1. programs in anonymous function & IIFE
2. odd numbers in an array;

let a = [1,2,3,4,5,6,7,8,9];

(function(array) {

    let oddNumbers = array.filter(function(num) {

        return num % 2 !== 0;

    });

console.log(oddNumbers);

})(a);

b. strings to title caps in a string array;

let stringArray = ["welcome to javascript day three"];

(function(array) {

    let titleCaseArray = array.map(function(str) {

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

            return char.toUpperCase();

        });

    });

    console.log(titleCaseArray);

})(stringArray);

c. Sum of all numbers in an array;

let numbers = [1,3,5,7,9];

  (function(array) {

    let sum = array.reduce(function(total, num) {

        return total + num;

    }, 0);

      console.log("sum", sum);

})(numbers);

d. the prime numbers in an array

let numbers = [12,13,14,15,16,17,18,19];

(function(array) {

    let 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;

    };

    let primeNumbers = array.filter(num => isPrime(num));

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

})(numbers);

 e.palindromes in an array;

(function(array) {

    const isPalindrome = str => {

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

        return str === reversed;

    };

    const palindromes = array.filter(str => isPalindrome(str));

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

})(["madam","welcome","growth","rotor",]);

f.median of two sorted arrays of the same size;

(function(arr1, arr2) {

    const mergeArrays = (arr1, arr2) => {

        let merged = [];

        let i = 0, j = 0;

        while (i < arr1.length && j < arr2.length) {

            if (arr1[i] <= arr2[j]) {

                merged.push(arr1[i]);

                i++;

            } else {

                merged.push(arr2[j]);

                j++;

            }

        }

        while (i < arr1.length) {

            merged.push(arr1[i]);

            i++;

        }

        while (j < arr2.length) {

            merged.push(arr2[j]);

            j++;

        }

return merged;

    };

2:

    const mergedArray = mergeArrays(arr1, arr2);

    const length = mergedArray.length;

    const median = (mergedArray[length / 2 - 1] + mergedArray[length / 2]) / 2

    console.log("Median of the two sorted arrays:", median);

})([1,3,5], [2,4,6]);

 g.duplicates from an array;

(function(array) {

    const uniqueArray = array.filter((value, index, self) =>

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

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

})([1,2,2,3,4,4,5]);

h.Rotate an array by k times;

(function(array, k) {

    const rotateArray = (arr, k) => {

const n = array.length;

k %= n;

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

    const temp = array.pop();

    array.unshift(temp);

}

return array;

};

const rotatedArray = rotateArray(array, k);

console.log("Array after rotation:", rotatedArray);

})([1,2,3,4,5],2);

Thank You