Name – Aryan Singh

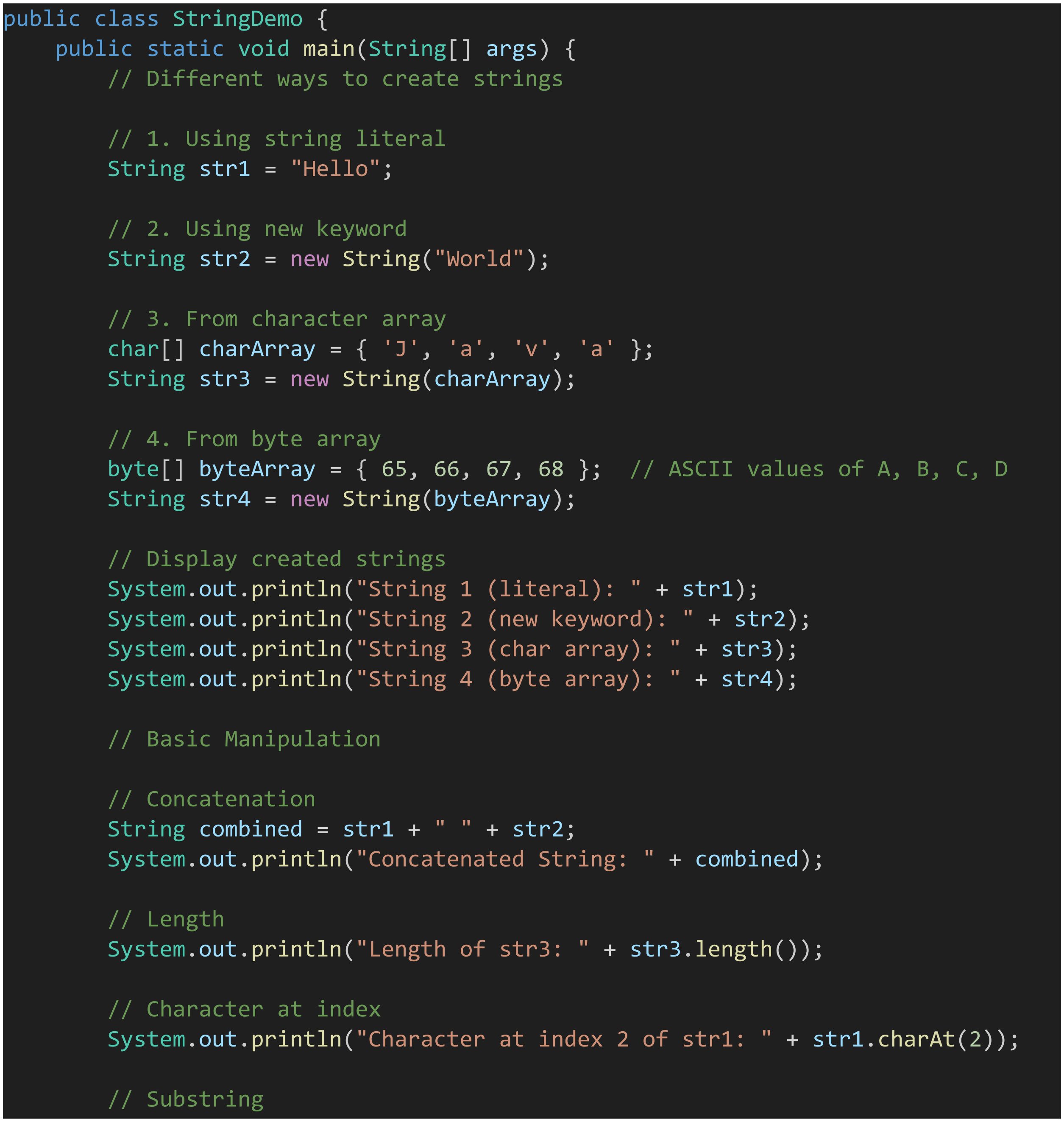
Reg no. – RA2411030010159

Y1 section
practice problem week 1

Q1. Create a program that demonstrates different ways to create strings and basic

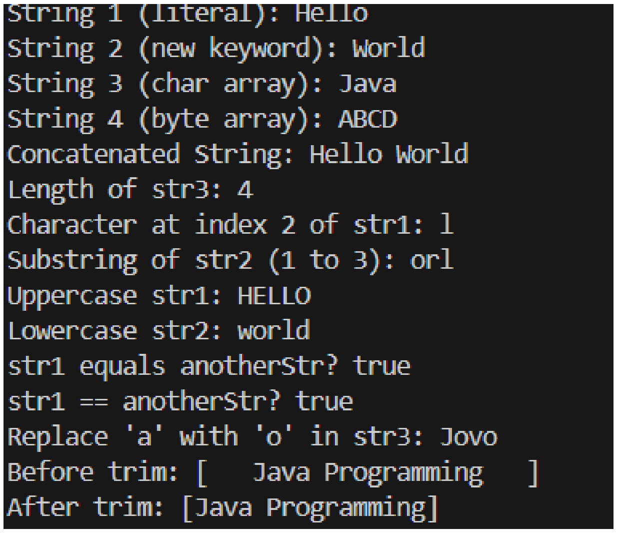
manipulation.

PROGRAM



```java
public class StringDemo {
    public static void main(String[] args) {
        // Different ways to create strings

        // 1. Using string literal
        String str1 = "Hello";

        // 2. Using new keyword
        String str2 = new String("World");

        // 3. From character array
        char[] charArray = { 'J', 'a', 'v', 'a' };
        String str3 = new String(charArray);

        // 4. From byte array
        byte[] byteArray = { 65, 66, 67, 68 };  // ASCII values of A, B, C, D
        String str4 = new String(byteArray);

        // Display created strings
        System.out.println("String 1 (literal): " + str1);
        System.out.println("String 2 (new keyword): " + str2);
        System.out.println("String 3 (char array): " + str3);
        System.out.println("String 4 (byte array): " + str4);

        // Basic Manipulation

        // Concatenation
        String combined = str1 + " " + str2;
        System.out.println("Concatenated String: " + combined);

        // Length
        System.out.println("Length of str3: " + str3.length());

        // Character at index
        System.out.println("Character at index 2 of str1: " + str1.charAt(2));

        // Substring
        System.out.println("Substring of str2 (1 to 3): " + str2.substring(1,
4));

        // Changing case
        System.out.println("Uppercase str1: " + str1.toUpperCase());
        System.out.println("Lowercase str2: " + str2.toLowerCase());

        // Equality check
        String anotherStr = "Hello";
        System.out.println("str1 equals anotherStr? " + str1.equals(anotherStr));
        System.out.println("str1 == anotherStr? " + (str1 == anotherStr)); //
reference check

        // Replace characters
        System.out.println("Replace 'a' with 'o' in str3: " + str3.replace('a',
'o'));

        // Trim spaces
        String spaced = "   Java Programming   ";
        System.out.println("Before trim: [" + spaced + "]");
        System.out.println("After trim: [" + spaced.trim() + "]");
    }

}
```

OUTPUT

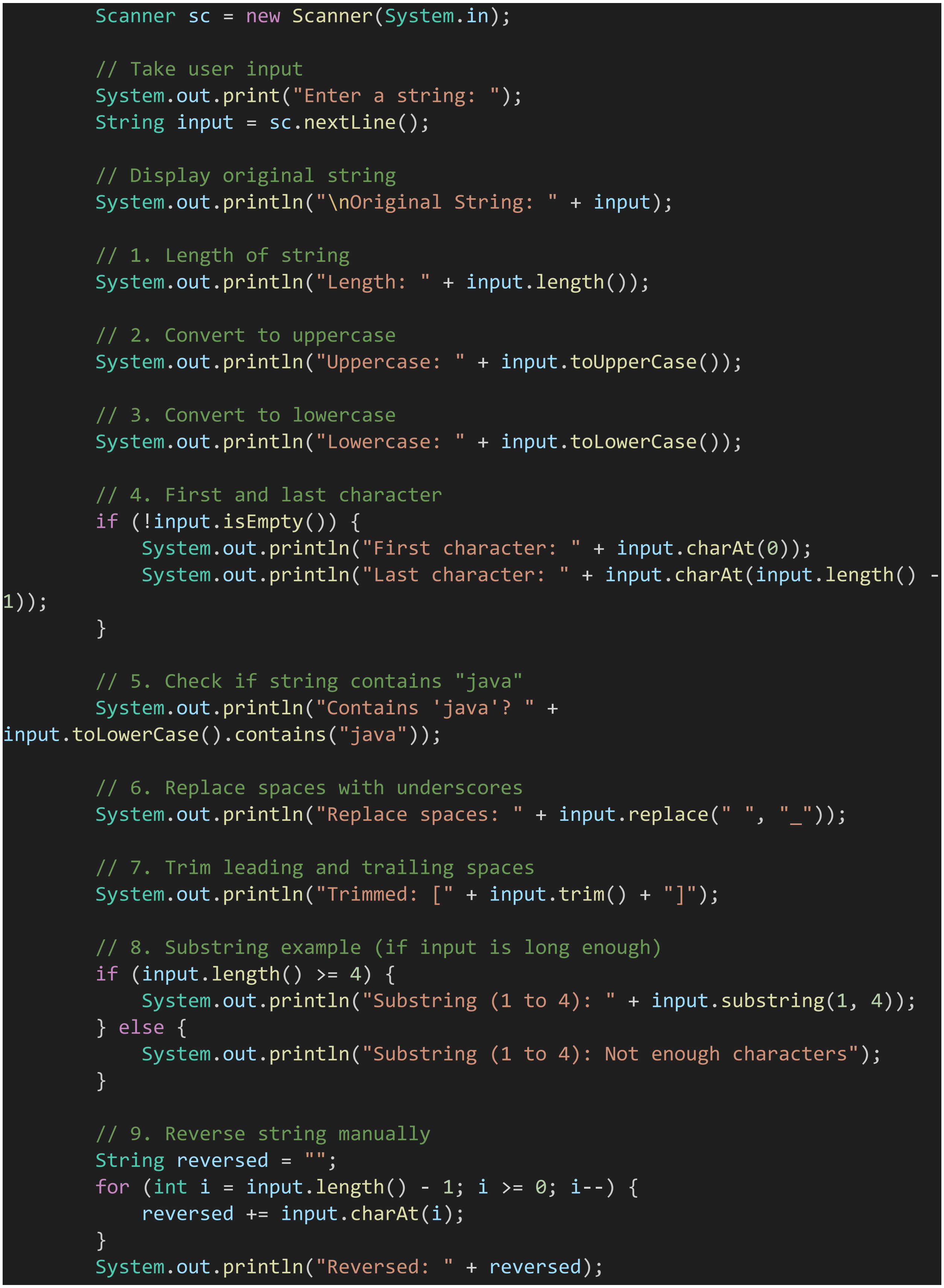


```
String 1 (literal): Hello
String 2 (new keyword): World
String 3 (char array): Java
String 4 (byte array): ABCD
Concatenated String: Hello World
Length of str3: 4
Character at index 2 of str1: l
Substring of str2 (1 to 3): orl
Uppercase str1: HELLO
Lowercase str2: world
str1 equals anotherStr? true
str1 == anotherStr? true
Replace 'a' with 'o' in str3: Jovo
Before trim: [   Java Programming   ]
After trim: [Java Programming]
```

Q2. Create a program that takes user input and processes it using various string methods.

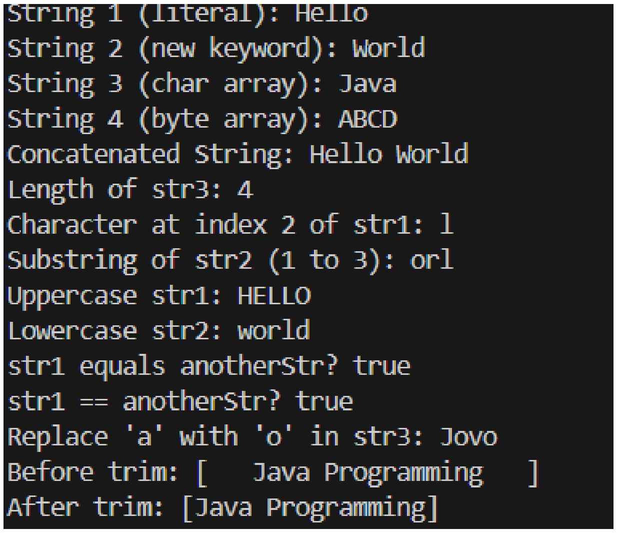
PROGRAM

```
import java.util.Scanner;

public class StringProcessingDemo {
    public static void main(String[] args) {
```



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

        // Take user input
        System.out.print("Enter a string: ");
        String input = sc.nextLine();

        // Display original string
        System.out.println("\nOriginal String: " + input);

        // 1. Length of string
        System.out.println("Length: " + input.length());

        // 2. Convert to uppercase
        System.out.println("Uppercase: " + input.toUpperCase());

        // 3. Convert to lowercase
        System.out.println("Lowercase: " + input.toLowerCase());

        // 4. First and last character
        if (!input.isEmpty()) {
            System.out.println("First character: " + input.charAt(0));
            System.out.println("Last character: " + input.charAt(input.length() -
1));
        }

        // 5. Check if string contains "java"
        System.out.println("Contains 'java'? " +
input.toLowerCase().contains("java"));

        // 6. Replace spaces with underscores
        System.out.println("Replace spaces: " + input.replace(" ", "_"));

        // 7. Trim leading and trailing spaces
        System.out.println("Trimmed: [" + input.trim() + "]");

        // 8. Substring example (if input is long enough)
        if (input.length() >= 4) {
            System.out.println("Substring (1 to 4): " + input.substring(1, 4));
        } else {
            System.out.println("Substring (1 to 4): Not enough characters");
        }

        // 9. Reverse string manually
        String reversed = "";
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed += input.charAt(i);
        }
        System.out.println("Reversed: " + reversed);
        sc.close();
    }

}
```

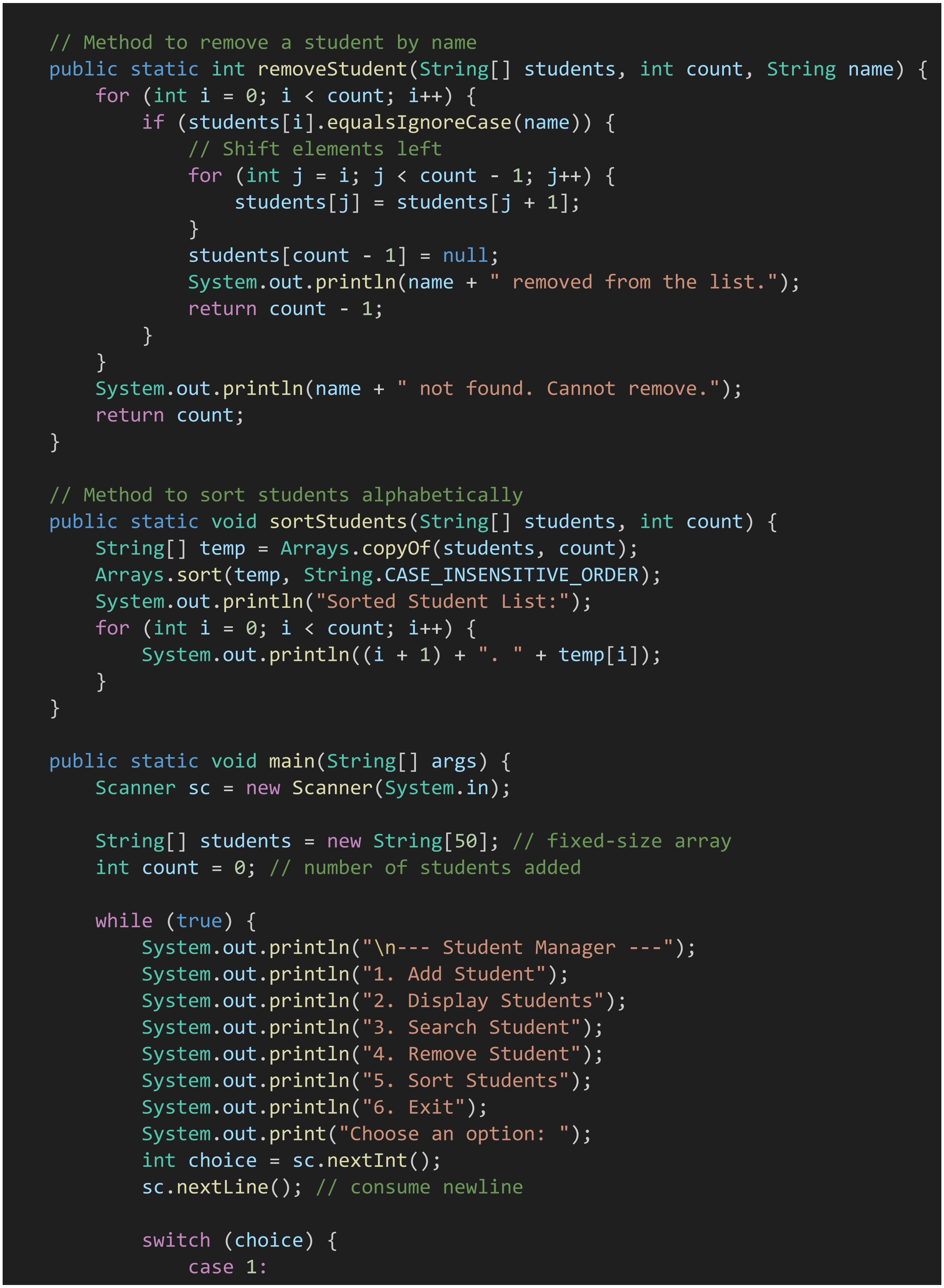
OUTPUT



```
String 1 (literal): Hello
String 2 (new keyword): World
String 3 (char array): Java
String 4 (byte array): ABCD
Concatenated String: Hello World
Length of str3: 4
Character at index 2 of str1: l
Substring of str2 (1 to 3): orl
Uppercase str1: HELLO
Lowercase str2: world
str1 equals anotherStr? true
str1 == anotherStr? true
Replace 'a' with 'o' in str3: Jovo
Before trim: [   Java Programming   ]
After trim: [Java Programming]
```

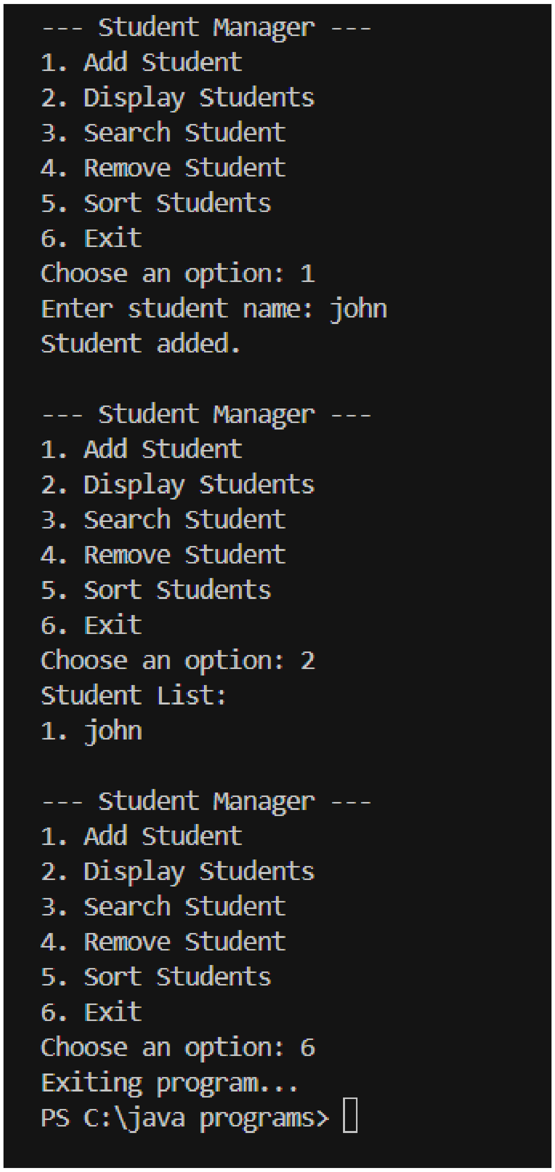
Q3. Create a program that manages a list of student names using string arrays and methods.

```
import java.util.Arrays;
import java.util.Scanner;
public class StudentManager {
    public static void displayStudents(String[] students, int count) {
        if (count == 0) {
            System.out.println("No students in the list.");
            return;
        }
        System.out.println("Student List:");
        for (int i = 0; i < count; i++) {
            System.out.println((i + 1) + ". " + students[i]);
        }
    }

    // Method to search for a student
    public static void searchStudent(String[] students, int count, String name) {
        for (int i = 0; i < count; i++) {
            if (students[i].equalsIgnoreCase(name)) {
                System.out.println(name + " found at position " + (i + 1));
                return;
            }
        }
        System.out.println(name + " not found.");
    }
```



```java
// Method to remove a student by name
public static int removeStudent(String[] students, int count, String name) {
    for (int i = 0; i < count; i++) {
        if (students[i].equalsIgnoreCase(name)) {
            // Shift elements left
            for (int j = i; j < count - 1; j++) {
                students[j] = students[j + 1];
            }
            students[count - 1] = null;
            System.out.println(name + " removed from the list.");
            return count - 1;
        }
    }
    System.out.println(name + " not found. Cannot remove.");
    return count;
}

// Method to sort students alphabetically
public static void sortStudents(String[] students, int count) {
    String[] temp = Arrays.copyOf(students, count);
    Arrays.sort(temp, String.CASE_INSENSITIVE_ORDER);
    System.out.println("Sorted Student List:");
    for (int i = 0; i < count; i++) {
        System.out.println((i + 1) + ". " + temp[i]);
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    String[] students = new String[50]; // fixed-size array
    int count = 0; // number of students added

    while (true) {
        System.out.println("\n--- Student Manager ---");
        System.out.println("1. Add Student");
        System.out.println("2. Display Students");
        System.out.println("3. Search Student");
        System.out.println("4. Remove Student");
        System.out.println("5. Sort Students");
        System.out.println("6. Exit");
        System.out.print("Choose an option: ");
        int choice = sc.nextInt();
        sc.nextLine(); // consume newline

        switch (choice) {
            case 1:
                    if (count < students.length) {
                        System.out.print("Enter student name: ");
                        students[count] = sc.nextLine();
                        count++;
                        System.out.println("Student added.");
                    } else {
                        System.out.println("List is full! Cannot add more 
students.");
                    }
                    break;

                case 2:
                    displayStudents(students, count);
                    break;

                case 3:
                    System.out.print("Enter student name to search: ");
                    String searchName = sc.nextLine();
                    searchStudent(students, count, searchName);
                    break;

                case 4:
                    System.out.print("Enter student name to remove: ");
                    String removeName = sc.nextLine();
                    count = removeStudent(students, count, removeName);
                    break;

                case 5:
                    sortStudents(students, count);
                    break;

                case 6:
                    System.out.println("Exiting program...");
                    sc.close();
                    return;

                default:
                    System.out.println("Invalid choice. Try again.");
            }
        }
    }

}
```

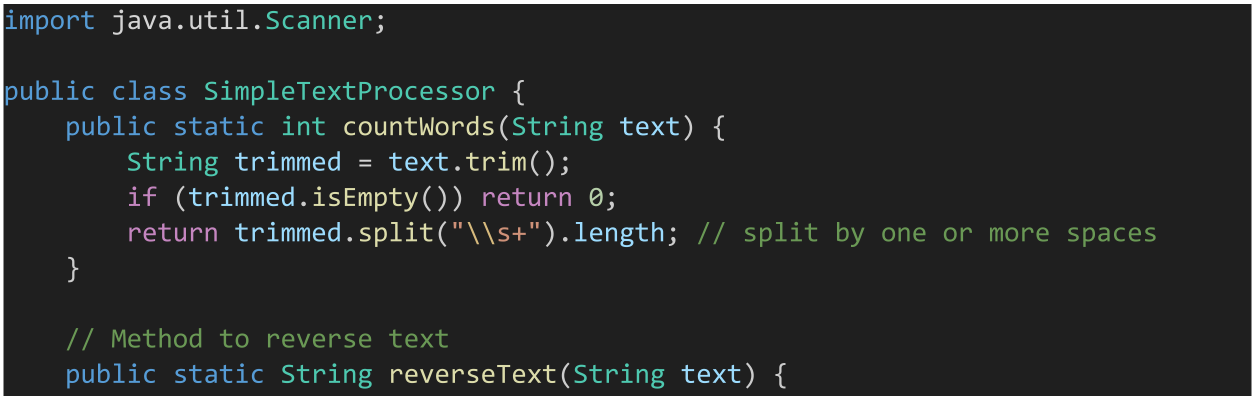
OUTPUT



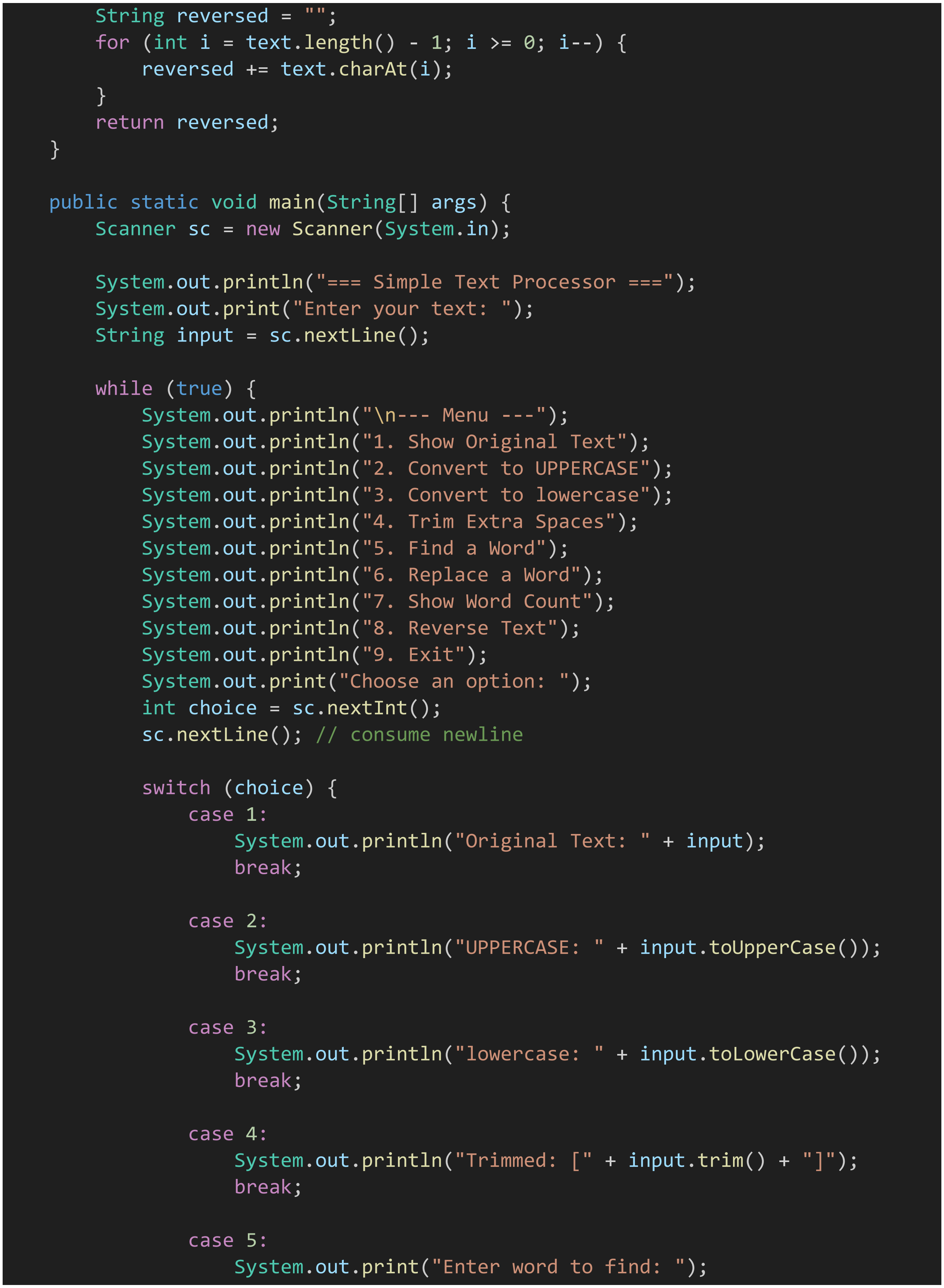
```
--- Student Manager ---
1. Add Student
2. Display Students
3. Search Student
4. Remove Student
5. Sort Students
6. Exit
Choose an option: 1
Enter student name: john
Student added.

--- Student Manager ---
1. Add Student
2. Display Students
3. Search Student
4. Remove Student
5. Sort Students
6. Exit
Choose an option: 2
Student List:
1. john

--- Student Manager ---
1. Add Student
2. Display Students
3. Search Student
4. Remove Student
5. Sort Students
6. Exit
Choose an option: 6
Exiting program...
PS C:\java programs> 
```

Q4 . Create a simple text processor that combines all concepts learned.

PROGRAM

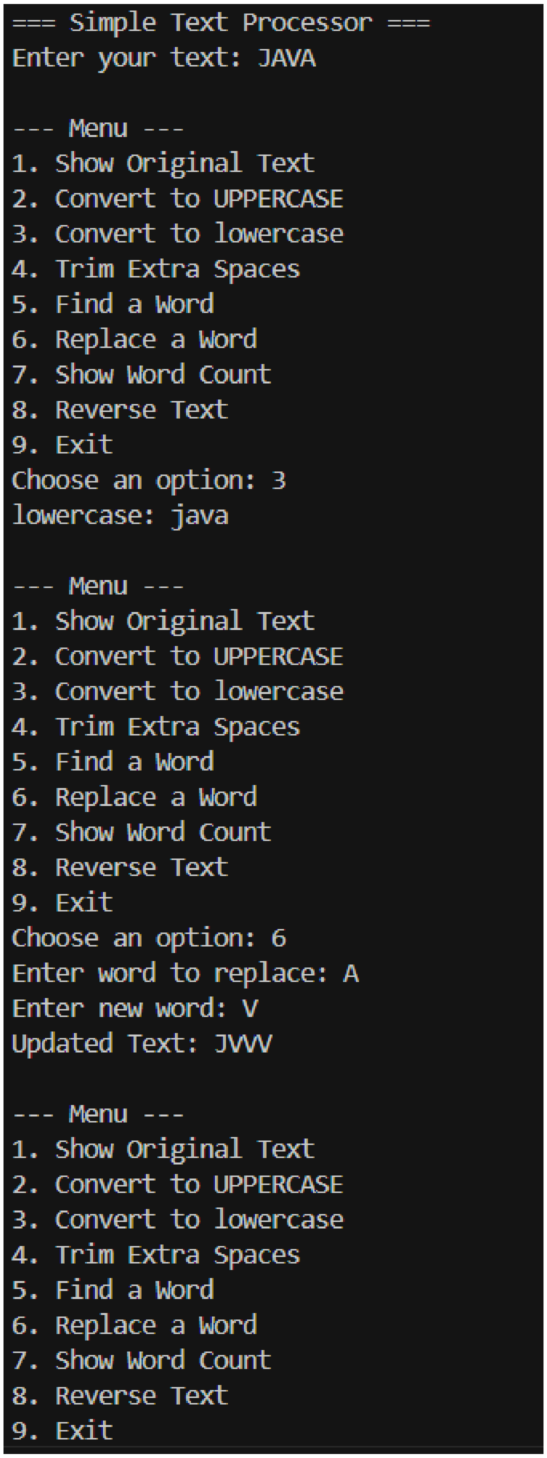


```java
import java.util.Scanner;

public class SimpleTextProcessor {
    public static int countWords(String text) {
        String trimmed = text.trim();
        if (trimmed.isEmpty()) return 0;
        return trimmed.split("\\s+").length; // split by one or more spaces
    }

    // Method to reverse text
    public static String reverseText(String text) {
```



```
        String reversed = "";
        for (int i = text.length() - 1; i >= 0; i--) {
            reversed += text.charAt(i);
        }
        return reversed;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("=== Simple Text Processor ===");
        System.out.print("Enter your text: ");
        String input = sc.nextLine();

        while (true) {
            System.out.println("\n--- Menu ---");
            System.out.println("1. Show Original Text");
            System.out.println("2. Convert to UPPERCASE");
            System.out.println("3. Convert to lowercase");
            System.out.println("4. Trim Extra Spaces");
            System.out.println("5. Find a Word");
            System.out.println("6. Replace a Word");
            System.out.println("7. Show Word Count");
            System.out.println("8. Reverse Text");
            System.out.println("9. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine(); // consume newline

            switch (choice) {
                case 1:
                    System.out.println("Original Text: " + input);
                    break;

                case 2:
                    System.out.println("UPPERCASE: " + input.toUpperCase());
                    break;

                case 3:
                    System.out.println("lowercase: " + input.toLowerCase());
                    break;

                case 4:
                    System.out.println("Trimmed: [" + input.trim() + "]");
                    break;

                case 5:
                    System.out.print("Enter word to find: ");
                    String findWord = sc.nextLine();
                    if (input.toLowerCase().contains(findWord.toLowerCase())) {
                        System.out.println("'" + findWord + "' found in text.");
                    } else {
                        System.out.println("'" + findWord + "' not found.");
                    }
                    break;

                case 6:
                    System.out.print("Enter word to replace: ");
                    String oldWord = sc.nextLine();
                    System.out.print("Enter new word: ");
                    String newWord = sc.nextLine();
                    System.out.println("Updated Text: " + input.replace(oldWord,
newWord));

                    break;

                case 7:
                    System.out.println("Word Count: " + countWords(input));
                    break;

                case 8:
                    System.out.println("Reversed Text: " + reverseText(input));
                    break;

                case 9:
                    System.out.println("Exiting Text Processor...");
                    sc.close();
                    return;

                default:
                    System.out.println("Invalid choice. Try again.");
            }
        }
    }

}
```

OUTPUT



```
=== Simple Text Processor ===
Enter your text: JAVA

--- Menu ---
1. Show Original Text
2. Convert to UPPERCASE
3. Convert to lowercase
4. Trim Extra Spaces
5. Find a Word
6. Replace a Word
7. Show Word Count
8. Reverse Text
9. Exit
Choose an option: 3
lowercase: java

--- Menu ---
1. Show Original Text
2. Convert to UPPERCASE
3. Convert to lowercase
4. Trim Extra Spaces
5. Find a Word
6. Replace a Word
7. Show Word Count
8. Reverse Text
9. Exit
Choose an option: 6
Enter word to replace: A
Enter new word: V
Updated Text: JVVV

--- Menu ---
1. Show Original Text
2. Convert to UPPERCASE
3. Convert to lowercase
4. Trim Extra Spaces
5. Find a Word
6. Replace a Word
7. Show Word Count
8. Reverse Text
9. Exit
```