1. Counting the Number of Times Something Occurs (Frequency Counter) in JS :-

    const reactions = ["like","love","like","wow","like","love"];

    const frequency = {};

    for(let reaction of reactions){

        if(frequency[reaction]){

            frequency[reaction] ++1;

        } else {

            frequency[reaction] = 1;

        }

    }

    console.log(frequency);

    // Output: {like: 3, love: 2, wow: 1}

2. Counting the Number of Times Something Occurs (Frequency Counter) in Python:-

    # Example list

    data = ['apple', 'banana', 'apple', 'orange', 'banana', 'apple']

    frequency = {}

    for item in data:

      if item in frequency:

        frequency[item] += 1

      else:

        frequency[item] = 1

        OR

    for item in data:

        frequency[item] = frequency.get(item, 0) + 1

3. from collections import Counter:-

    # Example sentence

    sentence = "this is a test this is only a test"

    # Split sentence into words

    words = sentence.split()

    # Count word frequencies

    word\_frequency = Counter(words)

    print(word\_frequency)

    # Output: Counter({'this': 2, 'is': 2, 'a': 2, 'test': 2, 'only': 1})

4. Finding the Higest and lowest values:-

    const prices = [10, 20, 30, 40, 50];

    let maxPrice = prices[0];

    let minPrice = prices[0];

    for(let price of prices){

        if(price > maxPrice){

            maxPrice = price;

        } else if(price < minPrice){

            minPrice = price;

        }

    }

    // Second highest

    let sh = minPrice;

    // Applying the same logic as heighest price to find the second

    //heighest price just skipping heighest value by saying < Heighest value

    for (let price in prices){

        if(price < maxPrice && price > sh){

            sh = price;

    }

}

    // second lowest values

    let sl = maxPrice;

    // Applying the same logic as lowest price to find the second

    // lowest price just skipping lowest value by saying > lowest value

    for (let price in prices){

        if(price > minPricePrice && price < sl){

            sl = price;

    }

}

    console.log(maxPrice); // Output: 50

    console.log(minPrice); // Output: 10

    console.log(sh); // Output: 40

    console.log(sl); // Output: 20

5. Solving the Two Sum Problem in JS:-

    const arr = [2, 4, 6, 7];

    const target = 9;

    const num = {}

    function twosum(arr, target){

        for(let i = 0; i < arr.length-1; i++){

            for(let j = i+1; j < arr.length; j++){

                if(arr[i] + arr[j] === target){

                    num[arr[i]] = arr[j];

                }

            }

            return num;

        }

    }

    OR

    const arr = [2, 4, 6, 7];

    const target = 9;

    function twoSum(arr, target){

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

            const diff = target - arr[i];

            debugger;

            if(arr.includes(diff)){

                return [arr[i], diff];

            }

    }

}

6. Solving the Two Sum Problem in Python:-

    def two\_sum(arr, target):

        num = {}

        for i in range(len(arr)):

            for j in range(i+1, len(arr)):

                if arr[i] + arr[j] == target:

                    num[arr[i]] = arr[j]

        return num

    arr = [2, 4, 6, 7, 5]

    target = 9

    print(two\_sum(arr, target))

    # Output: {2: 7, 4: 5}

7. Print the duplicate element in an array:-

    import java.util.HashSet;

    public class Main {

        public static void main(String[] args){

            int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 23};

            HashSet<Integer> set = new HashSet<>();

            HashSet<Integer> duplicate = new HashSet<>();

            System.out.println("Duplicate elements in given array: ");

            for(int i=0; i<arr.length; i++) {

                if(set.add(arr[i]) == false){ // If the value is already present in the set then it returns false

                    System.out.println(duplicate.add(arr[i]));

            }

            System.out.println(set); // Unique elements

            System.out.println(duplicate);// Duplicate elements

// Add method returns only unique values, if the value is already present in the set then it returns false,

        }

    }

}

// Output: Duplicate elements in given array:

//set: [1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 23]

//duplicate: [1, 2, 3, 4, 5, 6, 7, 8, 9]

8. Print the elements of an array in reverse order:-

public class Main {

    public static void main(String[] args){

        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9};

        System.out.println("Elements of the array in reverse order: ");

        for (int i= arr.length-1; i>=0; i--){

            System.out.println(arr[i]);

        }

    }

}

9. Print the largest element in an array:-

public class Main {

    public static void main(String[] args){

        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9};

        int max = arr[0];

        for(int i=1; i<arr.length; i++){

            if(arr[i] > max){

                max = arr[i];

            }

        }

        System.out.println("Largest element in the array is: " + max);

    }

}

10. Print the sum of all elements in an array:-

public class Main {

    public static void main(String[] args){

        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9};

        int sum = 0;

        for(int num: arr){

            sum += num;

        }

        System.out.println("Sum of all elements in the array is: " + sum);

    }

}

11. Bubble Sort:-

// in Bubble sort every element is compared with its adjacent

//elemet. Last elemt if fixed so every iteration j has lesser numebrs

public class Main{

    public static void Main(String[] args){

    int[] arr = {5,2,9,5,6,1};

    System.out.println("Array after Bubble Sort:");

    bubbleSort(arr);

    for(int num: arr){

        System.out.println(num + " ");

    }

    System.out.println(arr);

    public static void bubbleSort(int[] arr){

    for(int i=0, i< arr.length-1, i++){ // Number of passes/iterations

        for(int j=0, i<= arr.length-1-i, j++){ // Adjaent element comparison

            if(arr[j] >arr[j+1]){

                int temp = arr[j];

                arr[j] = arr[j+1];

                arr[j+1] = temp

                }

            }

        }

    }

}

}

// Output: Array after Bubble Sort:

//arr: [1, 2, 5, 5, 6, 9]

12. Selection Sort:-

public class Main {

    public static void main(String[] args) {

        int[] arr = {5, 2, 9, 1, 5, 6};

        System.out.println("Array after Selection Sort:");

        selectionSort(arr);

        for (int num : arr) {

            System.out.print(num + " ");

        }

    }

    public static void selectionSort(int[] arr) {

        int n = arr.length;

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

            int minIndex = i;

            for (int j = i + 1; j < n; j++) {

                if (arr[j] < arr[minIndex]) {

                    minIndex = j;

                }

            }

            int temp = arr[i];

            arr[i] = arr[minIndex];

            arr[minIndex] = temp;

        }

    }

}

13. Write code in Java to prove string objects are immutable

    public class Main {

    public static void main(String[] args) {

        String str = "Hello World!";

        String result = str.replaceAll("\\s", "");

        System.out.println("String without white spaces: " + result);

    }

}

14. Print the first letter of each word in the string

public class Main {

    public static void main(String[] args) {

        String str = "Hello World Example";

        String[] words = str.split(" ");

        System.out.print("First letter of each word: ");

        for (String word : words) {

            System.out.print(word.charAt(0) + " ");

        }

    }

}

15. Given a string find the length of the longest substring without repeating characters // This is difficult

import java.util.HashSet;

public class Main {

    public static void main(String[] args) {

        String str = "abcabcbb";

        System.out.println("Length of the longest substring without repeating characters: " + longestUniqueSubstring(str));

    }

    public static int longestUniqueSubstring(String str) {

        HashSet<Character> set = new HashSet<>();

        int maxLength = 0;

        int left = 0;

        for (int right = 0; right < str.length(); right++) {

            while (set.contains(str.charAt(right))) {

                set.remove(str.charAt(left));

                left++;

            }

            set.add(str.charAt(right));

            maxLength = Math.max(maxLength, right - left + 1);

        }

        return maxLength;

    }

}

16. Check if a string is a palindrome using a function

    public class Main {

        public static void main(String[] args) {

            String str = "madam";

            if (isPalindrome(str)) {

                System.out.println(str + " is a palindrome.");

            } else {

                System.out.println(str + " is not a palindrome.");

            }

        }

        // Function to check if a string is a palindrome

        public static boolean isPalindrome(String str) {

            int left = 0;

            int right = str.length() - 1;

            while (left < right) {

                if (str.charAt(left) != str.charAt(right)) {

                    return false;  // If characters do not match, it's not a palindrome

                }

                left++;

                right--;

            }

            return true;  // If all characters match, it's a palindrome

        }

    }

    17. Reverse a string using a function

    public class Main {

        public static void main(String[] args) {

            String str = "Hello World!";

            String reversedStr = reverseString(str);

            System.out.println("Reversed string: " + reversedStr);

        }

        // Function to reverse a string

        public static String reverseString(String str) {

            StringBuilder sb = new StringBuilder(str);

            return sb.reverse().toString();

        }

    }

18. Reverse a string(Without using a function)

    public class Main {

        public static void main(String[] args) {

            String str = "Hello World!";

            String reversedStr = "";

            // Reverse the string manually

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

                reversedStr += str.charAt(i);  // Append characters in reverse order

            }

            System.out.println("Reversed string: " + reversedStr);

        }

    }