, 2, 3, 4, 5];

const sum = numbers.reduce((total, num) => total + num, 0);

console.log(sum);

1. Return all the prime numbers in an array:

const numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11];

const isPrime = num => {

if (num <= 1) return false;

if (num <= 3) return true;

if (num % 2 === 0 || num % 3 === 0) return false;

for (let i = 5; i \* i <= num; i += 6) {

if (num % i === 0 || num % (i + 2) === 0) return false;

}

return true;

};

const primeNumbers = numbers.filter(number => isPrime(number));

console.log(primeNumbers);

1. Return all the palindromes in an array:

const words = ["radar", "hello", "level", "world", "deified"];

const isPalindrome = str => {

const reversed = str.split('').reverse().join('');

return str === reversed;

};

const palindromeWords = words.filter(word => isPalindrome(word));

console.log(palindromeWords);