// 1. Reverse a string without built-in methods

function reverseString(str) {

let reversed = '';

for (let i = str.length - 1; i >= 0; i--) {

reversed += str[i];

}

return reversed;

}

// 2. Check if a number is a palindrome

function isPalindromeNumber(num) {

let original = num.toString();

let reversed = reverseString(original);

return original === reversed;

}

// 3. Find factorial of a number

function factorial(n) {

if (n === 0) return 1;

return n \* factorial(n - 1);

}

// 4. Find nth Fibonacci number (recursive)

function fibonacci(n) {

if (n <= 1) return n;

return fibonacci(n - 1) + fibonacci(n - 2);

}

// 5. Check if two strings are anagrams

function isAnagram(str1, str2) {

const normalize = str => str.replace(/[^a-z]/gi, '').toLowerCase().split('').sort().join('');

return normalize(str1) === normalize(str2);

}

// 6. Check if a number is prime

function isPrime(num) {

if (num < 2) return false;

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

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

}

return true;

}

// 7. Sum of digits of a number

function sumOfDigits(num) {

return num.toString().split('').reduce((sum, digit) => sum + parseInt(digit), 0);

}

// 8. Find maximum in array

function findMax(arr) {

let max = arr[0];

for (let num of arr) {

if (num > max) max = num;

}

return max;

}

// 9. Bubble sort

function bubbleSort(arr) {

let n = arr.length;

for (let i = 0; i < n - 1; i++) {

for (let j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

[arr[j], arr[j + 1]] = [arr[j + 1], arr[j]];

}

}

}

return arr;

}

// 10. Count occurrences of a character

function countChar(str, char) {

let count = 0;

for (let c of str) {

if (c === char) count++;

}

return count;

}

// 11. Find longest word

function longestWord(sentence) {

let words = sentence.split(' ');

let longest = '';

for (let word of words) {

if (word.length > longest.length) longest = word;

}

return longest;

}

// 12. Check if string has all unique characters

function hasUniqueChars(str) {

let set = new Set();

for (let char of str) {

if (set.has(char)) return false;

set.add(char);

}

return true;

}

// 13. Merge two sorted arrays

function mergeSortedArrays(arr1, arr2) {

let merged = [], i = 0, j = 0;

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

if (arr1[i] < arr2[j]) merged.push(arr1[i++]);

else merged.push(arr2[j++]);

}

return merged.concat(arr1.slice(i)).concat(arr2.slice(j));

}

// 14. First non-repeated character

function firstNonRepeatedChar(str) {

for (let char of str) {

if (str.indexOf(char) === str.lastIndexOf(char)) return char;

}

return null;

}

// 15. Remove duplicates from array

function removeDuplicates(arr) {

return [...new Set(arr)];

}

// 16. Binary search

function binarySearch(arr, target) {

let left = 0, right = arr.length - 1;

while (left <= right) {

let mid = Math.floor((left + right) / 2);

if (arr[mid] === target) return mid;

else if (arr[mid] < target) left = mid + 1;

else right = mid - 1;

}

return -1;

}

// 17. GCD

function gcd(a, b) {

return b === 0 ? a : gcd(b, a % b);

}

// 18. Find missing number in 1 to n

function findMissingNumber(arr, n) {

let expectedSum = (n \* (n + 1)) / 2;

let actualSum = arr.reduce((acc, num) => acc + num, 0);

return expectedSum - actualSum;

}

// 19. Second largest number

function secondLargest(arr) {

let first = -Infinity, second = -Infinity;

for (let num of arr) {

if (num > first) {

second = first;

first = num;

} else if (num > second && num !== first) {

second = num;

}

}

return second;

}

// 20. Roman numeral to integer

function romanToInt(s) {

const map = {I: 1, V: 5, X: 10, L: 50, C: 100, D: 500, M: 1000};

let total = 0;

for (let i = 0; i < s.length; i++) {

let curr = map[s[i]], next = map[s[i + 1]];

total += curr < next ? -curr : curr;

}

return total;

}

// 21. Intersection of arrays

function arrayIntersection(arr1, arr2) {

let set = new Set(arr2);

return arr1.filter(item => set.has(item));

}

// 22. Valid palindrome

function isValidPalindrome(str) {

let cleaned = str.toLowerCase().replace(/[^a-z0-9]/g, '');

return cleaned === reverseString(cleaned);

}

// 23. Sum of numbers excluding negatives

function sumPositiveNumbers(arr) {

return arr.filter(num => num >= 0).reduce((acc, num) => acc + num, 0);

}

// 24. Most frequent element

function mostFrequentElement(arr) {

const freq = {};

let maxCount = 0, maxElement = null;

for (let num of arr) {

freq[num] = (freq[num] || 0) + 1;

if (freq[num] > maxCount) {

maxCount = freq[num];

maxElement = num;

}

}

return maxElement;

}

// 25. Simple hash table

class SimpleHashTable {

constructor() {

this.table = {};

}

insert(key, value) {

this.table[key] = value;

}

get(key) {

return this.table[key];

}

remove(key) {

delete this.table[key];

}

}