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

**Anonymous function::**

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

(function(arr) {

console.log("Odd numbers in the array:");

arr.forEach(function(num) {

if (num % 2 !== 0) {

console.log(num);

}

});

})(numbers);

**IIFE::**

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

(function(arr) {

console.log("Odd numbers in the array:");

arr.forEach(num => {

if (num % 2 !== 0) {

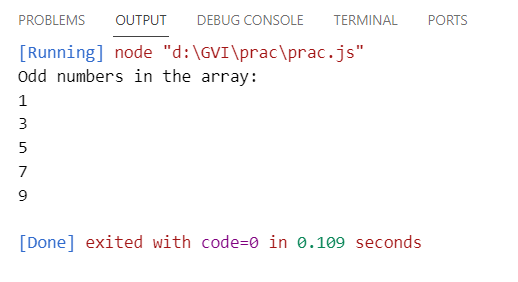
console.log(num);

}

});

})(numbers);

**Output::**



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

**Anonymous::**

const stringArray = ["hello world", "goodbye universe", "javascript is awesome"];

(function(arr) {

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

return str.replace(/\b\w/g, function(char) {

return char.toUpperCase();

});

});

console.log(titleCapsArray);

})(stringArray);

**IIFE::**

const stringArray = ["hello world", "goodbye universe", "javascript is awesome"];

(function(arr) {

const titleCapsArray = arr.map(str => {

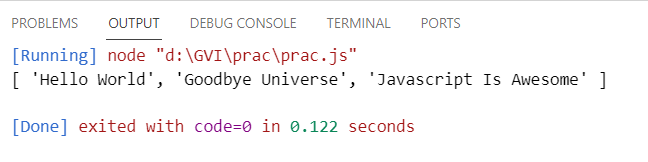
return str.replace(/\b\w/g, char => char.toUpperCase());

});

console.log(titleCapsArray);

})(stringArray);

**Output::**



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

**Anonymous::**

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

(function(arr) {

let sum = 0;

arr.forEach(function(num) {

sum += num;

});

console.log("Sum of all numbers:", sum);

})(numbers);

**IIFE::**

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

(function(arr) {

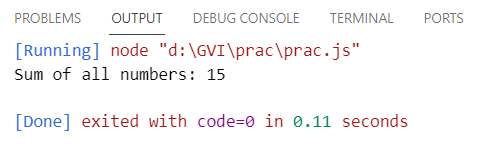
const sum = arr.reduce(function(acc, num) {

return acc + num;

}, 0);

console.log("Sum of all numbers:", sum);

})(numbers);



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

**Anonymous Function::**

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

(function(arr) {

const isPrime = function(num) {

if (num <= 1) return false;

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) return false;

}

return true;

};

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

return isPrime(num);

});

console.log("Prime numbers in the array:", primeNumbers);

})(numbers);

**IIFE::**

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

(function(arr) {

const isPrime = function(num) {

if (num <= 1) return false;

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) return false;

}

return true;

};

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

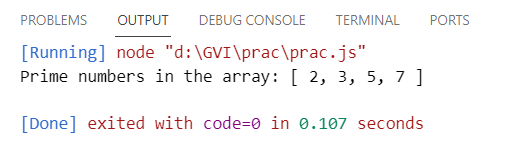
return isPrime(num);

});

console.log("Prime numbers in the array:", primeNumbers);

})(numbers);

**Output::**



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

**Anonymous::**

const array1 = [1, 3, 5];

const array2 = [2, 4, 6];

(function(arr1, arr2) {

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

const mid = Math.floor(mergedArray.length / 2);

const median = mergedArray.length % 2 === 0 ?

(mergedArray[mid - 1] + mergedArray[mid]) / 2 :

mergedArray[mid];

console.log("Median:", median);

})(array1, array2);

**IIFE::**

const array1 = [1, 3, 5];

const array2 = [2, 4, 6];

(function(arr1, arr2) {

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

const mid = Math.floor(mergedArray.length / 2);

const median = mergedArray.length % 2 === 0 ?

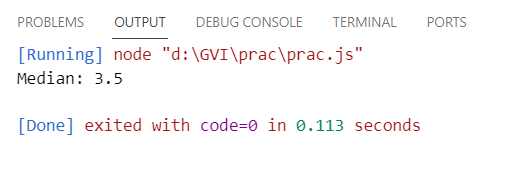
(mergedArray[mid - 1] + mergedArray[mid]) / 2 :

mergedArray[mid];

console.log("Median:", median);

})(array1, array2);

Output::



**1.g. Remove duplicates from an array**

**Anonymous::**

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

(function(arr) {

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

return arr.indexOf(item) === index;

});

console.log("Array with duplicates removed:", uniqueArray);

})(array);

**IIFE::**

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

(function(arr) {

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

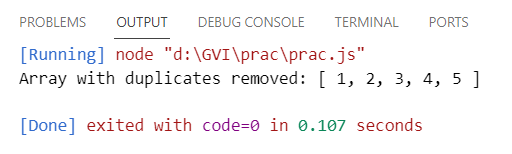
return arr.indexOf(item) === index;

});

console.log("Array with duplicates removed:", uniqueArray);

})(array);

**Output::**



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

**Anonymous::**

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

(function(arr) {

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

const reversedWord = word.split("").reverse().join("");

return word === reversedWord;

});

console.log("Palindromes in the array:", palindromes);

})(words);

**IIFE::**

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

(function(arr) {

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

const reversedWord = word.split("").reverse().join("");

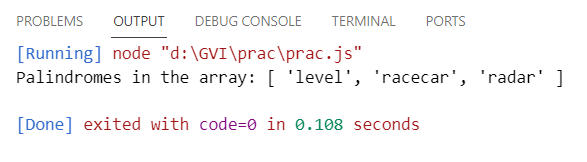
return word === reversedWord;

});

console.log("Palindromes in the array:", palindromes);

})(words);

**Output::**



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

**Anonymous::**

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

const k = 3;

(function(arr, k) {

for (let i = 0; i < k; i++) {

arr.unshift(arr.pop());

}

console.log("Array rotated by", k, "times:", arr);

})(array, k);

IIFE::

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

const k = 3;

(function(arr, k) {

for (let i = 0; i < k; i++) {

arr.unshift(arr.pop());

}

console.log("Array rotated by", k, "times:", arr);

})(array, k);

**Output::**

