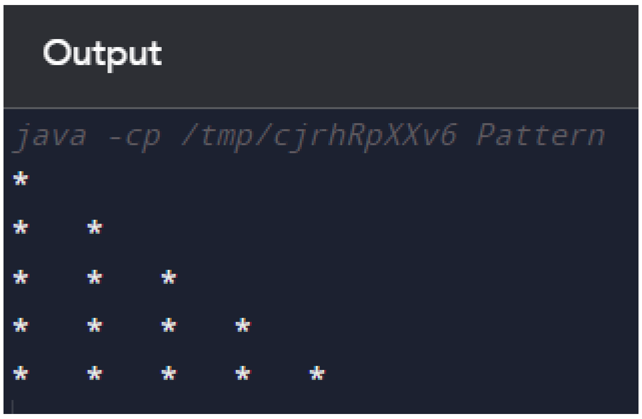
Name - Himanshu Tiwari
Roll number - 10
Class - MSc Computational Science and Application
Semester - 1

JAVA PROGRAMMING ASSIGNMENT

Q1. Write a program in Java to print the given pattern.

A1.

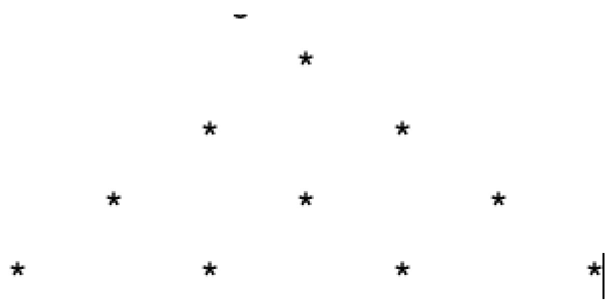
```
public class Pattern {
    public static void main(String[] args) {
        int n = 5;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*\t");
            }
            System.out.println();
        }
    }
}
```


Output

```
java -cp /tmp/cjrhRpXXv6 Pattern
*
*   *
*   *   *
*   *   *   *
*   *   *   *   *
```

2.

```
public class PrintPattern {
    public static void main(String[] args) {
        int rows = 4;
        System.out.println();
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows - i; j++) {
                System.out.print(" ");
            }
            for (int k = 1; k <= 2 * i - 1; k++) {
                System.out.print("*");
                System.out.print(" ");
            }
            System.out.println();
        }
    }
}
```



```
         *
      *     *
   *     *     *
*     *     *     *
```

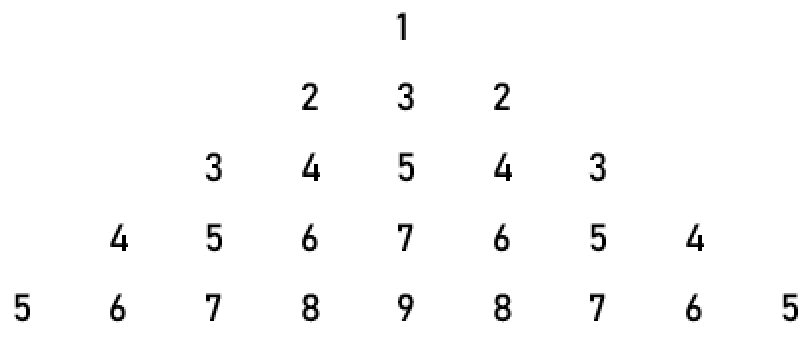
3.

```
public class PrintPattern {
    public static void main(String[] args) {
        int size = 5; // size of the pattern
        int mid = size / 2; // midpoint of the pattern

        // loop through each row of the pattern
        for (int i = 0; i < size; i++) {

            // loop through each column of the pattern
            for (int j = 0; j < size; j++) {

                // check if we need to print a star
                if (i == j || i == mid || j == mid || i + j == size - 1) {
                    System.out.print("*\t");
                } else {
                    System.out.print("\t");
                }
            }

            // move to the next line
            System.out.println();
        }
    }
}
```

```
                *

            *       *

        *               *

    *                       *

*   *   *   *   *   *   *   *   *
```

4.

```
public class Pattern {

    public static void main(String[] args) {

        int n = 5; // number of rows
        int count = 1; // starting number
        int spaces = n - 1; // number of spaces before each row

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

            // print spaces before each row
            for (int j = 1; j <= spaces; j++) {
                System.out.print("\t");
            }

            // print numbers for the current row
            int k = count;
            for (int j = 1; j <= i; j++) {
                System.out.print(k + "\t");
                k++;
            }
            for (int j = 1; j <= i-1; j++) {
                k--;
                System.out.print(k + "\t");
            }

            System.out.println();
            spaces--;
            count++;
        }
    }
}
```

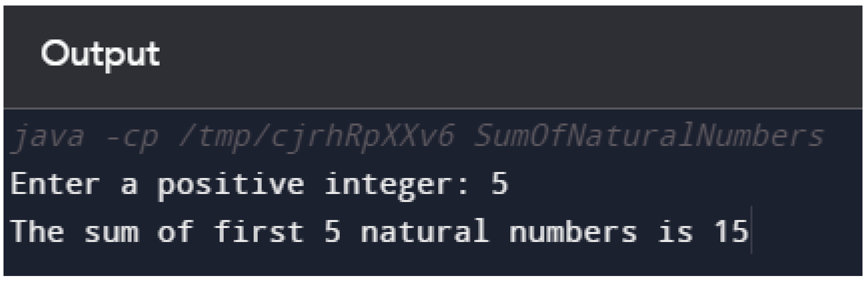


```
                1
            2   3   2
        3   4   5   4   3
    4   5   6   7   6   5   4
5   6   7   8   9   8   7   6   5
```

Q2. Program in Java to find the sum of N Natural Numbers.

A2.

```
import java.util.Scanner;

public class SumOfNaturalNumbers {

    public static void main(String[] args) {
        int n, sum = 0;

        Scanner input = new Scanner(System.in);
        System.out.print("Enter a positive integer: ");
        n = input.nextInt();

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

        System.out.println("The sum of first " + n + " natural numbers is " + sum);
    }
}
```



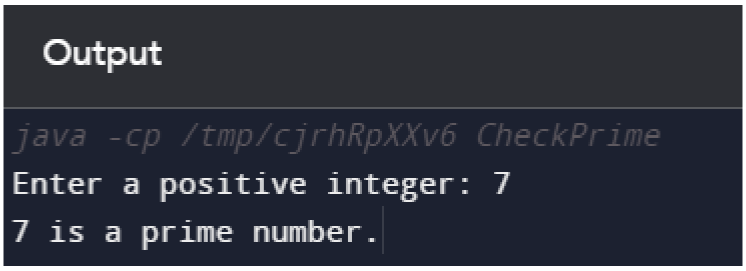
Output

```
java -cp /tmp/cjrhRpXXv6 SumOfNaturalNumbers
Enter a positive integer: 5
The sum of first 5 natural numbers is 15
```

Q3. Program in Java to Check Prime Number.

A3.

```java
import java.util.Scanner;

public class CheckPrime {

    public static void main(String[] args) {
        int n;
        boolean isPrime = true;

        Scanner input = new Scanner(System.in);
        System.out.print("Enter a positive integer: ");
        n = input.nextInt();

        if (n <= 1) {
            isPrime = false;
        }
        else {
            for (int i = 2; i <= Math.sqrt(n); i++) {
                if (n % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

        if (isPrime) {
            System.out.println(n + " is a prime number.");
        }
        else {
            System.out.println(n + " is not a prime number.");
        }
    }
}
```

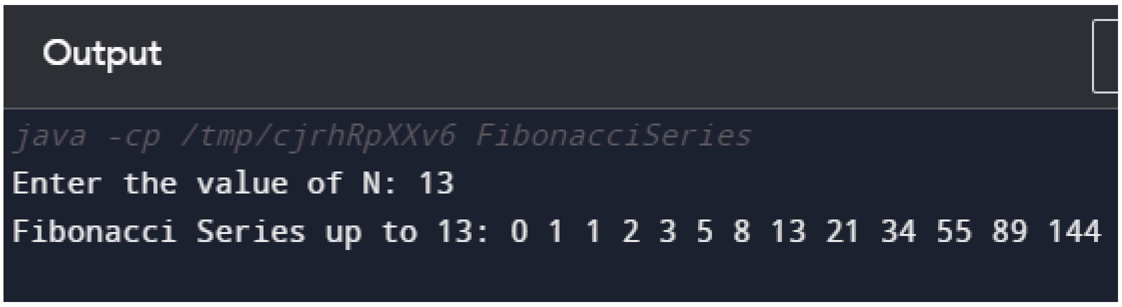

Output

```
java -cp /tmp/cjrhRpXXv6 CheckPrime
Enter a positive integer: 7
7 is a prime number.
```

Q4. Program in Java to print Fibonacci Series up to N.

A4.

```java
import java.util.Scanner;

public class FibonacciSeries {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the value of N: ");
        int n = scanner.nextInt();
        scanner.close();
        int n1 = 0, n2 = 1, sum;
        System.out.print("Fibonacci Series up to " + n + ": ");
        for (int i = 1; i <= n; i++) {
            System.out.print(n1 + " ");
            sum = n1 + n2;
            n1 = n2;
            n2 = sum;
        }
    }
}
```

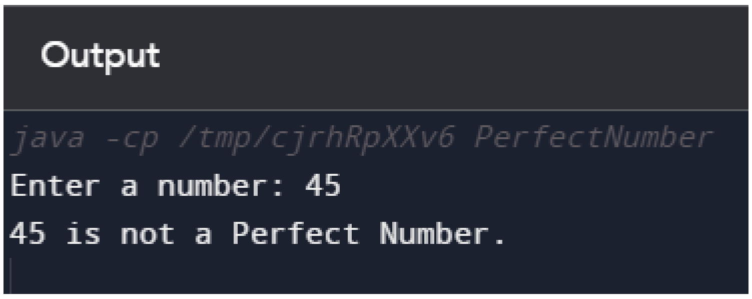


```
java -cp /tmp/cjrhRpXXv6 FibonacciSeries
Enter the value of N: 13
Fibonacci Series up to 13: 0 1 1 2 3 5 8 13 21 34 55 89 144
```

Q5. Program in Java to Check Perfect Numbers.

A5.

```java
import java.util.Scanner;

public class PerfectNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        scanner.close();
        int sum = 0;
        for (int i = 1; i < num; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }
        if (sum == num) {
            System.out.println(num + " is a Perfect Number.");
        } else {
            System.out.println(num + " is not a Perfect Number.");
        }
    }
}
```

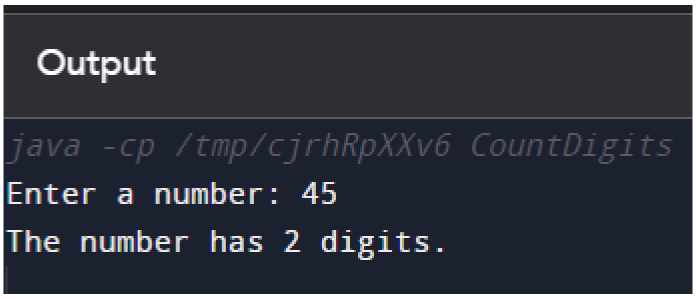


```
java -cp /tmp/cjrhRpXXv6 PerfectNumber
Enter a number: 45
45 is not a Perfect Number.
```

Q6. Program in Java to Count the digits of a number given by the user.

A6.

```
import java.util.Scanner;

public class CountDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        scanner.close();
        int count = 0;
        while (num > 0) {
            num /= 10;
            count++;
        }
        System.out.println("The number has " + count + " digits.");
    }
}
```

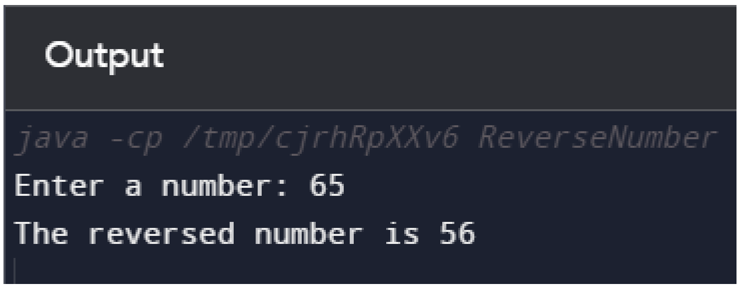


```
Output
java -cp /tmp/cjrhRpXXv6 CountDigits
Enter a number: 45
The number has 2 digits.
```

Q7. Program in Java to Reverse a number given input by the user.

A7.

```
import java.util.Scanner;

public class ReverseNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        scanner.close();
        int reversedNum = 0;
        while (num != 0) {
            int digit = num % 10;
            reversedNum = reversedNum * 10 + digit;
            num /= 10;
        }
        System.out.println("The reversed number is " + reversedNum);
    }
}
```



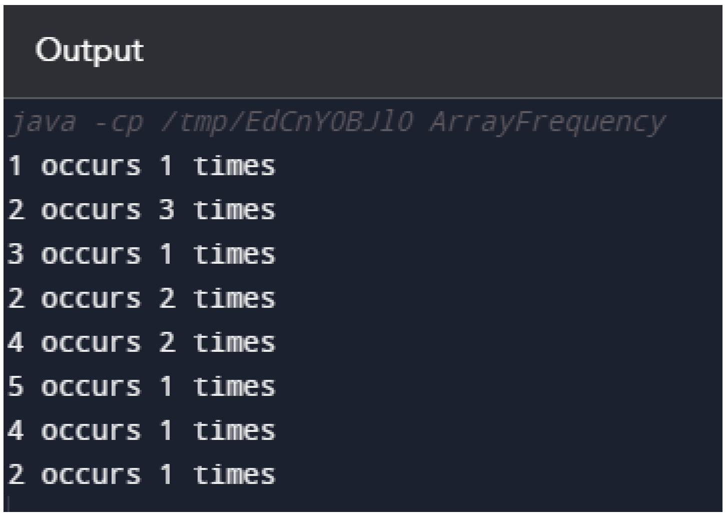
```
Output
java -cp /tmp/cjrhRpXXv6 ReverseNumber
Enter a number: 65
The reversed number is 56
```

Assignment 8

Q1. To print the elements of an array and also to find the frequency of each element in the array.

A1.

```
public class ArrayFrequency {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 2, 4, 5, 4, 2}; // example array
        int[] freq = new int[arr.length]; // array to store frequencies

        // loop through each element of the array
        for (int i = 0; i < arr.length; i++) {
            int count = 1; // initialize frequency count to 1

            // loop through the remaining elements of the array
            for (int j = i + 1; j < arr.length; j++) {
                // if the current element is the same as another element
                // in the remaining array, increment the frequency count
                if (arr[i] == arr[j]) {
                    count++;
                }
            }

            // store the frequency count for the current element
            freq[i] = count;

            // print the current element and its frequency
            System.out.println(arr[i] + " occurs " + freq[i] + " times");
        }
    }
}
```



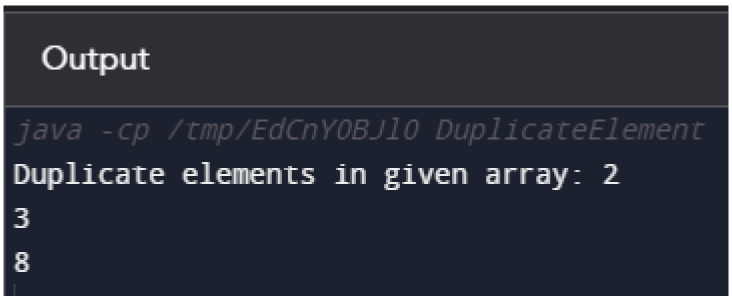
Output

```
java -cp /tmp/EdCnYOBJl0 ArrayFrequency
1 occurs 1 times
2 occurs 3 times
3 occurs 1 times
2 occurs 2 times
4 occurs 2 times
5 occurs 1 times
4 occurs 1 times
2 occurs 1 times
```

Q2. To print the duplicate elements of an array.

A2.

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

        //Initialize array
        int [] arr = new int [] {1, 2, 3, 4, 2, 7, 8, 8, 3};

        System.out.println("Duplicate elements in given array: ");
        //Searches for duplicate element
        for(int i = 0; i < arr.length; i++) {
            for(int j = i + 1; j < arr.length; j++) {
                if(arr[i] == arr[j])
                    System.out.println(arr[j]);
            }
        }
    }
}
```



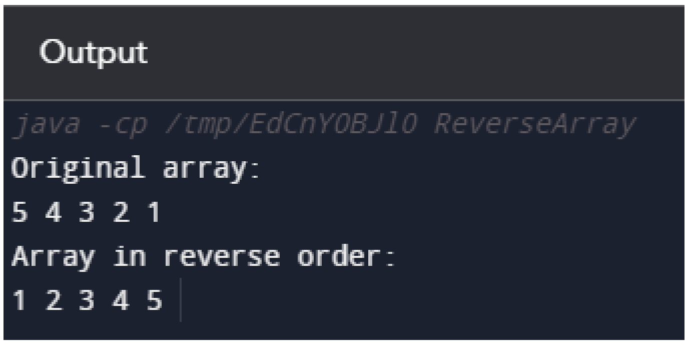
Output

```
java -cp /tmp/EdCnYOBJl0 DuplicateElement
Duplicate elements in given array: 2
3
8
```

Q3. To print the elements of an array in reverse order.

A3.

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

        //Initialize array
        int [] arr = new int [] {5, 4, 3, 2, 1};

        System.out.println("Original array: ");
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }

        System.out.println();

        System.out.println("Array in reverse order: ");
        //Loop through the array in reverse order
        for (int i = arr.length-1; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

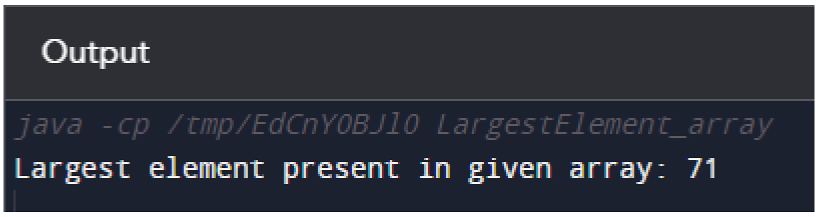


```
Output
java -cp /tmp/EdCnYOBJ10 ReverseArray
Original array:
5 4 3 2 1
Array in reverse order:
1 2 3 4 5
```

Q4. To print the largest element in the array.

A4.

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

        //Initialize array
        int [] arr = new int [] {15, 12, 71, 71, 26};
        //Initialize max with first element of array.
        int max = arr[0];
        //Loop through the array
        for (int i = 0; i < arr.length; i++) {
            //Compare elements of array with max
           if(arr[i] > max)
               max = arr[i];
        }
        System.out.println("Largest element present in given array: " + max);
    }
}
```

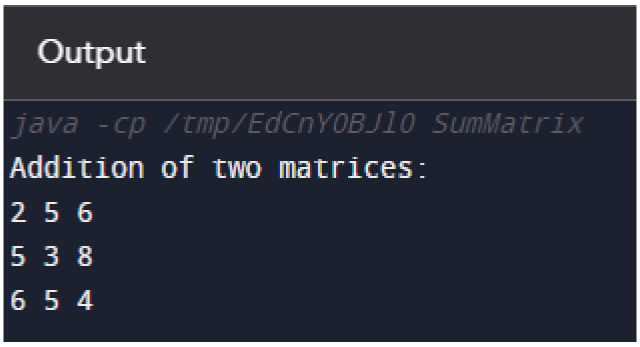


```
Output
java -cp /tmp/EdCnYOBJ10 LargestElement_array
Largest element present in given array: 71
```

Q5. To implement matrix addition.

A5.

```
public class SumMatrix
{
   public static void main(String[] args) {
      int rows, cols;

      //Initialize matrix a
        int a[][] = {
                    {1, 4, 1},
                    {3, 1, 2},
                    {5, 2, 3}
                  };

        //Initialize matrix b
        int b[][] = {
                    {1, 1, 5},
                    {2, 2, 6},
                    {1, 3, 1}
                 };

        //Calculates number of rows and columns present in given matrix
        rows = a.length;
      cols = a[0].length;

        //Array sum will hold the result
      int sum[][] = new int[rows][cols];

      //Performs addition of matrices a and b. Store the result in matrix sum
      for(int i = 0; i < rows; i++){
         for(int j = 0; j < cols; j++){
            sum[i][j] = a[i][j] + b[i][j];
         }
      }

      System.out.println("Addition of two matrices: ");
      System.out.println();
      for(int i = 0; i < rows; i++){
         for(int j = 0; j < cols; j++){
           System.out.print(sum[i][j] + " ");
         }
         System.out.println();
      }
   }
}
```



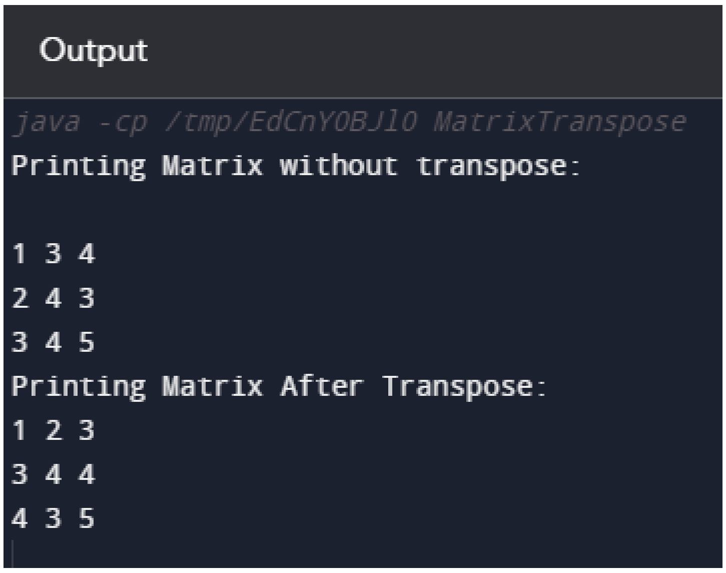
```
Output
java -cp /tmp/EdCnYOBJ1O SumMatrix
Addition of two matrices:
2 5 6
5 3 8
6 5 4
```

Q6. To find the transpose of a matrix.

A6.

```
public class MatrixTranspose{
  public static void main(String args[]){
  //creating a matrix
    int original[][]={{1,3,4},{2,4,3},{3,4,5}};

    //creating another matrix to store transpose of a matrix
    int transpose[][]=new int[3][3];  //3 rows and 3 columns

    //Code to transpose a matrix
    for(int i=0;i<3;i++){
      for(int j=0;j<3;j++){
        transpose[i][j]=original[j][i];
      }
    }

    System.out.println("Printing Matrix without transpose:");
    System.out.println();
    for(int i=0;i<3;i++){
      for(int j=0;j<3;j++){
        System.out.print(original[i][j]+" ");
      }
      System.out.println();//new line
    }
    System.out.println("Printing Matrix After Transpose:");
    for(int i=0;i<3;i++){
      for(int j=0;j<3;j++){
        System.out.print(transpose[i][j]+" ");
      }
      System.out.println();//new line
```

```
        }
    }
}
```

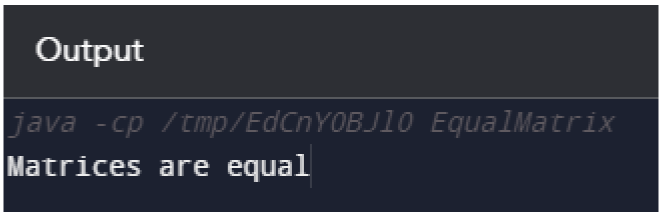

Output

```
java -cp /tmp/EdCnYOBJl0 MatrixTranspose
Printing Matrix without transpose:

1 3 4
2 4 3
3 4 5
Printing Matrix After Transpose:
1 2 3
3 4 4
4 3 5
```

Q7. To check whether the two matrices are equal or not.

A7.

```
public class EqualMatrix
{
    public static void main(String[] args) {
        int row1, col1, row2, col2;
        boolean flag = true;

        //Initialize matrix a
        int a[][] = {
                    {1, 2, 3},
                    {1, 4, 3},
                    {2, 5, 7}
                };

         //Initialize matrix b
        int b[][] = {
                    {1, 2, 3},
                    {1, 4, 3},
                    {2, 5, 7}
          };

        //Calculates the number of rows and columns present in the first matrix
        row1 = a.length;
        col1 = a[0].length;

        //Calculates the number of rows and columns present in the second matrix

        row2 = b.length;
        col2 = b[0].length;

        //Checks if dimensions of both the matrices are equal
        if(row1 != row2 || col1 != col2){
            System.out.println("Matrices are not equal");
        }
        else {
            for(int i = 0; i < row1; i++){
                for(int j = 0; j < col1; j++){
                  if(a[i][j] != b[i][j]){
                      flag = false;
                      break;
                  }
                }
            }

            if(flag)
                System.out.println("Matrices are equal");
            else
                System.out.println("Matrices are not equal");
        }
    }
}
```

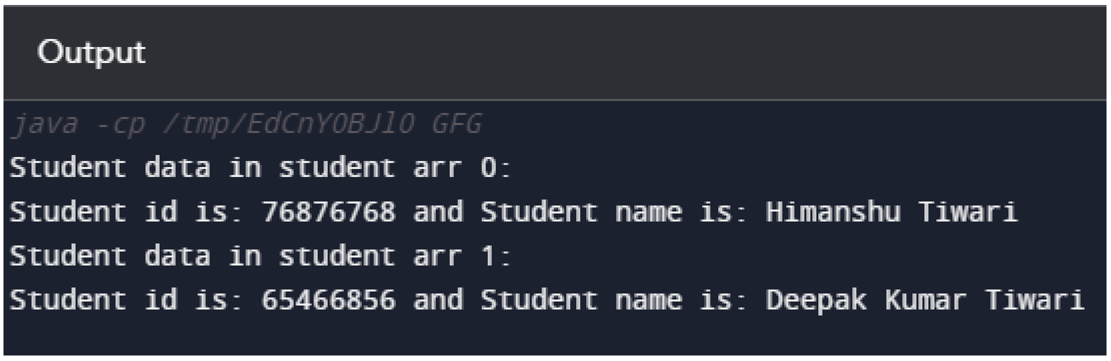


```
Output
java -cp /tmp/EdCnYOBJlO EqualMatrix
Matrices are equal
```

Q8. To create an array of objects.

A8.

```
// Java program to demonstrate initializing
// an array of objects using constructor

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

                // Declaring an array of student
                Student[] arr;

                // Allocating memory for 2 objects
                // of type student
                arr = new Student[2];

                // Initializing the first element
                // of the array
                arr[0] = new Student(76876768, "Himanshu Tiwari");

                // Initializing the second element
                // of the array
                arr[1] = new Student(65466856, "Deepak Kumar Tiwari");

                // Displaying the student data
                System.out.println(
                        "Student data in student arr 0: ");
                arr[0].display();

                System.out.println(
                        "Student data in student arr 1: ");
                arr[1].display();
        }
}

// Creating a student class with
// id and name as a attributes
class Student {

        public int id;
        public String name;

        // Student class constructor
        Student(int id, String name)
        {
                this.id = id;
                this.name = name;
        }

        // display() method to display
        // the student data
        public void display()
        {
                System.out.println("Student id is: " + id + " "
                                                + "and Student name is: "
                                                + name);
                System.out.println();
        }
}
```

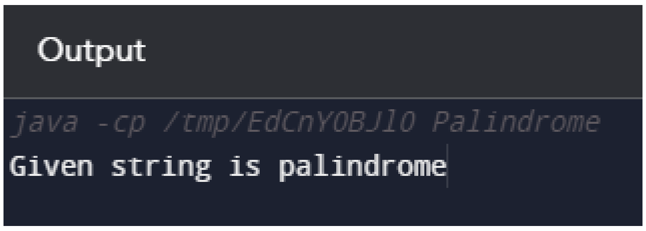


```
Output
java -cp /tmp/EdCnYOBJ10 GFG
Student data in student arr 0:
Student id is: 76876768 and Student name is: Himanshu Tiwari
Student data in student arr 1:
Student id is: 65466856 and Student name is: Deepak Kumar Tiwari
```

Q9. To check whether a string is palindrome or not.

A9.

```
public class Palindrome
{
  public static void main(String[] args) {
    String string = "Naman";
    boolean flag = true;

    //Converts the given string into lowercase
    string = string.toLowerCase();

    //Iterate the string forward and backward, compare one character at a time
    //till middle of the string is reached
    for(int i = 0; i < string.length()/2; i++){
      if(string.charAt(i) != string.charAt(string.length()-i-1)){
        flag = false;
        break;
      }
    }
    if(flag)
      System.out.println("Given string is palindrome");
    else
      System.out.println("Given string is not a palindrome");
    }
}
```

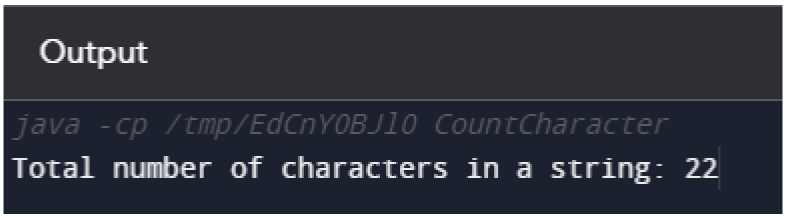


```
Output
java -cp /tmp/EdCnYOBJl0 Palindrome
Given string is palindrome
```

Q10. To count the number of characters in the string.

A10.

```
public class CountCharacter
{
    public static void main(String[] args) {
        String string = "My name is Himanshu Tiwari";
        int count = 0;

        //Counts each character except space
        for(int i = 0; i < string.length(); i++) {
            if(string.charAt(i) != ' ')
                count++;
        }

        //Displays the total number of characters present in the given string
        System.out.println("Total number of characters in a string: " + count);
    }
}
```

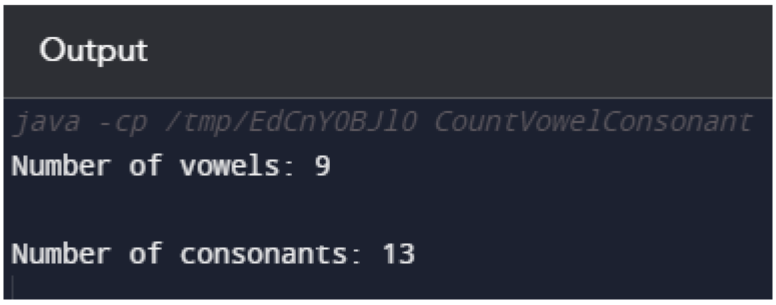


```
Output
java -cp /tmp/EdCnYOBJl0 CountCharacter
Total number of characters in a string: 22
```

Q11. To display the vowels and consonants in a given string.

A11.

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

        //Counter variable to store the count of vowels and consonant
        int vCount = 0, cCount = 0;

        //Declare a string
        String str = "My name is Himanshu Tiwari";

        //Converting entire string to lower case to reduce the comparisons
        str = str.toLowerCase();

        for(int i = 0; i < str.length(); i++) {
            //Checks whether a character is a vowel
            if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o'
|| str.charAt(i) == 'u') {
                //Increments the vowel counter
                vCount++;
            }
            //Checks whether a character is a consonant
            else if(str.charAt(i) >= 'a' && str.charAt(i)<='z') {
                //Increments the consonant counter
                cCount++;
            }
        }
        System.out.println("Number of vowels: " + vCount);
        System.out.println();
        System.out.println("Number of consonants: " + cCount);
    }
}
```

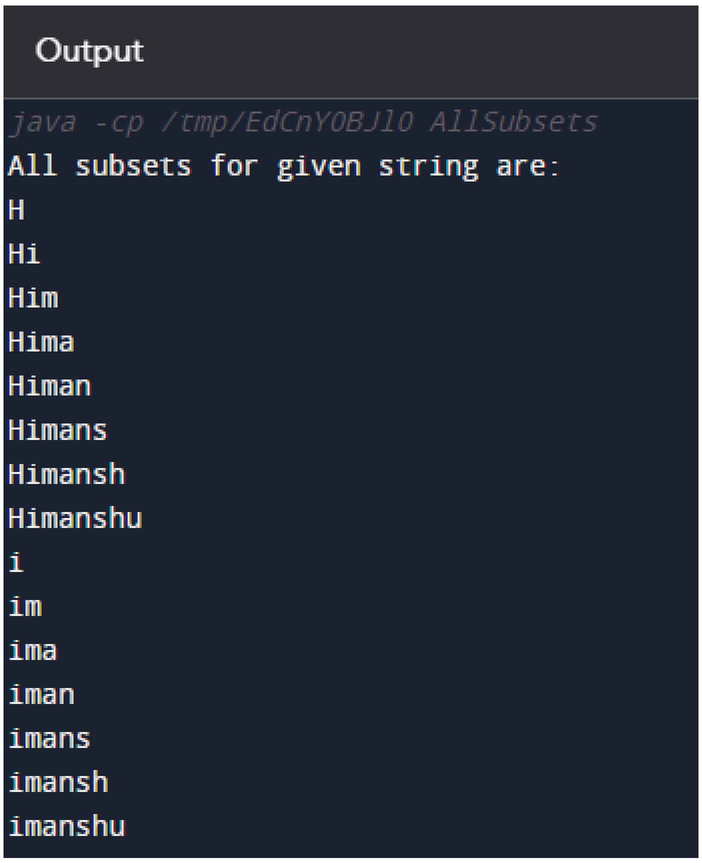

Output

```
java -cp /tmp/EdCnYOBJl0 CountVowelConsonant
Number of vowels: 9

Number of consonants: 13
```

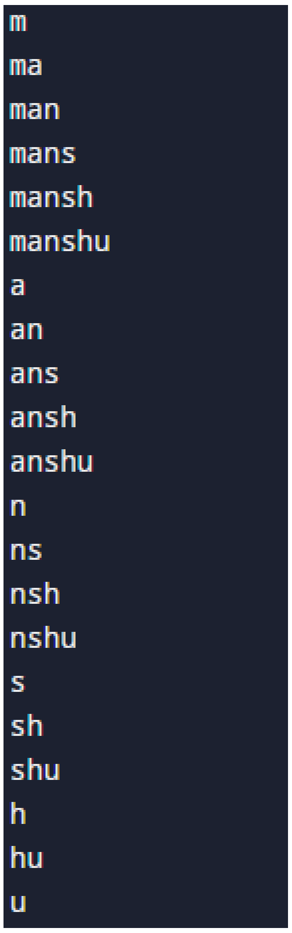
Q12. To find all the subsets of a given string.

A12.

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

        String str = "Himanshu";
        int len = str.length();
        int temp = 0;
        //Total possible subsets for string of size n is n*(n+1)/2
        String arr[] = new String[len*(len+1)/2];

        //This loop maintains the starting character
        for(int i = 0; i < len; i++) {
            //This loop adds the next character every iteration for the subset to form
and add it to the array
            for(int j = i; j < len; j++) {
                arr[temp] = str.substring(i, j+1);
                temp++;
            }
        }

        //This loop prints all the subsets formed from the string.
        System.out.println("All subsets for given string are: ");
        for(int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
```



```
Output
java -cp /tmp/EdCnYOBJl0 AllSubsets
All subsets for given string are:
H
Hi
Him
Hima
Himan
Himans
Himansh
Himanshu
i
im
ima
iman
imans
imansh
imanshu
```



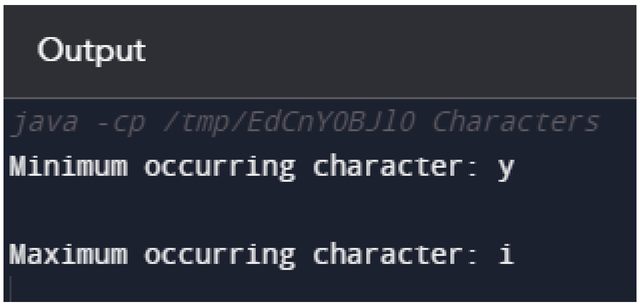
```
m
ma
man
mans
mansh
manshu
a
an
ans
ansh
anshu
n
ns
nsh
nshu
s
sh
shu
h
hu
u
```

Q13. To find the maximum and minimum occurring character in a string.

A13.

```
public class Characters
{
    public static void main(String[] args) {
      String str = "my name is himanshu tiwari";
      int[] freq = new int[str.length()];
      char minChar = str.charAt(0), maxChar = str.charAt(0);
      int i, j, min, max;

      //Converts given string into character array
      char string[] = str.toCharArray();

      //Count each word in given string and store in array freq
      for(i = 0; i < string.length; i++) {
         freq[i] = 1;
         for(j = i+1; j < string.length; j++) {
            if(string[i] == string[j] && string[i] != ' ' && string[i] != '0') {
               freq[i]++;

               //Set string[j] to 0 to avoid printing visited character
               string[j] = '0';
            }
         }
      }

      //Determine minimum and maximum occurring characters
      min = max = freq[0];
      for(i = 0; i <freq.length; i++) {

         //If min is greater than frequency of a character
         //then, store frequency in min and corresponding character in minChar
         if(min > freq[i] && freq[i] != '0') {
            min = freq[i];
            minChar = string[i];
         }
         //If max is less than frequency of a character
         //then, store frequency in max and corresponding character in maxChar
         if(max < freq[i]) {
            max = freq[i];
            maxChar = string[i];
         }
```

```
        }

        System.out.println("Minimum occurring character: " + minChar);
        System.out.println();
        System.out.println("Maximum occurring character: " + maxChar);
    }
}
```

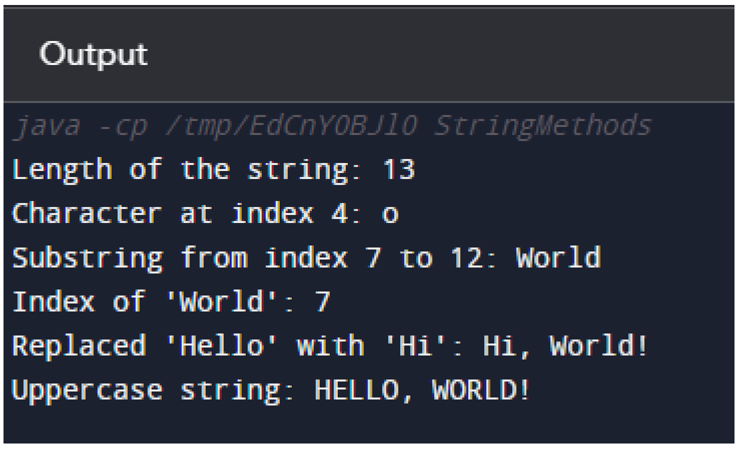


```
Output
java -cp /tmp/EdCnYOBJl0 Characters
Minimum occurring character: y

Maximum occurring character: i
```

Q14. To show the implementation of at least six in-built string methods.

A14.

```
public class StringMethods {
    public static void main(String[] args) {
        String str = "Hello, World!"; // example string

        // length() method - returns the length of the string
        int length = str.length();
        System.out.println("Length of the string: " + length);

        // charAt() method - returns the character at the specified index
        char c = str.charAt(4);
        System.out.println("Character at index 4: " + c);

        // substring() method - returns a substring of the original string
        String substring = str.substring(7, 12);
        System.out.println("Substring from index 7 to 12: " + substring);

        // indexOf() method - returns the index of the first occurrence of a specified
substring
        int index = str.indexOf("World");
        System.out.println("Index of 'World': " + index);

        // replace() method - replaces all occurrences of a specified character or
substring with another character or substring
        String replaced = str.replace("Hello", "Hi");
        System.out.println("Replaced 'Hello' with 'Hi': " + replaced);

        // toUpperCase() method - returns a new string in all uppercase characters
        String upper = str.toUpperCase();
        System.out.println("Uppercase string: " + upper);
    }
}
```

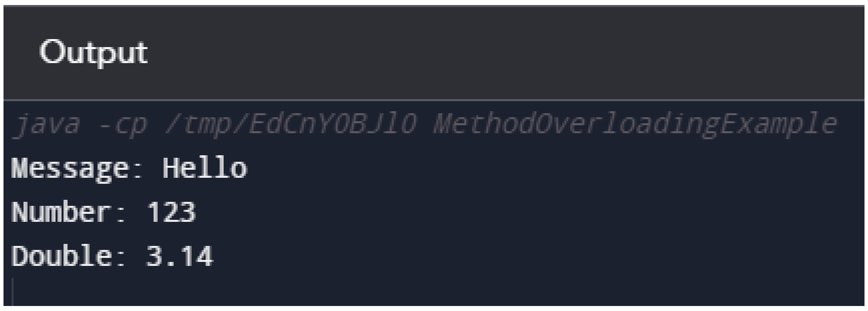


```
Output
java -cp /tmp/EdCnYOBJl0 StringMethods
Length of the string: 13
Character at index 4: o
Substring from index 7 to 12: World
Index of 'World': 7
Replaced 'Hello' with 'Hi': Hi, World!
Uppercase string: HELLO, WORLD!
```

Q15. To show the implementation of method overloading.

A15.

```
public class MethodOverloadingExample {

    public void display(String message) {
        System.out.println("Message: " + message);
    }

    public void display(int number) {
        System.out.println("Number: " + number);
    }

    public void display(double number) {
        System.out.println("Double: " + number);
    }

    public static void main(String[] args) {
        MethodOverloadingExample obj = new MethodOverloadingExample();
        obj.display("Hello");
        obj.display(123);
        obj.display(3.14);
    }
}
```



Output

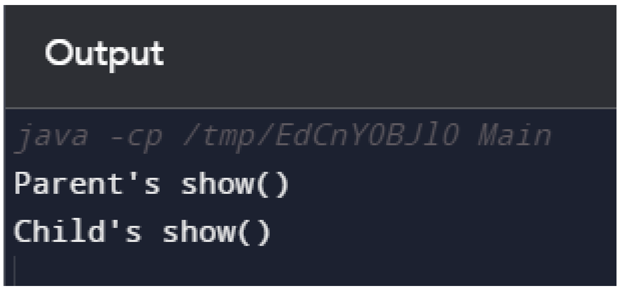
```
java -cp /tmp/EdCnYOBJlO MethodOverloadingExample
Message: Hello
Number: 123
Double: 3.14
```

Q16. To show the implementation of method overriding.

A16.

```
// A Simple Java program to demonstrate
// method overriding in java

// Base Class
class Parent {
	void show()
	{
		System.out.println("Parent's show()");
	}
}

// Inherited class
class Child extends Parent {
	// This method overrides show() of Parent
	@Override
	void show()
	{
		System.out.println("Child's show()");
	}
}

// Driver class
class Main {
	public static void main(String[] args)
	{
		// If a Parent type reference refers
		// to a Parent object, then Parent's
		// show is called
		Parent obj1 = new Parent();
		obj1.show();
```

```
                // If a Parent type reference refers
                // to a Child object Child's show()
                // is called. This is called RUN TIME
                // POLYMORPHISM.
                Parent obj2 = new Child();
                obj2.show();
        }
}
```



Output

```
java -cp /tmp/EdCnYOBJlO Main
Parent's show()
Child's show()
```

Q17. To show the functionality of Encapsulation.

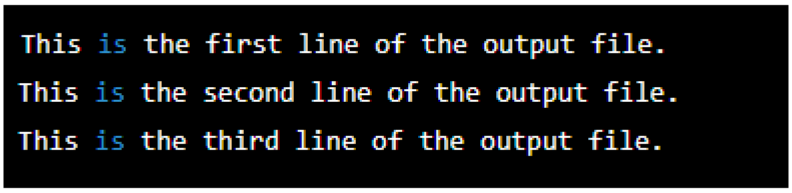
A17.

```
public class EncapsulationExample {
   private String name;
   private int age;

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public int getAge() {
      return age;
   }

   public void setAge(int age) {
      this.age = age;
   }
}

public class EncapsulationDemo {
   public static void main(String[] args) {
      EncapsulationExample obj = new EncapsulationExample();
      obj.setName("John");
      obj.setAge(25);
      System.out.println("Name: " + obj.getName());
      System.out.println("Age: " + obj.getAge());
   }
}
```

Q18. To read and write a text file.

A18.

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class TextFileReadWriteExample {
    public static void main(String[] args) {
        String inputFile = "input.txt";
        String outputFile = "output.txt";

        // read the input file
        try (BufferedReader reader = new BufferedReader(new FileReader(inputFile)))
{
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        // write to the output file
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile))) {
            writer.write("This is the first line of the output file.\n");
            writer.write("This is the second line of the output file.\n");
            writer.write("This is the third line of the output file.\n");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



```
This is the first line of the output file.
This is the second line of the output file.
This is the third line of the output file.
```