

JAVA:-

1.Create an array with the values (1, 2, 3, 4, 5, 6, 7) and shuffle it.

```
import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.List;

public class ShuffleArrayExample {

    public static void main(String[] args) {

        // Create an array with the values (1, 2, 3, 4, 5, 6, 7)

        Integer[] array = {1, 2, 3, 4, 5, 6, 7};

        // Convert the array to a list

        List<Integer> list = new ArrayList<>(Arrays.asList(array));

        // Shuffle the list using Collections.shuffle

        Collections.shuffle(list);

        // Convert the shuffled list back to an array

        Integer[] shuffledArray = list.toArray(new Integer[0]);

        // Print the shuffled array

        System.out.println("Shuffled Array: " + Arrays.toString(shuffledArray));

    }

}
```

2. Enter a Roman Number as input and convert it to an integer. (Example: IX = 9)

```
import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;
```

```

public class RomanToIntegerConverter {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a Roman numeral

        System.out.print("Enter a Roman Numeral: ");

        String romanNumeral = scanner.nextLine().toUpperCase(); // Convert to uppercase for case-
insensitivity

        // Convert Roman numeral to integer

        int result = romanToInteger(romanNumeral);

        // Display the result

        System.out.println("Integer Equivalent: " + result);
    }

    // Function to convert a Roman numeral to an integer

    private static int romanToInteger(String s) {

        Map<Character, Integer> romanMap = new HashMap<>();

        // Initialize the Roman numeral to integer mapping

        romanMap.put('I', 1);

        romanMap.put('V', 5);

        romanMap.put('X', 10);

        romanMap.put('L', 50);

        romanMap.put('C', 100);

        romanMap.put('D', 500);
    }
}

```

```

romanMap.put('M', 1000);

int result = 0;

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

    char currentSymbol = s.charAt(i);

    int currentValue = romanMap.get(currentSymbol);

    // Check if subtractive notation is used (e.g., IV for 4, IX for 9)
    if (i < s.length() - 1) {

        char nextSymbol = s.charAt(i + 1);

        int nextValue = romanMap.get(nextSymbol);

        // If the next value is greater than the current value, subtract the current value
        if (nextValue > currentValue) {

            result -= currentValue;

        } else {

            result += currentValue;

        }

    } else {

        result += currentValue;

    }

}

return result;

}
}

```

3. Check if the input is pangram or not. (A pangram is a sentence that contains all the alphabets from A to Z)

```
import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class PangramChecker {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a sentence

        System.out.print("Enter a sentence: ");

        String input = scanner.nextLine().toLowerCase(); // Convert to lowercase for case-insensitivity

        // Check if the input is a pangram

        boolean isPangram = checkPangram(input);

        // Display the result

        if (isPangram) {

            System.out.println("The input is a pangram.");

        } else {

            System.out.println("The input is not a pangram.");

        }

    }

}

// Function to check if a string is a pangram

private static boolean checkPangram(String s) {

    Set<Character> alphabetSet = new HashSet<>();

    // Iterate through each character in the string

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

        char ch = s.charAt(i);
```

```

        // Check if the character is an alphabet letter
        if (Character.isLetter(ch)) {
            alphabetSet.add(ch);
        }
    }

    // Check if the set of unique alphabet letters contains all 26 letters
    return alphabetSet.size() == 26;
}
}

```

JAVASCRIPT:-

1. Take a sentence as an input and reverse every word in that sentence.
 Example - This is a sunny day > shiT si a ynnus yad.

```

function reverseWords(sentence) {

    // Split the sentence into an array of words

    let words = sentence.split(' ');


    // Reverse each word in the array

    let reversedWords = words.map(word => reverseString(word));


    // Join the reversed words back into a sentence

    let reversedSentence = reversedWords.join(' ');


    return reversedSentence;
}


// Function to reverse a string

```

```
function reverseString(str) {  
    return str.split("").reverse().join("");  
}
```

// Example usage

```
let inputSentence = "This is a sunny day";  
let result = reverseWords(inputSentence);  
console.log(result);
```

2. Perform sorting of an array in descending order.

// Function to perform sorting in descending order

```
function sortDescending(arr) {  
    return arr.sort(function(a, b) {  
        // Compare b to a for descending order  
        return b - a;  
    });  
}
```

// Example usage

```
let numbers = [5, 2, 8, 1, 4];  
let sortedDescending = sortDescending(numbers);  
console.log("Original Array:", numbers);  
console.log("Sorted in Descending Order:", sortedDescending);
```

HTML:-

1. Create a basic calculator using HTML, CSS, and JavaScript with the functionality of add, subtract, multiply and divide. Use the following picture for reference

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

font-family: Arial, sans-serif;

display: flex;

align-items: center;

justify-content: center;

height: 100vh;

margin: 0;

}

#calculator {

border: 1px solid #ccc;

border-radius: 5px;

padding: 20px;

text-align: center;

}

input {

width: 100%;

margin-bottom: 10px;

padding: 10px;

box-sizing: border-box;

```
}
```

```
button {
```

```
    width: 48px;
```

```
    height: 48px;
```

```
    margin: 5px;
```

```
    font-size: 18px;
```

```
    cursor: pointer;
```

```
}
```

```
</style>
```

```
<title>Basic Calculator</title>
```

```
</head>
```

```
<body>
```

```
    <div id="calculator">
```

```
        <input type="text" id="display" readonly>
```

```
        <br>
```

```
        <button onclick="appendToDisplay('1')">1</button>
```

```
        <button onclick="appendToDisplay('2')">2</button>
```

```
        <button onclick="appendToDisplay('3')">3</button>
```

```
        <button onclick="appendToDisplay('+')">+</button>
```

```
        <br>
```

```
        <button onclick="appendToDisplay('4')">4</button>
```

```
        <button onclick="appendToDisplay('5')">5</button>
```

```
        <button onclick="appendToDisplay('6')">6</button>
```

```
        <button onclick="appendToDisplay('-')">-</button>
```



```
<br>

<button onclick="appendToDisplay('7')">7</button>

<button onclick="appendToDisplay('8')">8</button>

<button onclick="appendToDisplay('9')">9</button>

<button onclick="appendToDisplay('*')">*</button>

<br>

<button onclick="appendToDisplay('0')">0</button>

<button onclick="clearDisplay()">C</button>

<button onclick="calculate()">=</button>

<button onclick="appendToDisplay('/')">/</button>

</div>
```

```
<script>

function appendToDisplay(value) {

    document.getElementById('display').value += value;

}

function clearDisplay() {

    document.getElementById('display').value = '';

}

function calculate() {

    try {

        document.getElementById('display').value = eval(document.getElementById('display').value);

    } catch (error) {
```

```
        document.getElementById('display').value = 'Error';
    }
}
</script>
</body>
</html>
```