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.Collections;

import java.util.List;


public class ArrayShuffle {

public static void main(String[] args) {

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


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

Collections.shuffle(list);


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


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

}

}
```

Output: Shuffled Array: [5, 3, 1, 4, 6, 7, 2]

Description

- Create an array of integers containing values from 1 to 7.
- Convert the array to an ArrayList to leverage the Collections.shuffle method.
- Use Collections.shuffle to randomly shuffle the elements in the ArrayList.
- Convert the shuffled ArrayList back to an array.
- Print the shuffled array to the console.

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

```
import java.util.Scanner;

public class RomanToInteger {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);
```

```
System.out.print("Enter a Roman numeral: ");
String romanNumeral = scanner.nextLine();
int result = romanToInt(romanNumeral);
System.out.println("Integer equivalent: " + result);
}

public static int romanToInt(String s) {
int result = 0;
for (int i = 0; i < s.length(); i++) {
char currentChar = s.charAt(i);
int current = charToValue(currentChar);
if (i < s.length() - 1) {
char nextChar = s.charAt(i + 1);
int next = charToValue(nextChar);
if (current < next) {
result -= current;
} else {
result += current;
}
} else {
result += current;
}
}
return result;
}

public static int charToValue(char c) {
switch (c) {
case 'I': return 1;
case 'V': return 5;
case 'X': return 10;
case 'L': return 50;

case 'C': return 100;

case 'D': return 500;

case 'M': return 1000;

default: return 0;

}

}

}
```

Output:Enter a Roman numeral: IX

Integer equivalent: 9

Description

- Input Roman Numeral: The program first prompts the user to enter a Roman numeral using the Scanner class. The entered Roman numeral is stored as a string.
- Roman to Integer Conversion: The romanToInt method is used to convert the Roman numeral to its integer equivalent. It processes the Roman numeral character by character, considering the Roman numeral rules (e.g., "IV" represents 4, "IX" represents 9).
- Character-to-Value Mapping: The charToValue method provides a mapping from Roman numeral characters to their corresponding integer values.
- Calculation of Integer Equivalent: The program iterates through the Roman numeral string, calculating the integer equivalent by considering the values of individual characters and their positions in the Roman numeral.
- Display Result: The final integer equivalent is displayed in the console.

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

```
public class PangramChecker {

public static void main(String[] args) {

String input = "The quick brown fox jumps over the lazy dog";

boolean isPangram = checkIfPangram(input);

if (isPangram) {

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

} else {

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

}

}
public static boolean checkIfPangram(String str) {
str = str.toLowerCase();
for (char ch = 'a'; ch <= 'z'; ch++) {
if (str.indexOf(ch) == -1) {
return false;
}
}
return true;
}
}
```

Output: The input is a pangram.

Description

- Input String: The program takes an input string, which is expected to be a sentence or a sequence of words.
- Pangram Check: The goal of this program is to check if the input string is a pangram. A pangram is a sentence that contains every letter of the alphabet at least once.
- Case Insensitive: To simplify the check, the input string is converted to lowercase using the toLowerCase method. This ensures that both uppercase and lowercase letters are treated the same way.
- Character-by-Character Check: The program then iterates through the alphabet, from 'a' to 'z', checking if each letter is present in the converted input string using the indexOf method. If any letter is not found, it returns false immediately.
- Pangram Result: If all the letters of the alphabet are found in the input string, the program concludes that the input is a pangram and returns true. Otherwise, it returns false.
- Display Result: The result, whether the input is a pangram or not, is displayed in the console.

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 reverseWordsInSentence(sentence) {
const words = sentence.split(' ');
const reversedWords = words.map(word => {
return word.split('').reverse().join('');
});
const reversedSentence = reversedWords.join(' ');
return reversedSentence;
}
const inputSentence = "This is a sunny day";
const reversed = reverseWordsInSentence(inputSentence);
console.log(reversed);
```

Output: "sihT si a ynnus yad"

Description

The provided JavaScript code defines a function called reverseWordsInSentence. This function takes a sentence as input, splits it into individual words, reverses the characters within each word, and then reassembles the sentence with the reversed words. The reversed sentence is then returned.The code includes a sample input sentence, "This is a sunny day," and demonstrates how to use the function to reverse the words within it. The result is logged to the console.

2. Perform sorting of an array in descending order.

```
function sortDescending(input) {
if (!input) {
return "No input provided.";
}
const inputArray = input.split(" ");
const numbers = [];
for (let i = 0; i < inputArray.length; i++) {
const num = parseFloat(inputArray[i]);
if (!isNaN(num)) {
numbers.push(num);
}
}
if (numbers.length === 0) {
return "No valid numbers found in the input.";
}
numbers.sort(function(a, b) {
return b - a;
});
return "Sorted in descending order: " + numbers.join(", ");
}
const input = prompt("Enter numbers separated by spaces:");
const result = sortDescending(input);
console.log(result);
```

Output: Sorted in descending order: 67, 45, 21, 12, 9, 3.14

Description

- Input Validation:It first checks if there's any input provided. If the input is empty or not provided, it returns "No input provided."
- Parsing and Filtering Numbers:The input is split into an array of strings using spaces as the delimiter.It iterates through the array, attempting to convert each element to a floating-point number (using parseFloat).If a valid number is found, it is added to the numbers array. Non-numeric elements are ignored.
- Handling No Valid Numbers:If there are no valid numbers found in the input, it returns "No valid numbers found in the input."
- Sorting in Descending Order:The numbers array, which contains the valid numeric values, is sorted in descending order using a custom comparison function. This function sorts the numbers from highest to lowest.
- Result:The function returns a string that states the numbers sorted in descending order, with the sorted numbers joined together and separated by commas.
- User Interaction:The code then prompts the user to enter a list of numbers separated by spaces using the prompt function.
- Sorting and Display:The sortDescending function is called with the user's input, and the result is stored in the result variable.

The sorted result is displayed, indicating that the numbers have been sorted in descending order.

HTML

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

```
<!DOCTYPE html>
<html>
<head>
<title>Calculator</title>
<style>
body {
font-family: Arial, sans-serif;
```

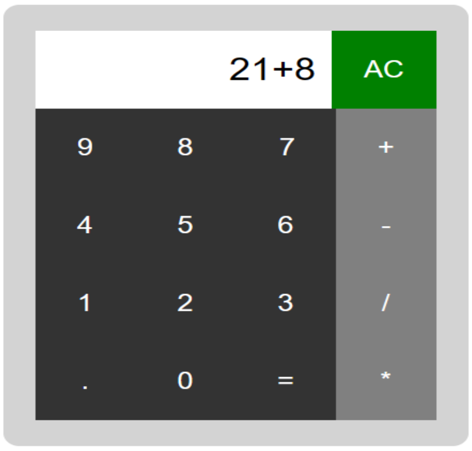
```
}

.calculator {
width: 250px;
background: lightgray;
 padding: 20px;
border-radius: 10px;
margin: 0 auto;
text-align: center;       }

.input-container {
display: flex;
align-items: center;
justify-content: space-between;
margin: 0;       }

.input-text {
color: #000;
font-size: 24px;
text-align: right;
background: white;
border: none;
width: 70%;
padding: 0 10px;
outline: none;
height: 60px;
}

.button {
background: #333;
color: #fff;
```

```
font-size: 18px;
padding: 10px;
border: none;
border-radius: 0;
cursor: pointer;
height: 60px;
margin: 0;
}
.ac-button {
background: green;
color: #fff;
font-size: 18px;
padding: 10px 20px;
border: none;
border-radius: 0;
cursor: pointer;
margin: 0;
}
.gray-bg-button {
background: gray;
}
.button-container {
display: grid;
grid-template-columns: repeat(4, 1fr);
gap: 0;
}

.equals-button {
background: #333;
}
```

```
.button:nth-child(3) {
margin-right: -2px;
}
</style>
</head>
<body>
<div class="calculator">
<div class="input-container">
<input class="input-text" id="result" readonly>
<button class="button ac-button" onclick="clearResult()">AC</button>
</div>
<div class="button-container">
<button class="button" onclick="appendToResult(9)">9</button>
<button class="button" onclick="appendToResult(8)">8</button>
<button class="button" onclick="appendToResult(7)">7</button>
<button class="button gray-bg-button" onclick="appendToResult('+')">+</button>
<button class="button" onclick="appendToResult(4)">4</button>
<button class="button" onclick="appendToResult(5)">5</button>
<button class="button" onclick="appendToResult(6)">6</button>
<button class="button gray-bg-button" onclick="appendToResult('-')">-</button>
<button class="button" onclick="appendToResult(1)">1</button>
<button class="button" onclick="appendToResult(2)">2</button>
<button class="button" onclick="appendToResult(3)">3</button>
<button class="button gray-bg-button" onclick="appendToResult('/')">/</button>
<button class="button" onclick="appendToResult('.')">.</button>
<button class="button zero-button" onclick="appendToResult(0)">0</button>
<button class="button equals-button" onclick="calculateResult()">=</button>
<button class="button gray-bg-button" onclick="appendToResult('*')">*</button>
</div>
</div>
```

```
<script>
function clearResult() {
document.getElementById("result").value = "";
}

function appendToResult(value) {
document.getElementById("result").value += value;
}

function calculateResult() {
try {
const result = eval(document.getElementById("result").value);
document.getElementById("result").value = result;
} catch (error) {
document.getElementById("result").value = "Error";
}
}
</script>
</body>
</html>
```

Output:



2. Create a survey form with Fields; First Name, Last Name, Date of Birth, Country (dropdown), Gender (checkbox), Profession, email, and mobile number. All the input fields are necessary to submit the form. Create two buttons Submit and Reset. Reset will reset the form while clicking on submit, first, it will check all the fields and necessary validations and then a popup will appear displaying all the selected values with labels in front of it. On closing the popup, the form should reset all the values. Use the following image for reference

```
<!DOCTYPE html>
<html>
<head>
<title>Survey Form</title>
<style>
body {
font-family: Arial, sans-serif;

background-color: lightblue;
}
.form-container {
background-color: #f0f0f0;
max-width: 400px;
margin: 20px auto;
padding: 20px;
border: 1px solid #ccc;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.form-field {
margin-bottom: 10px;
}
.custom-button {
background-color: blue;
color: white;
padding: 10px 20px;
border: none;
border-radius: 5px;
cursor: pointer;
}
.custom-button:hover {
background-color: darkblue;
}
button:hover {
background-color: darkblue;
}
.popup {
display: none;
position: fixed;
top: 0;
left: 0;
width: 100%;
height: 100%;
background: rgba(0, 0, 0, 0.7);
align-items: center;
justify-content: center;
text-align: center;
}
.popup-content {
background: #fff;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
padding: 20px;
}
h2 {
text-align: center;
}
```

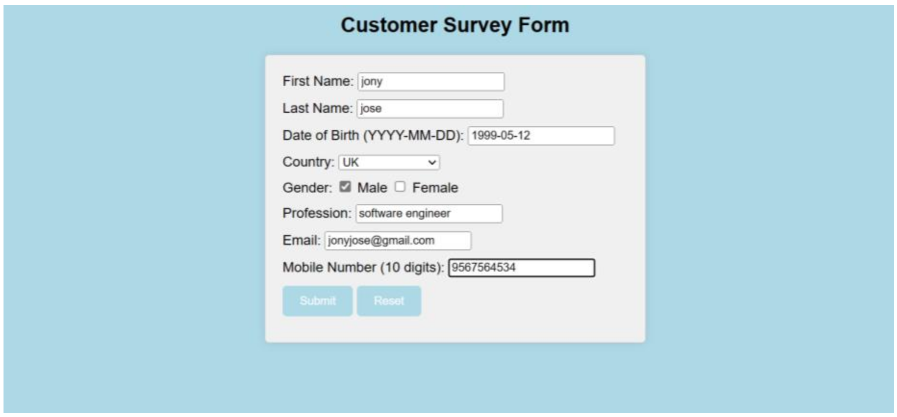
```
</style>
</head>
<h2 > Customer Survey Form</h2>
<body>
<div class="form-container">
<form id="surveyForm" onsubmit="return validateForm()">
<div class="form-field">
<label for="firstName">First Name:</label>
<input type="text" id="firstName" required>
</div>
<div class="form-field">
<label for="lastName">Last Name:</label>
<input type="text" id="lastName" required>
</div>
<div class="form-field">
<label for="dob">Date of Birth (YYYY-MM-DD):</label>
<input type="text" id="dob" pattern="[0-9]{4}-[0-9]{2}-[0-9]{2}" required>
</div>
<div class="form-field">
<label for="country">Country:</label>
<select id="country" required>
<option value="">Select a country</option>
<option value="USA">USA</option>
<option value="Canada">Canada</option>
<option value="UK">UK</option>
<option value="France">France</option>
<option value="Germany">Germany</option>
<option value="Japan">Japan</option>
<option value="Australia">Australia</option>
<option value="India">India</option>
<option value="Brazil">Brazil</option>
```

```
<option value="China">China</option>

<option value="South Africa">South Africa</option>

<option value="Mexico">Mexico</option>

</select>

</div>

<div class="form-field">

<label>Gender:</label>

<input type="checkbox" id="male" name="gender" value="Male"> <label for="male">Male</label>

<input type="checkbox" id="female" name="gender" value="Female"> <label
for="female">Female</label>

</div>

<div class="form-field">

<label for="profession">Profession:</label>

<input type="text" id="profession" required>

</div>

<div class="form-field">

<label for="email">Email:</label>

<input type="email" id="email" required>

</div>

<div class="form-field">

<label for="mobile">Mobile Number (10 digits):</label>

<input type="tel" id="mobile" pattern="[0-9]{10}" required>

</div>

<div class="form-field">

<button type="submit" style="background-color: lightblue; color: white; padding: 10px 20px; border:
none; border-radius: 5px; cursor: pointer;">

Submit

</button>

<button type="button" style="background-color: lightblue; color: white; padding: 10px 20px; border:
none; border-radius: 5px; cursor: pointer;" onclick="resetForm()">

Reset

</button>
```

```
</div>
</form>
</div>
<div id="popup" class="popup">
<div class="popup-content">
<h2>Survey Submitted</h2>
<p>First Name: <span id="popupFirstName"></span></p>
<p>Last Name: <span id="popupLastName"></span></p>
<p>Date of Birth: <span id="popupDob"></span></p>
<p>Country: <span id="popupCountry"></span></p>
<p>Gender: <span id="popupGender"></span></p>
<p>Profession: <span id="popupProfession"></span></p>
<p>Email: <span id="popupEmail"></span></p>
<p>Mobile Number: <span id="popupMobile"></span></p>
<button onclick="closePopup()">Close</button>
</div>
</div>
<script>
function validateForm() {
const firstName = document.getElementById("firstName").value;
const lastName = document.getElementById("lastName").value;
const dob = document.getElementById("dob").value;
const country = document.getElementById("country").value;
const gender = getSelectedGender();
const profession = document.getElementById("profession").value;
const email = document.getElementById("email").value;
const mobile = document.getElementById("mobile").value;
if (!firstName || !lastName || !dob || country === "" || gender === "" || !profession || !email ||
!mobile) {
alert("Please fill out all required fields.");
return false;
}
const emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
if (!email.match(emailPattern)) {
alert("Please enter a valid email address.");
return false;
}
const mobilePattern = /^\d{10}$/;
if (!mobile.match(mobilePattern)) {
alert("Please enter a valid 10-digit mobile number.");
return false;
}
document.getElementById("popupFirstName").textContent = firstName;
document.getElementById("popupLastName").textContent = lastName;
document.getElementById("popupDob").textContent = dob;
document.getElementById("popupCountry").textContent = country;
document.getElementById("popupGender").textContent = gender;
document.getElementById("popupProfession").textContent = profession;
document.getElementById("popupEmail").textContent = email;
document.getElementById("popupMobile").textContent = mobile;
document.getElementById("popup").style.display = "block";
document.getElementById("surveyForm").reset();
return false;
}
function getSelectedGender() {
const checkboxes = document.querySelectorAll('input[name="gender"]:checked');
const selectedGender = Array.from(checkboxes).map(checkbox => checkbox.value);
return selectedGender.join(', ');
}
function resetForm() {
document.getElementById("surveyForm").reset();
}
```

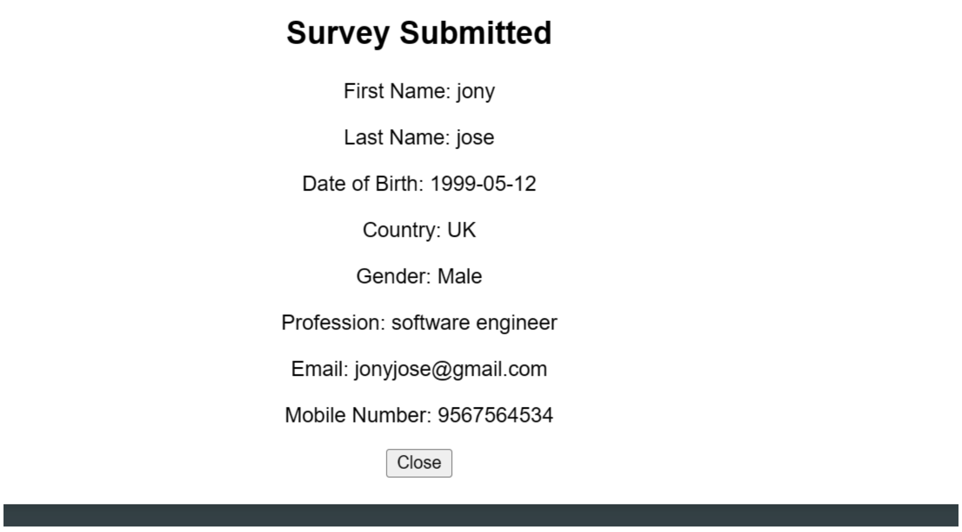
```
function closePopup() {
document.getElementById("popup").style.display = "none";
}
</script>
</body>
</html>
```

Output:



Customer Survey Form

First Name: jony
Last Name: jose
Date of Birth (YYYY-MM-DD): 1999-05-12
Country: UK
Gender: ☑ Male ☐ Female
Profession: software engineer
Email: jonyjose@gmail.com
Mobile Number (10 digits): 9567564534



Survey Submitted

First Name: jony
Last Name: jose
Date of Birth: 1999-05-12
Country: UK
Gender: Male
Profession: software engineer
Email: jonyjose@gmail.com
Mobile Number: 9567564534