JAVASCRIPT DAY3

1 a. Print odd numbers in an array

let array = [1, 2, 3, 4, 5, 6, 7];

let oddNumbers = array.filter(x => x % 2 !== 0);

console.log(oddNumbers);

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

let strings = ["hello", "world", "javascript", "programming"];

let titleCaps = strings.map(s => s.charAt(0).toUpperCase() + s.slice(1).toLowerCase());

console.log(titleCaps);

c. Sum of all numbers in an array

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

let totalSum = numbers.reduce((acc, cur) => acc + cur, 0);

console.log(totalSum);

d. Return all the prime numbers in an array

let isPrime = num => {

    if (num < 2) return false;

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

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

    }

    return true;

};

let primes = numbers.filter(isPrime);

console.log(primes);

e. Return all the palindromes in an array

let words = ["radar", "apple", "rotor", "blue", "level"];

let palindromes = words.filter(w => w === w.split('').reverse().join(''));

console.log(palindromes);

2 a. Print odd numbers in an array

array = [1, 2, 3, 4, 5, 6, 7]

odd\_numbers = list(filter(lambda x: x % 2 != 0, array))

print(odd\_numbers)

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

strings = ["hello", "world", "python", "programming"]

title\_caps = list(map(lambda x: x.title(), strings))

print(title\_caps)

c. Sum of all numbers in an array

from functools import reduce

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

total\_sum = reduce(lambda x, y: x + y, numbers)

print(total\_sum)

d. Return all the prime numbers in an array

def is\_prime(n):

    if n < 2:

        return False

    for i in range(2, int(n\*\*0.5) + 1):

        if n % i == 0:

            return False

    return True

primes = list(filter(lambda x: is\_prime(x), numbers))

print(primes)

e. Return all the palindromes in an array

words = ["radar", "hello", "level", "world"]

palindromes = list(filter(lambda x: x == x[::-1], words))

print(palindromes)

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

def find\_median\_sorted\_arrays(arr1, arr2):

    merged = sorted(arr1 + arr2)

    mid = len(merged) // 2

    return (merged[mid - 1] + merged[mid]) / 2 if len(merged) % 2 == 0 else merged[mid]

# Example arrays

arr1 = [1, 3, 5]

arr2 = [2, 4, 6]

median = find\_median\_sorted\_arrays(arr1, arr2)

print(median)

g. Remove duplicates from an array

nums = [1, 2, 2, 3, 4, 4, 5]

unique\_nums = list(dict.fromkeys(nums))

print(unique\_nums)

h. Rotate an array by k times

def rotate\_array(arr, k):

    k = k % len(arr)

    return arr[-k:] + arr[:-k]

# Example usage

rotated\_array = rotate\_array([1, 2, 3, 4, 5], 2)

print(rotated\_array)