

**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 ShuffleArray {

    public static void main(String[] args) {

        // Create a list with the values (1, 2, 3, 4, 5, 6, 7)
        List<Integer> myList = new ArrayList<>();

        myList.add(1);
        myList.add(2);
        myList.add(3);
        myList.add(4);
        myList.add(5);
        myList.add(6);
        myList.add(7);


        // Shuffle the list
        Collections.shuffle(myList);


        // Convert the shuffled list back to an array
        Integer[] shuffledArray = myList.toArray(new Integer[myList.size()]);

        // Print the shuffled array
        for (int i = 0; i < shuffledArray.length; i++) {

            System.out.print(shuffledArray[i]);

            if (i < shuffledArray.length - 1) {

                System.out.print(", ");

            }

        }

    }

}
```

**Output:**

Output

Clear

```
java -cp /tmp/lcTcOj4nym ShuffleArray  
3, 1, 6, 7, 2, 5, 4
```

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

```
import java.util.HashMap;  
  
public class RomanToInteger {  
    public static void main(String[] args) {  
        // Define a HashMap to store Roman numerals and their corresponding values  
        HashMap<Character, Integer> romanMap = new HashMap<>();  
        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);  
  
        // Replace this string with your Roman numeral input  
        String romanNumeral = "IX"; // Example: IX  
  
        int result = 0;  
        int prevValue = 0;  
  
        // Iterate through the Roman numeral string from right to left  
        for (int i = romanNumeral.length() - 1; i >= 0; i--) {  
            char currentChar = romanNumeral.charAt(i);  
            int currentValue = romanMap.get(currentChar);  
  
            // Check if the current value is less than the previous value, subtract it  
            if (currentValue < prevValue) {  
                result -= currentValue;  
            } else {  
                // Otherwise, add it to the result  
                result += currentValue;  
            }  
            prevValue = currentValue;  
        }  
    }  
}
```

```

    }

    // Update the previous value for the next iteration
    prevValue = currentValue;
}
// Print the integer equivalent of the Roman numeral
System.out.println("Integer value of " + romanNumeral + " is: " + result);
}
}

```

**Output:**

Output

Clear

```

java -cp /tmp/lcTcOj4nym RomanToInteger
Integer value of IX is: 9

```

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) {
        // Replace this string with your input
        String input = "The quick brown fox jumps over the lazy dog";

        boolean isPangram = isPangram(input);

        if (isPangram) {
            System.out.println("The input is a pangram.");
        } else {
            System.out.println("The input is not a pangram.");
        }
    }

    public static boolean isPangram(String input) {
        // Convert the input to lowercase to ignore case
        input = input.toLowerCase();

        // Create a boolean array to mark the presence of each alphabet
        boolean[] alphabetPresent = new boolean[26];

        // Iterate through the input string

```

```
for (int i = 0; i < input.length(); i++) {
    char c = input.charAt(i);

    // Check if the character is an English alphabet letter
    if (c >= 'a' && c <= 'z') {
        int index = c - 'a'; // Calculate the index for the letter in the array
        alphabetPresent[index] = true; // Mark the alphabet as present
    }
}

// Check if all alphabet letters are present
for (boolean isPresent : alphabetPresent) {
    if (!isPresent) {
        return false; // If any alphabet is not present, it's not a pangram
    }
}

return true; // All alphabets are present, so it's a pangram
}
```

**Output:**

Output

Clear

```
java -cp /tmp/lcTc0j4nym PangramChecker
The input is a pangram.
|
```