1)

```java
public class RemoveDuplicates {
    public static void main(String[] args) {
        String input = "Hello World";
        String result = removeDuplicates(input);
        System.out.println("String with duplicates removed: " + result);
    }
    public static String removeDuplicates(String input) {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < input.length(); i++) {
            char currentChar = input.charAt(i);
            if (sb.indexOf(String.valueOf(currentChar)) == -1) {
                sb.append(currentChar);
            }
        }
        return sb.toString();
    }
}
```

2)

```java
import java.util.HashMap;
import java.util.Map;
public class DuplicateCharacters {
    public static void main(String[] args) {
        String input = "Hello World";
        System.out.println("Duplicate characters in the string:");
        printDuplicateCharacters(input);
    }
    public static void printDuplicateCharacters(String input) {
        Map<Character, Integer> charCountMap = new HashMap<>();
        for (char c : input.toCharArray()) {
            charCountMap.put(c, charCountMap.getOrDefault(c, 0) + 1);
        }
        for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {
            if (entry.getValue() > 1) {
                System.out.println(entry.getKey());
            }
        }
    }
}
```

3)

```
public class PalindromeCheck {
    public static void main(String[] args) {
        String input = "2552";
        boolean isPalindrome = isPalindrome(input);
        System.out.println("Is \"" + input + "\" a palindrome? " + isPalindrome);
    }
    public static boolean isPalindrome(String input) {
        int left = 0;
        int right = input.length() - 1;
        while (left < right) {
            if (input.charAt(left) != input.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

4)

```
public class CharacterCount {
    public static void main(String[] args) {
        String input = "Hello World!";
        int vowelCount = 0;
        int consonantCount = 0;
        int specialCharCount = 0;
        for (char c : input.toLowerCase().toCharArray()) {
            if (isVowel(c)) {
                vowelCount++;
            } else if (isConsonant(c)) {
                consonantCount++;
            } else if (isSpecialCharacter(c)) {
                specialCharCount++;
            }
        }
        System.out.println("Number of vowels: " + vowelCount);
        System.out.println("Number of consonants: " + consonantCount);
        System.out.println("Number of special characters: " + specialCharCount);
    }
    public static boolean isVowel(char c) {
        return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
    }
    public static boolean isConsonant(char c) {
        return (c >= 'a' && c <= 'z') && !isVowel(c);
    }
    public static boolean isSpecialCharacter(char c) {
        return !Character.isLetterOrDigit(c) && !Character.isWhitespace(c);
    }
}
```

5)

```java
public class AnagramChecker {
    public static void main(String[] args) {
        String str1 = "listen";
        String str2 = "silent";
        boolean isAnagram = checkAnagram(str1, str2);
        System.out.println("Are \"" + str1 + "\" and \"" + str2 + "\" anagrams? " + isAnagram);
    }
    public static boolean checkAnagram(String str1, String str2) {
        if (str1.length() != str2.length()) {
            return false;
        }
        int[] charCount = new int[26];
        for (int i = 0; i < str1.length(); i++) {
            charCount[str1.charAt(i) - 'a']++;
            charCount[str2.charAt(i) - 'a']--;
        }
        for (int count : charCount) {
            if (count != 0) {
                return false;
            }
        }
        return true;
    }
}
```

6)

```java
public class PangramChecker {
    public static void main(String[] args) {
        String input = "The quick brown fox jumps over the lazy dog";
        boolean isPangram = checkPangram(input);
        System.out.println("Is the string a pangram? " + isPangram);

    }

    public static boolean checkPangram(String input) {

        int[] charCount = new int[26];

        int count = 0;

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

            char c = input.charAt(i);

            if (c >= 'A' && c <= 'Z') {

                c = (char) (c - 'A' + 'a'); // Convert to lowercase

            }

            if (c >= 'a' && c <= 'z') {

                if (charCount[c - 'a'] == 0) {

                    count++;

                }

                charCount[c - 'a']++;

            }

        }

        return count == 26;

    }

}
```

7)

```
public class UniqueCharacterChecker {

    public static void main(String[] args) {

        String input = "OpenAI";

        boolean hasUniqueCharacters = checkUniqueCharacters(input);

        System.out.println("Does the string contain all unique characters? " + hasUniqueCharacters);

    }

    public static boolean checkUniqueCharacters(String input) {

        boolean[] charSet = new boolean[256];
        for (int i = 0; i < input.length(); i++) {
            char c = input.charAt(i);
            if (charSet[c]) {
                return false;
            }
            charSet[c] = true;
        }
        return true;
    }
}
```

8)

```
public class MaxOccurringCharacter {
    public static void main(String[] args) {
        String input = "Hello World";
        char maxChar = findMaxOccurringCharacter(input);
        System.out.println("Maximum occurring character: " + maxChar);
    }
    public static char findMaxOccurringCharacter(String input) {
        int[] charCount = new int[256];
        int maxCount = 0;
        char maxChar = '\0';
        for (int i = 0; i < input.length(); i++) {
            char c = input.charAt(i);
            charCount[c]++;
            if (charCount[c] > maxCount) {
                maxCount = charCount[c];
                maxChar = c;
            }
        }
        return maxChar;
    }
}
```