Laboratory Work

Subject: Java Technologies
Branch: B.Tech. (CE)
Semester: IV
Batch: II

Student Roll No: CE030
Student Name: PARMAR LAKHMAN

Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University, Nadiad – 387001.
Gujarat, INDIA.

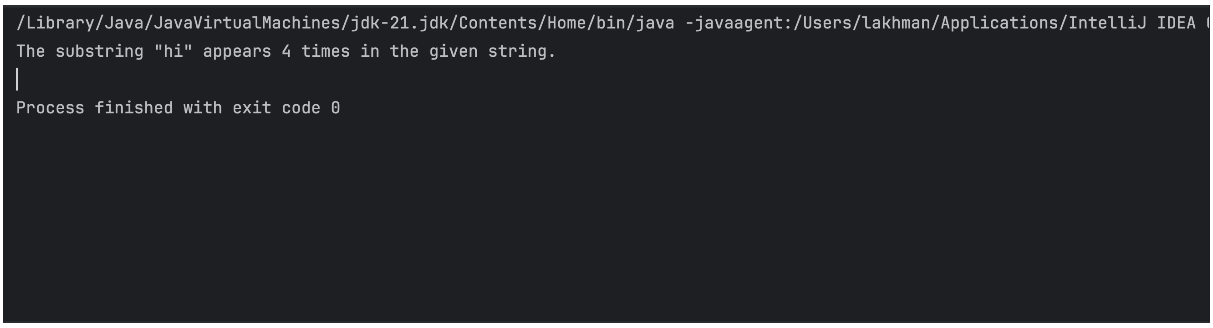
Lab-2

Q.1

1. Write a program that returns the number of times that the string “hi” appears anywhere in
   the given string.

Ans.

```
public class HiCount {

    public static int countHiOccurrences(String inputString) {
        String lowerCaseInput = inputString.toLowerCase();
        int count = 0;

        int index = lowerCaseInput.indexOf("hi");
        while (index != -1) {
            count++;
            index = lowerCaseInput.indexOf("hi", index + 2);
        }

        return count;
    }

    public static void main(String[] args) {
        String inputString = "hi there, hi how are you? Hi, this is a test.";
        int result = countHiOccurrences(inputString);
        System.out.println("The substring \"hi\" appears " + result + " times in the given
string.");
    }
}
```



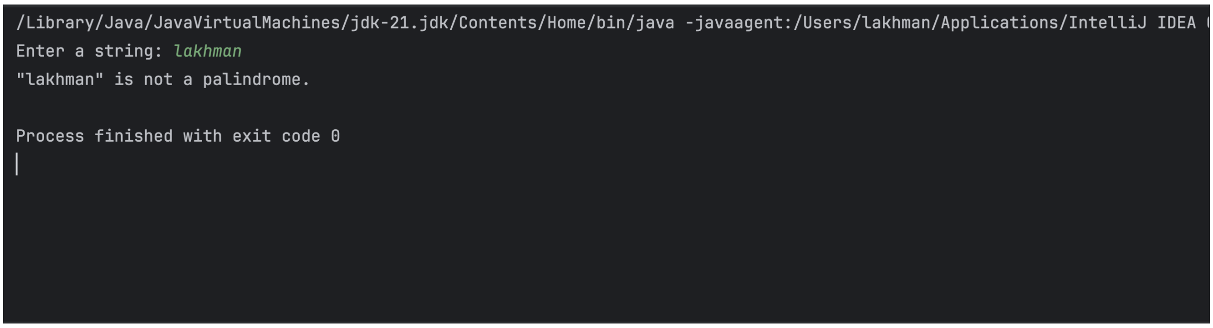
```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA (
The substring "hi" appears 4 times in the given string.

Process finished with exit code 0
```

Q.2

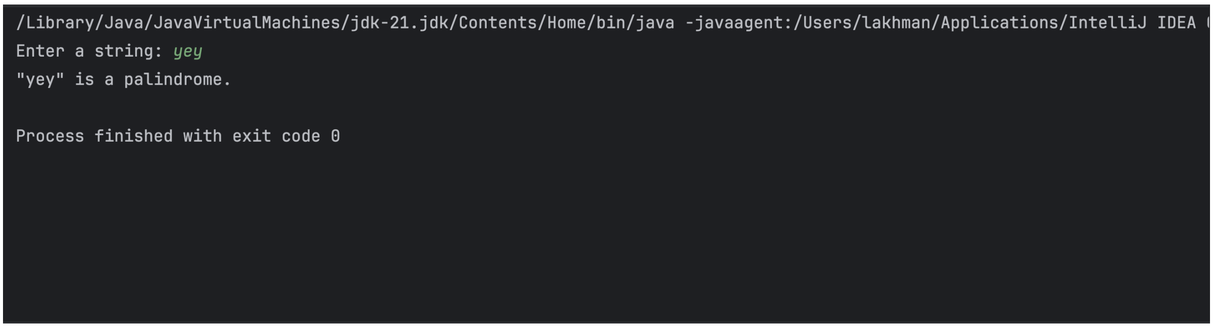
1. Write a program which checks whether the input string is palindrome or not and then display an appropriate message [e.g. “Refer” is a palindrome string].

Ans.

```
import java.util.Scanner;

public class PalindromeChecker {

    public static boolean isPalindrome(String inputString) {
        String cleanedInput = inputString.replaceAll("\\s", "").toLowerCase();
        String reversedString = "";

        for (int i = cleanedInput.length() - 1; i >= 0; i--) {
            reversedString += cleanedInput.charAt(i);
        }

        return cleanedInput.equals(reversedString);
    }

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

        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();
        if (isPalindrome(inputString)) {
            System.out.println("\"" + inputString + "\" is a palindrome string.");
        } else {
            System.out.println("\"" + inputString + "\" is not a palindrome string.");
        }

        scanner.close();
    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter a string: lakhman
"lakhman" is not a palindrome.

Process finished with exit code 0
```



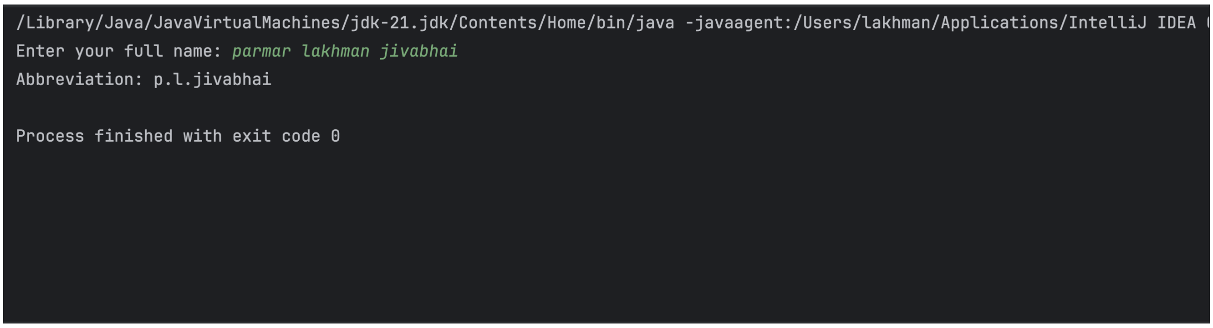
```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter a string: yey
"yey" is a palindrome.

Process finished with exit code 0
```

Q.3

1. Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B.Roser.

Ans.

```
import java.util.Scanner;

public class NameAbbreviation {

    public static String abbreviateName(String fullName) {
        String[] names = fullName.split("\\s");
        StringBuilder abbreviation = new StringBuilder();

        for (int i = 0; i < names.length - 1; i++) {
            abbreviation.append(names[i].charAt(0)).append(".");
        }

        abbreviation.append(names[names.length - 1]);

        return abbreviation.toString();
    }

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

        System.out.print("Enter your full name: ");
        String fullName = scanner.nextLine();

        System.out.println("Abbreviation: " + abbreviateName(fullName));

        scanner.close();
    }
}
```



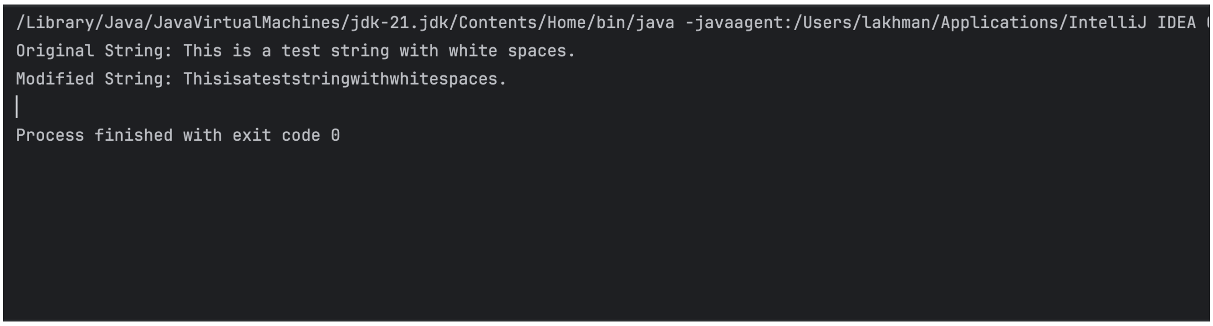
```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA (
Enter your full name: parmar lakhman jivabhai
Abbreviation: p.l.jivabhai

Process finished with exit code 0
```

Q.4

1. Write a method **String removeWhiteSpaces(String str)** method that removes all the white spaces from the string passed to the method and returns the modified string. Test the functionalities using the main() method of the Tester class.

Ans.

```
public class Tester {

    public static String removeWhiteSpaces(String str) {
        return str.replaceAll("\\s", "");
    }

    public static void main(String[] args) {
        String inputString = "This is a test string with white spaces.";

        System.out.println("Original String: " + inputString);
        System.out.println("Modified String: " + removeWhiteSpaces(inputString));
    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA (
Original String: This is a test string with white spaces.
Modified String: Thisisateststringwithwhitespaces.

Process finished with exit code 0
```

Q.5

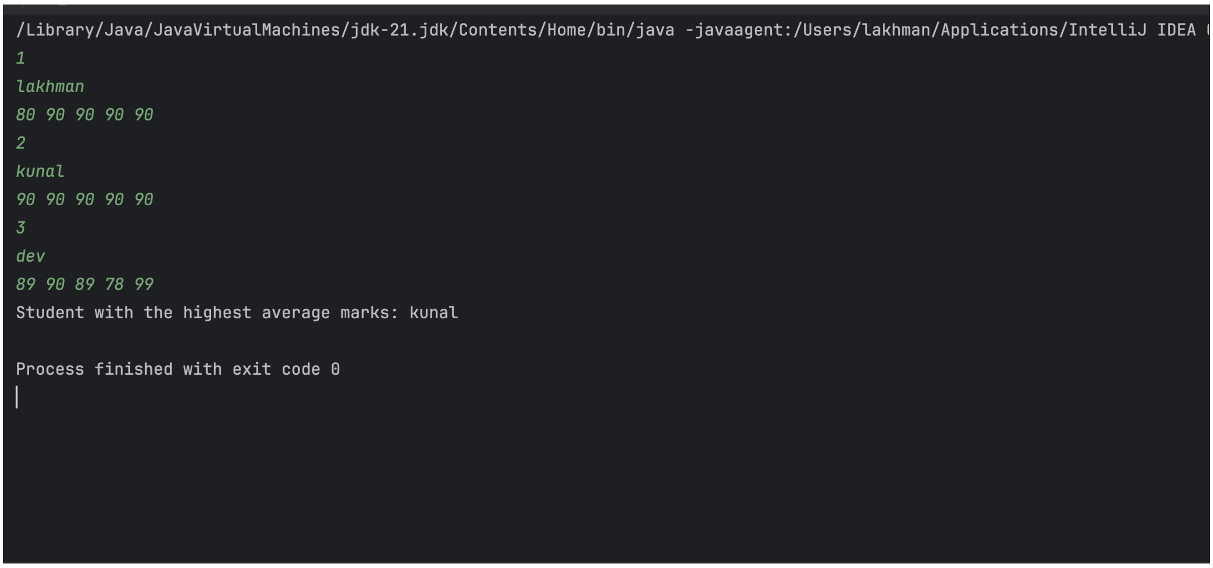
Write a class Student with member variables int roll_no, String name and an array to store marks of 5 subjects. Demonstrate constructor overloading and use this keyword. Write a findAverage() method that returns double value. Write a TestStudent class containing main() method to do the following:

Store the details of one student by creating one object of Student class and display them.

Store the details of 3 students by creating an array of objects of Student class and display the details of the student who has the highest average amongst the three students.

Ans.

```
public class Student {
    private int rollNo;
    private String name;
    private int[] marks;

    public Student() {
        this.rollNo = 0;
        this.name = "";
        this.marks = new int[5];
    }

    public Student(int rollNo, String name) {
        this();
        this.rollNo = rollNo;
        this.name = name;
    }

    public Student(int rollNo, String name, int[] marks) {
        this(rollNo, name);
        this.marks = marks;
    }
    public double findAverage() {
        double sum = 0;
        for (int mark : marks) {
            sum += mark;
        }
        return sum / marks.length;
    }
}

public class TestStudent {
    public static void main(String[] args) {
        Student student1 = new Student();
        System.out.println("Student 1 Average: " + student1.findAverage());

        Student student2 = new Student(101, "John Doe");
        System.out.println("Student 2 Average: " + student2.findAverage());

        int[] marks = {90, 85, 78, 92, 88};
        Student student3 = new Student(102, "Jane Doe", marks);
        System.out.println("Student 3 Average: " + student3.findAverage());
    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
1
lakhman
80 90 90 90 90
2
kunal
90 90 90 90 90
3
dev
89 90 89 78 99
Student with the highest average marks: kunal

Process finished with exit code 0
```

Practice Problems:

Q.6

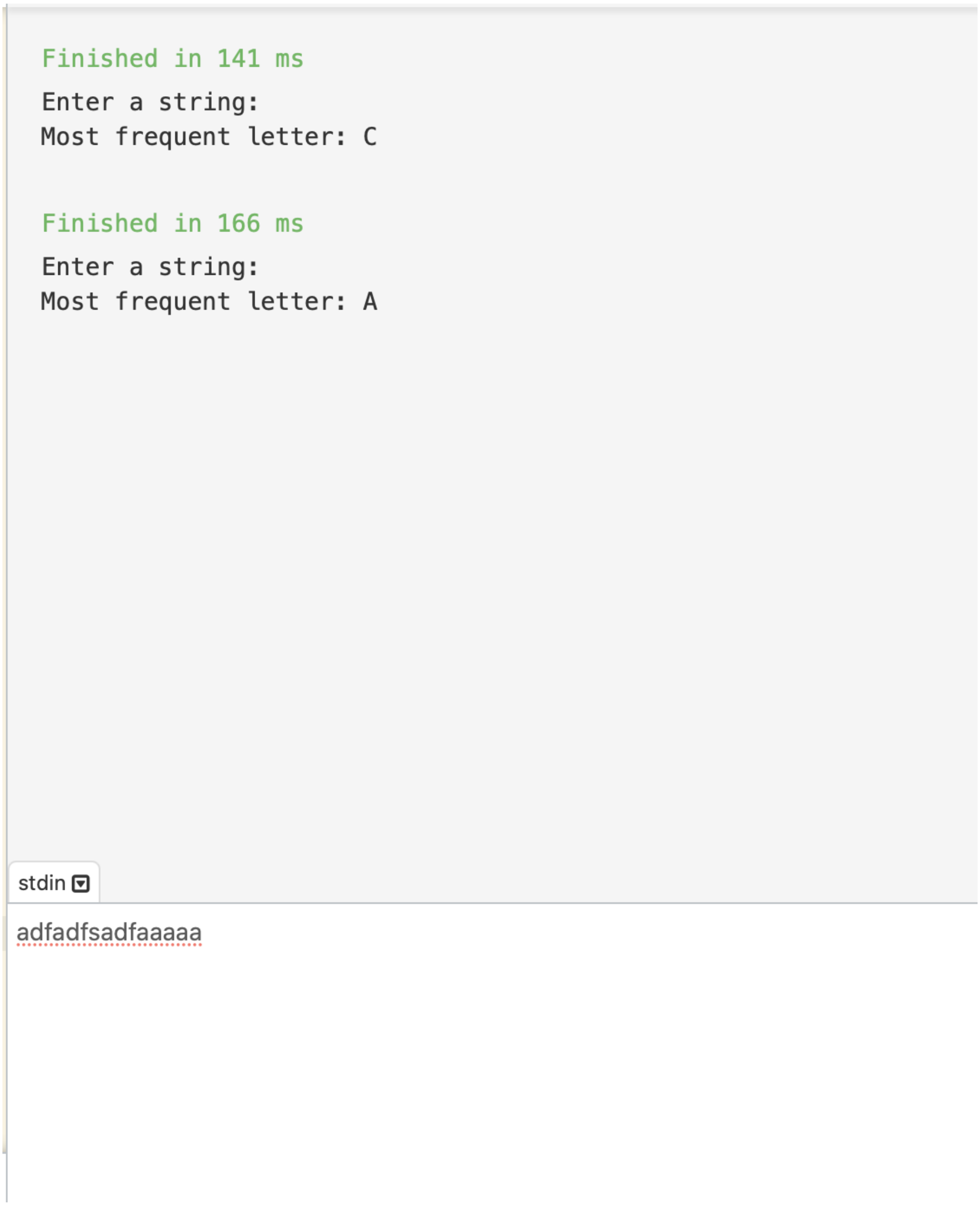
- Write a Java program to find the second most frequent character in a given string.

E.g.

Input	Success
Output	c

Ans.

```
import java.util.*;

class Main {
    static void countFreq(String s, int[] freq) {
        for (int i = 0; i < s.length(); i++) {
            int temp = s.charAt(i);
            if (temp >= 97 && temp <= 122) {
                temp = temp - 32;
            }
            freq[temp - 65] = freq[temp - 65] + 1;
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] freq = new int[26];
        for (int i = 0; i < 26; i++) {
            freq[i] = 0;
        }
        String item = scanner.nextLine();
        countFreq(item, freq);
        int max = 0;
        int index = 0;
        for (int i = 0; i < 26; i++) {
            if (freq[i] > max) {
                max = freq[i];
                index = i;
            }
        }
        index += 65;
        char x = (char) index;
        System.out.print(x);
        System.out.print(" ");
    }
}
```



```
Finished in 141 ms

Enter a string:
Most frequent letter: C


Finished in 166 ms

Enter a string:
Most frequent letter: A
```

```
adfadfsadfaaaaa
```

Q.7

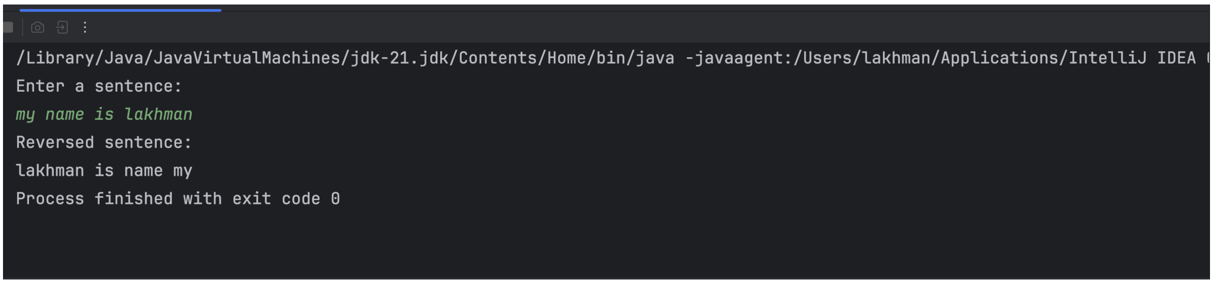
- Write a program that prints the words of a sentence given as input in reverse order.

E.g.

Input	This is a java programming lab
Output	lab programming java a is This

Ans.

```
import java.util.Scanner;

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

        System.out.println("Enter a sentence: ");
        String sentence = scanner.nextLine();

        String[] words = sentence.split("\\s+"); // Split the sentence into words

        System.out.println("Reversed sentence:");
        for (int i = words.length - 1; i >= 0; i--) {
            System.out.print(words[i] + " ");
        }
    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter a sentence:
my name is lakhman
Reversed sentence:
lakhman is name my
Process finished with exit code 0
```

Q.8

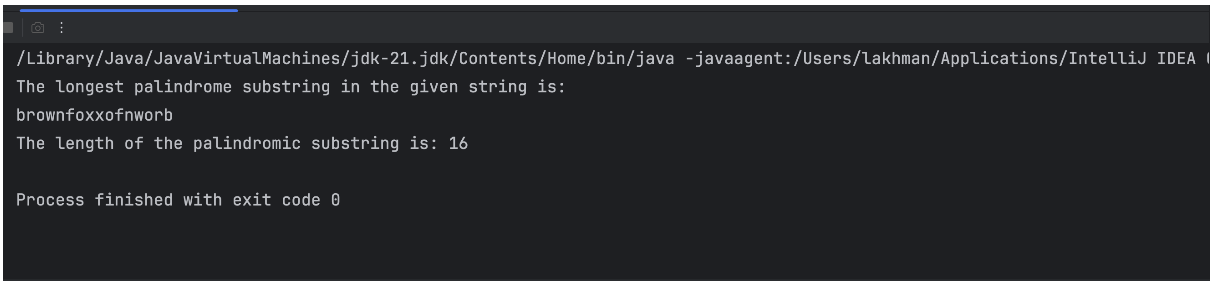
- Write a Java program to find the longest palindrome substring within a string.

E.g.

Input	thequickbrownfoxxofnworbquickthe
Output	The longest palindrome substring in the given string is: brownfoxxofnworb The length of the palindromic substring is: 16

Ans.

```
public class LongestPalindromeSubstring {
public static void main(String[] args) {
String input = "thequickbrownfoxxofnworbquickthe";

String longestPalindrome = findLongestPalindrome(input);

System.out.println("The longest palindrome substring in the given
string is:");
System.out.println(longestPalindrome);
System.out.println("The length of the palindromic substring is: " +

longestPalindrome.length());
}

private static String findLongestPalindrome(String input) {
int maxLength = 0;
String longestPalindrome = "";

for (int i = 0; i < input.length(); i++) {
for (int j = i + 1; j <= input.length(); j++) {
String sub = input.substring(i, j);
if (isPalindrome(sub) && sub.length() > maxLength) {
maxLength = sub.length();
longestPalindrome = sub;
}
}
}

return longestPalindrome;
}

private static boolean isPalindrome(String str) {
int left = 0;
int right = str.length() - 1;

while (left < right) {
if (str.charAt(left) != str.charAt(right)) {
return false;
}
left++;
right--;
}
return true;
}

}
```



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
The longest palindrome substring in the given string is:
brownfoxxofnworb
The length of the palindromic substring is: 16

Process finished with exit code 0
```