

Question one: [15 marks]

- a. A prime number is a number that is evenly divisible only by itself and 1. For example, the number 5 is prime because it can be evenly divided only by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 4, and 6. Write a method named isPrime, which takes an integer as an argument and returns true if the argument is a prime number, or false otherwise. Also write main method that displays prime numbers between 1 to 500.

```
Ans; public class PrimeNumbers {
    public static boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        for (int i = 2; i <= 500; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
    }
}
```

- b. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

ANW: public static void generateFibonacci(int n) {

int a = 0, b = 1, c;

for (int i = 1; i <= n; i++) {

System.out.print(a + " ");

c = a + b;

a = b;

```

        b = c;
    }
}

```

- c. By considering the terms in the Fibonacci sequence whose values do not exceed four million, write a Java method to find the sum of all the even- valued terms.

Anw: public static int sumEvenFibonacci(int limit) {

```

    int sum = 0;

    int a = 1, b = 2, c;

    while (a <= limit) {

        if (a % 2 == 0) {

            sum += a;

        }

        c = a + b;

        a = b;

        b = c;

    }

    return sum;

}

```

Question two: [15 marks]

A palindrome number is a number that remain the same when read from behind or front (a number that is equal to reverse of number) for example, 353 is palindrome because reverse of 353 is 353 (you see the number remains the same). But a number like 591 is not palindrome because reverse of 591 is 195 which is not equal to 591. Write Java program to check if a number entered by the user is palindrome or not. You should provide the user with a GUI interface to enter the number and display the results on the same interface.

Anw: import javax.swing.JOptionPane;

```
public class PalindromeChecker {

    public static void main(String[] args) {

        // Ask the user to enter a number

        String input = JOptionPane.showInputDialog(null, "Enter a number:");

        // Convert the input to an integer

        int num = Integer.parseInt(input);

        // Check if the number is a palindrome

        boolean isPalindrome = true;

        String numStr = Integer.toString(num);

        int len = numStr.length();

        for (int i = 0; i < len/2; i++) {

            if (numStr.charAt(i) != numStr.charAt(len-i-1)) {

                isPalindrome = false;

                break;

            }

        }

        // Display the result

        if (isPalindrome) {
```

```

        JOptionPane.showMessageDialog(null, num + " is a palindrome number.");
    } else {
        JOptionPane.showMessageDialog(null, num + " is not a palindrome number.");
    }
}
}
}

```

Question three: [15 marks]

Write a Java program that takes 15 values of type integer as inputs from user, store the values in an array.

- a) Print the values stored in the array on screen.
- b) Ask user to enter a number, check if that number (entered by user) is present in array

or not. If it is present print, "the number found at index (index of the number) " and the text "number not found in this array"

- c) Create another array, copy all the elements from the existing array to the new array but in reverse order. Now print the elements of the new array on the screen

- d) Get the sum and product of all elements of your array. Print product and the sum each on its own line.

Anw: import java.util.Scanner;

```

public class ArrayOperations {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int[] array = new int[15];

        System.out.println("Enter 15 integer values:");
    }
}

```

```
// Taking input values from the user and storing them in the array
```

```
for (int i = 0; i < 15; i++) {  
    array[i] = input.nextInt();  
}
```

```
// Printing the values stored in the array
```

```
System.out.println("Values stored in the array:");  
for (int i = 0; i < 15; i++) {  
    System.out.print(array[i] + " ");  
}
```

```
// Checking if a number entered by the user is present in the array
```

```
System.out.println("\nEnter a number to search:");  
int num = input.nextInt();  
boolean found = false;  
int index = -1;  
for (int i = 0; i < 15; i++) {  
    if (array[i] == num) {  
        found = true;  
        index = i;  
        break;  
    }  
}  
  
if (found) {
```

```
        System.out.println("The number found at index " + index);
    } else {
        System.out.println("Number not found in this array");
    }

    // Creating another array in reverse order and printing its elements

    int[] reverseArray = new int[15];

    for (int i = 0, j = 14; i < 15; i++, j--) {
        reverseArray[j] = array[i];
    }

    System.out.println("Values stored in the reversed array:");

    for (int i = 0; i < 15; i++) {
        System.out.print(reverseArray[i] + " ");
    }

    // Calculating sum and product of all elements of the array

    int sum = 0, product = 1;

    for (int i = 0; i < 15; i++) {
        sum += array[i];
        product *= array[i];
    }

    System.out.println("\nSum of all elements: " + sum);

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

}

Sample output for the program

Enter 15 integer values:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Values stored in the array:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Enter a number to search:

9

The number found at index 8

Values stored in the reversed array:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Sum of all elements: 120

Product of all elements: 1307674368000