VTHINK ANSWERS

Round 1: Online Assessment

1.

```
import java.util.Scanner;

public class ReverseString {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        String input = sc.nextLine();

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

        StringBuilder sb = new StringBuilder();

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

            sb.append(words[i]);

            if (i > 0) sb.append(" ");

        }

        System.out.println("Reversed String: " + sb.toString());

        sc.close();

    }

}
```

2.

```
import java.util.Scanner;
public class SumOfDigits {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string with digits: ");
        String input = sc.nextLine();
        int sum = 0;
        for (char c : input.toCharArray()) {
            if (Character.isDigit(c)) {
                sum += Character.getNumericValue(c);
            }
        }
        System.out.println("Sum of Digits: " + sum);
        sc.close();
    }
}
```

3.

```
import java.util.Scanner;
import java.util.Arrays;
public class AnagramCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = sc.nextLine();
        System.out.print("Enter second string: ");
        String str2 = sc.nextLine();
        char[] arr1 = str1.replaceAll("\\s", "").toLowerCase().toCharArray();
        char[] arr2 = str2.replaceAll("\\s", "").toLowerCase().toCharArray();
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        if (Arrays.equals(arr1, arr2)) {
            System.out.println("Are Anagrams: true");
        } else {
            System.out.println("Are Anagrams: false");
        }

        sc.close();
    }
}
```

4.

```
import java.util.*;
public class FirstNonRepeatingChar {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String input = sc.nextLine().toLowerCase();
        Map<Character, Integer> map = new LinkedHashMap<>();
        for (char c : input.toCharArray()) {
            map.put(c, map.getOrDefault(c, 0) + 1);
        }
        for (char c : map.keySet()) {
            if (map.get(c) == 1) {
                System.out.println("First non-repeating: " + c);
                return;
            }
        }
        System.out.println("No non-repeating character found.");
    }
}
```

5.

```
import java.util.*;
public class ReverseStringOrder {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a sentence:");
        String input = sc.nextLine();
        String[] words = input.split(" ");
        StringBuilder reversed = new StringBuilder();
        for (int i = words.length - 1; i >= 0; i--) {
            reversed.append(words[i]);
            if (i != 0) {
                reversed.append(" ");
            }
        }
        System.out.println("Reversed Order: " + reversed.toString());
    }
}
```

6.

```
import java.util.*;
public class CountVowels {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String input = sc.nextLine().toLowerCase();
        int count = 0;
        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                count++;
            }
        }
        System.out.println("Number of vowels: " + count);
    }
}
```

7.

```
import java.util.*;
public class RomanToNumber {
    int value(char r) {
        if (r == 'I') return 1;
        if (r == 'V') return 5;
        if (r == 'X') return 10;
        if (r == 'L') return 50;
        if (r == 'C') return 100;
        if (r == 'D') return 500;
        if (r == 'M') return 1000;
        return -1;
    }
    int romanToDecimal(String str) {
        int res = 0;
        for (int i = 0; i < str.length(); i++) {
            int s1 = value(str.charAt(i));
            if (i + 1 < str.length()) {
                int s2 = value(str.charAt(i + 1));
                if (s1 >= s2) {
                    res = res + s1;
                } else {
                    res = res + s2 - s1;
                    i++;
                }
            } else {
                res = res + s1;
            }
        }
        return res;
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        RomanToNumber ob = new RomanToNumber();
        System.out.print("Enter a Roman numeral: ");
        String str = sc.nextLine().toUpperCase().trim();
        System.out.println("Integer form of Roman Numeral is " + ob.romanToDecimal(str));
        sc.close();
    }
}
```

8.

```
import java.util.*;
public class ReverseEachWord {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String input = sc.nextLine();
        String[] words = input.split(" ");
        StringBuilder result = new StringBuilder();
        for (String word : words) {
            StringBuilder revWord = new StringBuilder(word);
            result.append(revWord.reverse().toString()).append(" ");
        }
        System.out.println("Reversed words: " + result.toString().trim());
    }
}
```

9.

```
import java.util.*;
public class LongestSubstring {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = sc.nextLine();

        System.out.println("Longest substring without repeating characters: " +
longestUniqueSubstring(s));
    }
    public static String longestUniqueSubstring(String s) {
        int n = s.length();
        Set<Character> set = new HashSet<>();
        int left = 0, right = 0;
        int maxLen = 0, start = 0;
        while (right < n) {
            char c = s.charAt(right);
            while (set.contains(c)) {
                set.remove(s.charAt(left));
                left++;
            }
            set.add(c);
            if (right - left + 1 > maxLen) {
                maxLen = right - left + 1;
                start = left;
            }
            right++;
        }
       return s.substring(start, start + maxLen);
    }
}
```

10.

```
import java.util.*;
public class CommonCharacters {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = sc.nextLine();
        System.out.print("Enter second string: ");
        String str2 = sc.nextLine();
        Set<Character> set1 = new HashSet<>();
        for (char c : str1.toCharArray()) {
            set1.add(c);
        }
        Set<Character> set2 = new HashSet<>();
        for (char c : str2.toCharArray()) {
            set2.add(c);
        }
        set1.retainAll(set2);
        if (set1.isEmpty()) {
            System.out.println("No common characters found.");
        } else {
            System.out.print("Common characters: ");
            for (char c : set1) {
                System.out.print(c + " ");
            }
        }
    }
}
```

11.

```
import java.util.*;
public class NonRepeatingCharacters {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = sc.nextLine();
        Map<Character, Integer> map = new LinkedHashMap<>();
        for (char c : input.toCharArray()) {
            map.put(c, map.getOrDefault(c, 0) + 1);
        }
        System.out.print("Non-repeating characters: ");
        for (char c : map.keySet()) {
            if (map.get(c) == 1) {
                System.out.print(c);
            }
        }
    }
}
```

Round 2: Virtual Interview

1.

```
import java.util.*;
public class CountCharacters {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = sc.nextLine();
        System.out.println("Number of characters: " + input.length());
    }
}
```

2.

```
import java.util.*;
public class ReplaceCharacter {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.nextLine();
        System.out.print("Enter the character to replace: ");
        char oldChar = sc.next().charAt(0);
        System.out.print("Enter the new character: ");
        char newChar = sc.next().charAt(0);
        String result = str.replace(oldChar, newChar);
        System.out.println("Modified string: " + result);
    }
}
```

3.

```
import java.util.*;
public class StringWithCommas {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.nextLine();
        String result = String.join(",", str.split(""));
        System.out.println("Output: " + result);
    }
}
```

4.

```
import java.util.Scanner;
public class ReverseWords {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = sc.nextLine();
        String[] words = input.split(" ");
        System.out.print("Reversed string: ");
        for (int i = words.length - 1; i >= 0; i--) {
            System.out.print(words[i] + " ");
        }
        sc.close();
    }
}
```

5.

```java
import java.util.Scanner;
public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        System.out.println("Multiplication Table of " + num);
        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " X " + i + " = " + (num * i));
        }
        sc.close();
    }
}
```

6.

```java
import java.util.Scanner;
public class StarPattern {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of rows: ");
        int rows = sc.nextInt();
        for (int i = rows; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

7.

```
import java.util.Scanner;
public class ReplaceSpaceWithUnderscore {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a string: ");
    String str = sc.nextLine();
    str = str.replace(" ", "_");
    System.out.println("Modified string: " + str);
    sc.close();
  }
}
```

8.

```
import java.util.Scanner;
public class PrintString {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a string: ");
    String str = sc.nextLine();
    System.out.println("You entered: " + str);
    sc.close();
  }
}
```

9.

```
import java.util.Scanner;
public class ReverseString {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.nextLine();
        String reversed = new StringBuilder(str).reverse().toString();
        System.out.println("Reversed string: " + reversed);
        sc.close();
    }
}
```

10.

```
import java.util.Scanner;
import java.util.LinkedHashSet;
public class RemoveDuplicates {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.nextLine();
        LinkedHashSet<Character> set = new LinkedHashSet<>();
        for (char c : str.toCharArray()) {
            set.add(c);
        }
        StringBuilder result = new StringBuilder();
        for (char c : set) {
            result.append(c);
        }
        System.out.println("String after removing duplicates: " + result.toString());
        sc.close();
    }
}
```