```
+---------------------+
| [1] TEST_ARRAY_LIST |
+---------------------+
package Task1 ;

public class Loan {
        private double annualInterestRate ;
        private int numberOfYears ;
        private double loanAmount ;
        private java.util.Date loanDate ;

        public Loan() {
                super();
                loanDate = new java.util.Date();
        }
        public Loan(double annualInterestRate, int numberOfYears, double loanAmount)
{
                super();
                this.annualInterestRate = annualInterestRate;
                this.numberOfYears = numberOfYears;
                this.loanAmount = loanAmount;
                loanDate = new java.util.Date();
        }
        public double getAnnualInterestRate() {
                return annualInterestRate;
        }
        public void setAnnualInterestRate(double annualInterestRate) {
                this.annualInterestRate = annualInterestRate;
        }
        public int getNumberOfYears() {
                return numberOfYears;
        }
        public void setNumberOfYears(int numberOfYears) {
                this.numberOfYears = numberOfYears;
        }
        public double getLoanAmount() {
                return loanAmount;
        }
        public void setLoanAmount(double loanAmount) {
                this.loanAmount = loanAmount;
        }
        public java.util.Date getLoanDate() {
                return loanDate;
        }
        public double getMonthlyPayment() {
                double monthlyInterestRate = annualInterestRate / 1200 ;
                double monthlyPayment = loanAmount * monthlyInterestRate / (1- (1/
Math.pow(1 + monthlyInterestRate, numberOfYears * 12))) ;
                return monthlyPayment ;
        }
```

```
        public double getTotalPayment() {
                return getMonthlyPayment() * numberOfYears * 12 ;
        }
        @Override
        public String toString() {
                return "Loan [annualInterestRate = " + annualInterestRate + ",
numberOfYears = " + numberOfYears + ", loanAmount = "
                                + loanAmount + ", loanDate = " + loanDate + "]";
        }

}
package Task1 ;

public class Circle {
        private double radius = 1 ;
        private static int numberOfObjects = 0 ;

        public Circle() {
                numberOfObjects++ ;
        }
        public Circle(double newRadius) {
                radius = newRadius ;
                numberOfObjects++ ;
        }
        public double getRadius() {
                return radius ;
        }
        public void setRadius(double newRadius) {
                radius = (newRadius >= 0) ? newRadius : 0 ;
        }
        public static int getNumberOfObjects() {
                return numberOfObjects ;
        }
        public double getArea() {
                return radius * radius * Math.PI ;
        }
        public double getPerimter() {
                return Math.PI * radius * 2 ;
        }
        @Override
        public String toString() {
                return "Circle [radius = " + radius + " Created on = " +(new
java.util.Date()).toString()+ "]";
        }

}
package Task1 ;
import java.util.ArrayList ;
import java.util.Date ;
```

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

		ArrayList<Object> list = new ArrayList<>();
		list.add(new Loan());
		list.add(new Circle());
		list.add(new Date());
		list.add(new String("HELLO WORLD"));

		for(int i = 0 ; i < list.size() ; i++) {
			System.out.println(list.get(i).toString());
		}

	}

}
```

```
================================================================================
================================================================================
```

```
+------------------------+
| [2] MAXIMUM_ROW_COLUMN |
+------------------------+
```

```
package Task2;
import java.util.Scanner ;
import java.util.ArrayList ;

public class MaxRowColumn {
	public static void main(String [] args) {
		Scanner input = new Scanner (System.in);
		System.out.println("Enter array size : ");
		int size = input.nextInt();

		int [][] array = new int [size][size];
		array = inputArray(array);
		printArray(array);

		ArrayList<Integer> row = new ArrayList<>();
		ArrayList<Integer> column = new ArrayList<>();

		maxRow(array,row);
		maxColumn(array,column);

		System.out.println("Row with most 1 : " +row.toString());
		System.out.println("Column with most 1 : " +column.toString());

	}

	public static int [][] inputArray(int [][] array) {
		for(int i = 0 ; i < array.length ; i++) {
			for(int j = 0 ; j < array[i].length ; j++) {
				array[i][j] = (int)(Math.random() * 2) ;
```

```
                }
            }
            return array ;
        }
        public static void printArray(int [][] array) {
            for(int i = 0 ; i < array.length ; i++) {
                for(int j = 0 ; j < array[i].length ; j++) {
                    System.out.print(array[i][j] +" ") ;
                }
                System.out.println();
            }
        }
        public static void maxRow(int[][] arr , ArrayList<Integer> row) {
            int max = 0 ;
            for(int i = 0 ; i < arr.length ; i++) {
                int count = 0 ;

                for(int j = 0 ; j < arr[i].length ; j++) {
                    if(arr[i][j] == 1) {
                        count++ ;
                    }
                    if(count > max) {
                        max = count ;
                        row.clear(); // if true then clear the array
then add
                        row.add(i);
                    }
                    else if(count == max) {
                        row.add(i);
                    }
                }
            }
        }

        public static void maxColumn(int[][] arr , ArrayList<Integer> column) {
            int max = 0 ;
            for(int i = 0 ; i < arr[0].length ; i++) {
                int count = 0 ;

                for(int j = 0 ; j < arr.length ; j++) {
                    if(arr[j][i] == 1) {
                        count++ ;
                    }
                    if(count > max) {
                        max = count ;
                        column.clear(); // if true then clear the
array then add
                        column.add(j);
                    }
                    else if(count == max) {
```

```
                                        column.add(j);
                                }
                        }
                }
        }

}
================================================================================
================================================================================
+------------------------+
|  [3] SUM OF ARRAYLIST  |
+------------------------+

package Task3;
import java.util.Scanner ;
import java.util.ArrayList ;

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

                Scanner input = new Scanner (System.in);
                ArrayList<Double> list = new ArrayList<>();

                System.out.println("Enter 5 values : ");
                for(int i = 0 ; i < 5 ; i++) {
                        list.add(input.nextDouble());
                }

                System.out.println("Sum is : " +sum(list));

        }
        public static double sum(ArrayList<Double> list) {
                double sum = 0 ;
                for(int i = 0 ; i < 5 ; i++) {
                        sum+= list.get(i);
                }
                return sum ;
        }

}
================================================================================
================================================================================
+------------------------+
|[4] APPENDIND TWO ARRYS |
+------------------------+
package Task4;
import java.util.* ;

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

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

                ArrayList<Integer> list1 = new ArrayList<>();
                ArrayList<Integer> list2 = new ArrayList<>();

                System.out.println("Enter 5 integrs for list 1 : ");
                for(int i = 0 ; i < 5 ; i++) {
                        list1.add(input.nextInt());
                }
                System.out.println("Enter 5 integrs for list 2 : ");
                for(int i = 0 ; i < 5 ; i++) {
                        list2.add(input.nextInt());
                }

                ArrayList<Integer> list3 = union(list1 , list2);
                System.out.println("Union : " +list3.toString());
        }

        public static ArrayList<Integer> union (ArrayList<Integer>list1
,ArrayList<Integer>list2){
                ArrayList<Integer>list3 = list1 ;
                for(int i = 0 ; i < list2.size() ; i++) {
                        list3.add(list2.get(i));
                }
                return list3 ;
        }

}
================================================================================
================================================================================
+------------------------+
|  [5] MY POINT [ AREA ] |
+------------------------+

package Task5;

import java.util.ArrayList;
import java.util.Scanner;

public class Task5 {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of the points: ");
        int numOfPoints = input.nextInt();

        System.out.print("Enter the coordinates of the points: ");
        ArrayList<MyPoint> points = new ArrayList<>();
        for (int i = 0; i < numOfPoints; i++) {
            points.add(new MyPoint(input.nextDouble(), input.nextDouble()));
```

```
        }


        System.out.println("The total area is " + getConvexPolygonArea(points));

    }

    //  Area of a Convex Polygon
    // http://www.mathwords.com/a/area_convex_polygon.htm
    public static double getConvexPolygonArea(ArrayList<MyPoint> points) {
        // points must be counter clockwise
        double sum1 = 0;
        double sum2 = 0;
        for (int i = 0; i < points.size(); i++) {
            int limitIndex = (i + 1) % points.size();
            MyPoint p1 = points.get(i);
            MyPoint p2 = points.get(limitIndex);
            System.out.println("P1 index = " + i);
            System.out.println("P2 index =" + limitIndex);
            sum1 += (p1.x * p2.y);
            sum2 += (p1.y * p2.x);
        }

        double area = 0.5 * (sum1 - sum2);
        return (area > 0) ? area : -area;
    }

} // COllected from Solution Manual Website //
// have to create MyPoint class to use it //
================================================================================
================================================================================
```