JAVA PROGRAMMING CAT

SCT221-0498/2023

MORAA RUTH

<u>Question 1</u> - Fibonacci sequence

```
public class Fibonacci{

    public static void main(String[] args) {
        int limit = 4000000;
        int sum = sumFibonacci(limit);
        System.out.println("Sum of even-valued terms: " + sumFibonacci(4000000));
}

    public static int sumFibonacci(int limit) {
        int a = 1, b = 2, sum = 0;

        while (b <= limit) {
            if (b % 2 == 0) {
                sum += b;
            }
            int nextTerm = a + b;
            a = b;
            b = nextTerm;
        }

        return sum;
    }
}
```

<u>Question 2</u> – **Palindrome**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class PalindromeChecker {

    public static void main(String[] args) {
        // Create the GUI frame
        JFrame frame = new JFrame("Palindrome Checker");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 150);

        // Create the text field for user input
        JTextField numberField = new JTextField(10);

        // Create the label to display output
        JLabel resultLabel = new JLabel("");

        // Create the button and add ActionListener to handle click events
        JButton checkButton = new JButton("Output");
        checkButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String input = numberField.getText();
                if (isPalindrome(input)) {
                    resultLabel.setText("Palindrome");
```

```
        } else {
            resultLabel.setText("Not palindrome");
        }
    }
});

// Create a panel with GridBagLayout to arrange components
JPanel panel = new JPanel(new GridBagLayout());
GridBagConstraints gbc = new GridBagConstraints();
gbc.fill = GridBagConstraints.HORIZONTAL;

// First row: Label and TextField
gbc.gridx = 0; // column 0
gbc.gridy = 0; // row 0
panel.add(new JLabel("Enter the number:"), gbc);

gbc.gridx = 1; // column 1
panel.add(numberField, gbc);

// Second row: Button and Result Label
gbc.gridx = 0; // column 0
gbc.gridy = 1; // row 1
panel.add(checkButton, gbc);

gbc.gridx = 1; // column 1
panel.add(resultLabel, gbc);

// Add the panel to the frame
```

```
        frame.add(panel);
        frame.setVisible(true);
    }

    public static boolean isPalindrome(String number) {
        int len = number.length();
        for (int index = 0; index < len / 2; i++) {
            if (number.charAt(i) != number.charAt(len - 1 - index)) {
                return false;
            }
        }
        return true;
    }
}
```

Question 3 – Arrays

```
import java.util.Arrays;
import java.util.Scanner;

public class Array {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] numbers = new int[15];

        // a) Input 15 numbers from the user and print them
        System.out.println("Enter 15 integers:");
        for (int index = 0; index < 15; index++) {
```

```
    numbers[index] = scanner.nextInt();  // Store each entered integer in the array
}

System.out.println("Array elements: " + Arrays.toString(numbers));

// b) Search for a number in the array
System.out.print("Enter a number to search: ");
int searchNumber = scanner.nextInt();  // Read the number to search for
int foundIndex = -1;  // Initialize index to -1 to indicate "not found"
for (int index = 0; index < numbers.length; index++) {
    if (numbers[index] == searchNumber) {
        foundIndex = index;  // Update foundIndex with the current index
        break;  // Exit the loop once the number is found
    }
}

if (foundIndex != -1) {
    System.out.println("Number found at index " + foundIndex);
} else {
    System.out.println("Number not found in this array");
}

// c) Sort the array in ascending order
Arrays.sort(numbers);
System.out.println("Sorted array: " + Arrays.toString(numbers));

// d) Create a new array with elements in reverse order
int[] reversedArray = new int[15];
```

```
        for (int index = 0; index < numbers.length; index++) {
            reversedArray[index] = numbers[numbers.length - 1 - index];
        }

        System.out.println("Reversed array: " + Arrays.toString(reversedArray));

        // e) Get the sum and product of all elements
        int sum = 0;
        long product = 1;
        for (int number : numbers) {
            sum += number;
            product *= number;
        }

        System.out.println("Sum of array elements: " + sum);
        System.out.println("Product of array elements: " + product);
    }
}
```