```
//PALINDROME

import java.util.*;
 class Palindrome{
    public static void main(String arr[]){
    String str;
    int flag = 0;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter a string: ");
    str = sc.nextLine();
    int len = str.length();
    str = str.toLowerCase();

    int strLen = len-1;

    for(int i=0;i<len;i++){
        if(str.charAt(i)!=str.charAt(strLen)){
            flag = 1;
            break;
        }
        strLen--;
    }

    if(flag == 1){
        System.out.println("Not palindrome..!!");
    }
    else{
        System.out.println("Is a palindrome..!!");
    }


    }
 }
```

OUTPUT

Output 1

```
Enter a string:
malayalam
Is a palindrome..!!
```

Output 2

```
Enter a string:
nothing
Not palindrome..!!
```

```
//FREQUENCY COUNTER

import java.util.*;
class FrequencyCounter{
        public static void main(String arr[]){

        int count = 0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string");
        String str = sc.nextLine();

        System.out.println("Enter a character: ");
        char a = sc.nextLine().charAt(0);
        for(int i=0;i<str.length();i++){
                if(str.charAt(i)==a){
                        count++;
                }
        }
        System.out.println(""+a+" repeated "+count+" times");



        }
}
```

OUTPUT

```
Enter a string
malayalam
Enter a character: a
a repeated 4 times
```

```
//MATRIX MULTIPLICATION

import java.util.*;

class MatrixMultiplication {
        public static void main(String arr[]) {

                int i, j, k, temp = 0;
                int row2,column2 ;
                int mat1[][] = new int[20][20];
                int mat2[][] = new int[20][20];
                int result[][] = new int[20][20];
                Scanner sc = new Scanner(System.in);

                System.out.println("Enter Matrix 1 size: (row1/column1)");
                int row1 = sc.nextInt();
                int column1 = sc.nextInt();

                System.out.println("Enter the matrix 1 values:");
                for (i = 0; i < row1; i++) {
                        for (j = 0; j < column1; j++) {
                                mat1[i][j] = sc.nextInt();
                        }
                }
                do {
                        System.out.println("Enter Matrix 2 size: (row2/column2)");
                        row2 = sc.nextInt();
                        column2 = sc.nextInt();
                } while (row2 != column1);

                System.out.println("Enter the matrix 2 values:");
                for (i = 0; i < row2; i++) {
                        for (j = 0; j < column2
                        ; j++) {
                                mat2[i][j] = sc.nextInt();
                        }
                }

                for (i = 0; i < row1; i++) {
                        for (j = 0; j < column2; j++) {
                                for (k = 0; k < column1; k++) {
                                        temp = temp + (mat1[i][k] * mat2[k][j]);

                                }
                                result[i][j] = temp;
                                temp = 0;
                        }
                }

                System.out.println("Result :");
                for (i = 0; i < row1; i++) {
                        for (j = 0; j < column2; j++) {
                                System.out.print(result[i][j] + "\t");
                        }
                        System.out.println("\n");
                }
        }}
```

OUTPUT

```
Enter Matrix 1 size: (row1/column1)
3
3
Enter the matrix 1 values:
1
2
3
4
5
6
7
8
9
Enter Matrix 2 size: (row2/column2)
3
3
Enter the matrix 2 values:
1
2
3
4
5
6
7
8
9
Result :
30      36      42

66      81      96

102     126     150
```

```
// INHERITANCE

import java.util.*;
class Employeee{
        Scanner sc = new Scanner(System.in);
        String name;
        String address, phNumber;
        int age, salary;
        Employeee(){
                System.out.println("Enter name  : ");
                name = sc.nextLine();
                System.out.println("Enter age  : ");
                age = sc.nextInt();
                sc.nextLine();
                System.out.println("Enter Phone number  : ");
                phNumber = sc.nextLine();
                System.out.println("Enter address : ");
                address = sc.nextLine();
                System.out.println("Enter salary: ");
                salary = sc.nextInt();
        }
        void printSalary(){
                System.out.println("\n salary: "+salary);
        }
}
class Officer extends Employeee{
        String specialization;
        String department;
        Officer(){
                sc.nextLine();
                System.out.println("Enter specialization : ");
                specialization = sc.nextLine();

                System.out.println("Enter department : ");
                department = sc.nextLine();
        }
}
class Manager extends Employeee{
        String specialization;
        String department;
        Manager(){
                sc.nextLine();
                System.out.println("Enter specialization : ");
                specialization = sc.nextLine();
                System.out.println("Enter department : ");
                department = sc.nextLine();
        }
}
class EmployeeImplement{
        public static void main(String arr[]){

                Officer off = new Officer();
                Manager mng = new Manager();
System.out.println("\nOfficer :\n");
			System.out.println("Name : "+off.name+"\nAge : "+off.age+"\nPh.No
:"+off.phNumber+"\nSalary : "+off.salary+"\nAddress : "+off.address+
			"\nSpecialization : "+off.specialization+"\nDepartment : "+off.department);
			System.out.println("\nManager :\n");
			System.out.println("Name : "+mng.name+"\nAge : "+mng.age+"\nPh.No
:"+mng.phNumber+"\nSalary : "+mng.salary+"\nAddress : "+mng.address+
			"\nSpecialization : "+mng.specialization+"\nDepartment : "+mng.department);
	}
}
```

OUTPUT

```
Enter name  :
Ashwin Wilson
Enter age  :
20
Enter Phone number  :
2323232323
Enter address :
Pathanamthitta
Enter salary:
1000000
Enter specialization :
web development
Enter department :
software development
Enter name  :
Aarav
Enter age  :
20
Enter Phone number  :
23232323
Enter address :
Alapuzha
Enter salary:
1000000
Enter specialization :
graphic designer
Enter department :
design

Officer :

Name : Ashwin Wilson
Age : 20
Ph.No :2323232323
Salary : 1000000
Address : Pathanamthitta
Specialization : web development
Department : software development

Manager :

Name : Aarav
Age : 20
Ph.No :23232323
Salary : 1000000
Address : Alapuzha
Specialization : graphic designer
Department : design
```

```
//ABSTRACT CLASS

import java.util.*;
abstract class Shape{
	abstract void numberOfSides();
}
class Triangle extends Shape{
	void numberOfSides(){
		System.out.println("Triangle : 3");
	}
}
class Rectangle extends Shape{
	void numberOfSides(){
		System.out.println("Rectangle : 4");
	}
}
class Hexagon extends Shape{
	void numberOfSides(){
		System.out.println("Hexagon : 6");
	}
}
class Abstraction{
	public static void main(String arr[]){
		Triangle t = new Triangle();
		t.numberOfSides();
		Rectangle r = new Rectangle();
		r.numberOfSides();
		Hexagon h = new Hexagon();
		h.numberOfSides();
	}
}
```

OUTPUT

```
Triangle : 3
Rectangle : 4
Hexagon : 6
```

```
//GARBAGE COLLECTION

public class GarbageCollectorDemo {

    public static void main(String[] args) {

        MyClass obj1 = new MyClass("Object 1");
        MyClass obj2 = new MyClass("Object 2");

        obj1 = null;
        obj2 = null;

        System.gc();

        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

class MyClass {
    private String name;

    public MyClass(String name) {
        this.name = name;
        System.out.println(name + " created.");
    }

    protected void finalize() {
        System.out.println(name + " is being garbage collected.");
    }
}
```

OUTPUT

```
Object 1 created.
Object 2 created.
Object 2 is being garbage collected.
Object 1 is being garbage collected.
```

```
//FILE WRITE & READ

import java.io.*;
public class Reader {
        public static void main(String[] args) throws IOException {


            FileWriter writer = new FileWriter("input.txt");
            BufferedWriter bufferedWriter = new BufferedWriter(writer);

            bufferedWriter.write("My name is Ashwin wilson");
            bufferedWriter.close();
            writer.close();


            FileReader reader = new FileReader("input.txt");
            BufferedReader bufferedReader = new BufferedReader(reader);

            String line;
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }
            bufferedReader.close();
            reader.close();

          }
}
```

OUTPUT

```
My name is Ashwin wilson
```

```
//READING AND WRITING FILE (including exception handling)

import java.io.*;
public class ReadAndWrite {
         public static void main(String[] args) {
         String inputFileName = "input.txt";
      try {

         FileReader reader = new FileReader(inputFileName);
               BufferedReader bufferedReader = new BufferedReader(reader);
                  FileWriter writer = new FileWriter("output.txt");
               BufferedWriter bufferedWriter = new BufferedWriter(writer);

               String line;
               while ((line = bufferedReader.readLine()) != null) {
                  bufferedWriter.write(line);
               }

               bufferedReader.close();
               reader.close();

               bufferedWriter.close();
               writer.close();



      } catch (FileNotFoundException e) {
         System.err.println("Error: File " + inputFileName + " not found.");
      } catch (IOException e) {
         System.err.println("Error: IOException occurred while reading or writing file.");
         e.printStackTrace();
      }
   }
}
```

OUTPUT

```
“Output.txt” file created
```

```
//STRING TOKENIZER

import java.util.StringTokenizer;

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

        System.out.println("Enter a line of integers separated by spaces:");
        String inputLine = System.console().readLine();


        StringTokenizer tokenizer = new StringTokenizer(inputLine);


        int sum = 0;
        System.out.println("Individual integers:");
        while (tokenizer.hasMoreTokens()) {
            String token = tokenizer.nextToken();
            int number = Integer.parseInt(token);
            System.out.println(number);
            sum += number;
        }


        System.out.println("Sum of all integers: " + sum);
    }
}
```

OUTPUT

```
Enter a line of integers separated by spaces:
1 2 3 4 5
Individual integers:
1
2
3
4
5
Sum of all integers: 15
```

```
//try, catch,throws & finally USAGE

class ThrowsClass{
        void throwException()throws ArithmeticException{
                        int result = 10/0;
        }
}

class Exception{
        public static void main(String arr[]){
                try{
                        ThrowsClass t = new ThrowsClass();
                        t.throwException();
                }
                catch(ArithmeticException e ){
                        System.out.println(e);
                }
                finally{
                        System.out.println("All the Exceptions are handled");
                }
        }
}
```

OUTPUT

```
java.lang.ArithmeticException: / by zero
All the Exceptions are handled
```

```
//MULTITHREAD

import java.util.Random;

class X implements Runnable {
    static int random;
    Random rand = new Random();

    public void run() {
        random = rand.nextInt(25);
        System.out.println(random);
    }
}

class Y implements Runnable {
    public void run() {
        if (X.random % 2 == 0)
            System.out.println((int) Math.pow(X.random, 2) + "\n");
    }
}

class Z implements Runnable {
    public void run() {
        if (X.random % 2 != 0)
            System.out.println((int) Math.pow(X.random, 3) + "\n");
    }
}

class Multithread {
    public static void main(String args[]) {
        for (int i = 0; i < 10; i++) {
            Thread objX = new Thread(new X());
            Thread objY = new Thread(new Y());
            Thread objZ = new Thread(new Z());
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
            objX.start();
            objY.start();
            objZ.start();
        }
    }
}
```

OUTPUT

```
3
27

10
100

20
400

17
4913

20
400

16
256

3
27

24
576

9
729

17
4913
```

```
//SYNCHRONISATION

class First extends Thread{
    void display(String msg){
        System.out.println("["+msg);
        try{
            sleep(1000);
        }catch(InterruptedException e){
             System.out.println(e);
        }
         System.out.println("]");
    }
}
class Second extends Thread{
    First ob;
    String str1;
    Second(First obj, String str){
        ob = obj;
        str1 = str;
        start();
    }
    public void run(){
        synchronized(ob){
            ob.display(str1);
        }
    }
}
class SynchronisationDemo{
    public static void main(String arg[]){
        First F =  new First();
        Second s1 = new Second(F, "Ashwin");
        Second s2 = new Second(F, "Aarav");
        Second s3 = new Second(F, "Anadhu");

    }
}
```

OUTPUT

```
[Ashwin
]
[Aarav
]
[Anadhu
]
```

```
//CALCULATOR

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;

public class Calculator implements ActionListener {
    Boolean isOperator = false;
    Boolean isMultiplication = false;
    Boolean isAddition = false;
    Boolean isSubstraction = false;
    Boolean isDivision = false;
    Boolean isPoint = true;

    float oldValue;
    float nexValue;
    String oldString;
    String newString;

    JLabel display = new JLabel("");

    JButton sevenButton = new JButton("7");
    JButton eightButton = new JButton("8");
    JButton nineButton = new JButton("9");
    JButton fourButton = new JButton("4");
    JButton fiveButton = new JButton("5");
    JButton sixButton = new JButton("6");
    JButton oneButton = new JButton("1");
    JButton twoButton = new JButton("2");
    JButton threeButton = new JButton("3");
    JButton DzeroButton = new JButton("00");
    JButton zeroButton = new JButton("0");
    JButton pointButton = new JButton(".");
    JButton divisionButton = new JButton("/");
    JButton multiButton = new JButton("x");
    JButton additionButton = new JButton("+");
    JButton substractionButton = new JButton("-");
    JButton clearButton = new JButton("C");
    JButton equalButton = new JButton("=");

    Calculator() {
        // FRAME SETUP
        JFrame jf = new JFrame("Calculator");
        jf.setBounds(400, 100, 382, 520);
        jf.setBackground(Color.black);
        jf.getContentPane().setBackground(Color.black);
```

```
// DISPLAY
display.setBounds(4, 30, 373, 93);
display.setBackground(Color.green);
display.setForeground(Color.black);
display.setOpaque(true);
display.setHorizontalAlignment(SwingConstants.RIGHT);
display.setBorder(new EmptyBorder(0, 10, 0, 10));
display.setFont(new Font("Arial", Font.BOLD, 22));

jf.add(display);

// BUTTONS
// row 1
sevenButton.setBounds(4, 130, 120, 50);
sevenButton.setFont(new Font("Arial", Font.PLAIN, 20));
sevenButton.setBackground(Color.darkGray);
sevenButton.setBorderPainted(false);
sevenButton.setFocusPainted(false);
sevenButton.setForeground(Color.lightGray);
sevenButton.addActionListener(this);
jf.add(sevenButton);

eightButton.setBounds(131, 130, 120, 50);
eightButton.setFont(new Font("Arial", Font.PLAIN, 20));
eightButton.setBackground(Color.darkGray);
eightButton.setBorderPainted(false);
eightButton.setFocusPainted(false);
eightButton.setForeground(Color.lightGray);
eightButton.addActionListener(this);

jf.add(eightButton);

nineButton.setBounds(258, 130, 120, 50);
nineButton.setFont(new Font("Arial", Font.PLAIN, 20));
nineButton.setBackground(Color.darkGray);
nineButton.setBorderPainted(false);
nineButton.setFocusPainted(false);
nineButton.setForeground(Color.lightGray);
nineButton.addActionListener(this);

jf.add(nineButton);

// row 2
fourButton.setBounds(4, 190, 120, 50);
fourButton.setFont(new Font("Arial", Font.PLAIN, 20));
fourButton.setBackground(Color.darkGray);
fourButton.setBorderPainted(false);
fourButton.setFocusPainted(false);
fourButton.setForeground(Color.lightGray);
fourButton.addActionListener(this);

jf.add(fourButton);
```

```
fiveButton.setBounds(131, 190, 120, 50);
    fiveButton.setFont(new Font("Arial", Font.PLAIN, 20));
    fiveButton.setBackground(Color.darkGray);
    fiveButton.setBorderPainted(false);
    fiveButton.setFocusPainted(false);
    fiveButton.setForeground(Color.lightGray);
    fiveButton.addActionListener(this);

    jf.add(fiveButton);

    sixButton.setBounds(258, 190, 120, 50);
    sixButton.setFont(new Font("Arial", Font.PLAIN, 20));
    sixButton.setBackground(Color.darkGray);
    sixButton.setBorderPainted(false);
    sixButton.setFocusPainted(false);
    sixButton.setForeground(Color.lightGray);
    sixButton.addActionListener(this);

    jf.add(sixButton);

    // row 3
    oneButton.setBounds(4, 250, 120, 50);
    oneButton.setFont(new Font("Arial", Font.PLAIN, 20));
    oneButton.setBackground(Color.darkGray);
    oneButton.setBorderPainted(false);
    oneButton.setFocusPainted(false);
    oneButton.setForeground(Color.lightGray);
    oneButton.addActionListener(this);

    jf.add(oneButton);

    twoButton.setBounds(131, 250, 120, 50);
    twoButton.setFont(new Font("Arial", Font.PLAIN, 20));
    twoButton.setBackground(Color.darkGray);
    twoButton.setBorderPainted(false);
    twoButton.setFocusPainted(false);
    twoButton.setForeground(Color.lightGray);
    twoButton.addActionListener(this);

    jf.add(twoButton);

    threeButton.setBounds(258, 250, 120, 50);
    threeButton.setFont(new Font("Arial", Font.PLAIN, 20));
    threeButton.setBackground(Color.darkGray);
    threeButton.setBorderPainted(false);
    threeButton.setFocusPainted(false);
    threeButton.setForeground(Color.lightGray);
    threeButton.addActionListener(this);

    jf.add(threeButton);

    // row 4
    DzeroButton.setBounds(4, 310, 120, 50);
DzeroButton.setFont(new Font("Arial", Font.PLAIN, 20));
DzeroButton.setBackground(Color.darkGray);
DzeroButton.setBorderPainted(false);
DzeroButton.setFocusPainted(false);
DzeroButton.setForeground(Color.lightGray);
DzeroButton.addActionListener(this);

jf.add(DzeroButton);

zeroButton.setBounds(131, 310, 120, 50);
zeroButton.setFont(new Font("Arial", Font.PLAIN, 20));
zeroButton.setBackground(Color.darkGray);
zeroButton.setBorderPainted(false);
zeroButton.setFocusPainted(false);
zeroButton.setForeground(Color.lightGray);
zeroButton.addActionListener(this);

jf.add(zeroButton);

pointButton.setBounds(258, 310, 120, 50);
pointButton.setFont(new Font("Arial", Font.PLAIN, 20));
pointButton.setBackground(Color.darkGray);
pointButton.setBorderPainted(false);
pointButton.setFocusPainted(false);
pointButton.setForeground(Color.lightGray);
pointButton.addActionListener(this);

jf.add(pointButton);

// row 5
divisionButton.setBounds(4, 370, 120, 50);
divisionButton.setFont(new Font("Arial", Font.PLAIN, 20));
divisionButton.setBackground(Color.darkGray);
divisionButton.setBorderPainted(false);
divisionButton.setFocusPainted(false);
divisionButton.setForeground(Color.lightGray);
divisionButton.addActionListener(this);
jf.add(divisionButton);

multiButton.setBounds(131, 370, 120, 50);
multiButton.setFont(new Font("Arial", Font.PLAIN, 20));
multiButton.setBackground(Color.darkGray);
multiButton.setBorderPainted(false);
multiButton.setFocusPainted(false);
multiButton.setForeground(Color.lightGray);
multiButton.addActionListener(this);

jf.add(multiButton);

additionButton.setBounds(258, 370, 120, 50);
additionButton.setFont(new Font("Arial", Font.PLAIN, 20));
additionButton.setBackground(Color.darkGray);
additionButton.setBorderPainted(false);
```

```
 additionButton.setFocusPainted(false);
    additionButton.setForeground(Color.lightGray);
    additionButton.addActionListener(this);

    jf.add(additionButton);

    // row 6
    substractionButton.setBounds(4, 430, 120, 50);
    substractionButton.setFont(new Font("Arial", Font.PLAIN, 20));
    substractionButton.setBackground(Color.darkGray);
    substractionButton.setBorderPainted(false);
    substractionButton.setFocusPainted(false);
    substractionButton.setForeground(Color.lightGray);
    substractionButton.addActionListener(this);

    jf.add(substractionButton);

    clearButton.setBounds(131, 430, 120, 50);
    clearButton.setFont(new Font("Arial", Font.PLAIN, 20));
    clearButton.setBackground(Color.darkGray);
    clearButton.setBorderPainted(false);
    clearButton.setFocusPainted(false);
    clearButton.setForeground(Color.lightGray);
    clearButton.addActionListener(this);
    jf.add(clearButton);

    equalButton.setBounds(258, 430, 120, 50);
    equalButton.setFont(new Font("Arial", Font.PLAIN, 20));
    equalButton.setBackground(Color.darkGray);
    equalButton.setBorderPainted(false);
    equalButton.setFocusPainted(false);
    equalButton.setForeground(Color.lightGray);
    equalButton.addActionListener(this);

    jf.add(equalButton);

    jf.setLayout(null);
    jf.setVisible(true);
    jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }

  public static void main(String[] args) {
    System.out.println("claculator");
    new Calculator();

  }

  public void actionPerformed(ActionEvent e) {
    // number section
    if (e.getSource() == sevenButton) {
      if (isOperator) {
        oldString = display.getText();
        display.setText("7");
    isOperator = false;
    } else {
        display.setText(display.getText() + "7");
    }
} else if (e.getSource() == eightButton) {
    if (isOperator) {
        oldString = display.getText();
        display.setText("8");
        isOperator = false;
    } else {
        display.setText(display.getText() + "8");
    }
} else if (e.getSource() == nineButton) {
    if (isOperator) {
        oldString = display.getText();
        display.setText("9");
        isOperator = false;
    } else {
        display.setText(display.getText() + "9");
    }
} else if (e.getSource() == fourButton) {
    if (isOperator) {
        oldString = display.getText();
        display.setText("4");
        isOperator = false;
    } else {
        display.setText(display.getText() + "4");
    }
} else if (e.getSource() == fiveButton) {
    if (isOperator) {
        oldString = display.getText();
        display.setText("5");
        isOperator = false;
    } else {
        display.setText(display.getText() + "5");
    }
} else if (e.getSource() == sixButton) {
    if (isOperator) {
        oldString = display.getText();
        display.setText("6");
        isOperator = false;
    } else {
        display.setText(display.getText() + "6");
    }
} else if (e.getSource() == oneButton) {
    if (isOperator) {
        oldString = display.getText();
        display.setText("1");
        isOperator = false;
    } else {
        display.setText(display.getText() + "1");
    }
} else if (e.getSource() == twoButton) {
```

```
if (isOperator) {
            oldString = display.getText();
            display.setText("2");
            isOperator = false;
        } else {
            display.setText(display.getText() + "2");
        }
    } else if (e.getSource() == threeButton) {
        if (isOperator) {
            oldString = display.getText();
            display.setText("3");
            isOperator = false;
        } else {
            display.setText(display.getText() + "3");
        }
    } else if (e.getSource() == pointButton) {
        if (isPoint) {
            display.setText(display.getText() + ".");
            isPoint = false;
        }
    } else if (e.getSource() == zeroButton) {
        if (isOperator) {
            oldString = display.getText();
            display.setText("0");
            isOperator = false;
        } else {
            display.setText(display.getText() + "0");

        }
    } else if (e.getSource() == DzeroButton) {
        if (isOperator) {
            oldString = display.getText();
            display.setText("00");
            isOperator = false;
        } else {
            display.setText(display.getText() + "00");

        }
    } else if (e.getSource() == clearButton) {
        display.setText("");
    }

    // operator section
    if (e.getSource() == divisionButton) {
        isOperator = true;
        isDivision = true;
        isPoint = true;
    } else if (e.getSource() == multiButton) {
        isOperator = true;
        isMultiplication = true;
        isPoint = true;
    } else if (e.getSource() == additionButton) {
        isOperator = true;
```

```
   isAddition = true;
            isPoint = true;
        } else if (e.getSource() == substractionButton) {
            isOperator = true;
            isSubstraction = true;
            isPoint = true;
        } else if (e.getSource() == equalButton) {
            newString = display.getText();
            if (isDivision) {
                display.setText("" + (Float.parseFloat(oldString) / Float.parseFloat(newString)));
                isDivision = false;
            } else if (isMultiplication) {
                display.setText("" + (Float.parseFloat(oldString) * Float.parseFloat(newString)));
                isMultiplication = false;
            } else if (isAddition) {
                display.setText("" + (Float.parseFloat(oldString) + Float.parseFloat(newString)));
                isAddition = false;
            } else if (isSubstraction) {
                display.setText("" + (Float.parseFloat(oldString) - Float.parseFloat(newString)));
                isSubstraction = false;
            }
        }

        throw new UnsupportedOperationException("Unimplemented method 'actionPerformed'");
    }
}
```

```
//TRAFFIC LIGHT SIMULATOR

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class TrafficLight extends JPanel implements ActionListener {
    private JRadioButton r1;
    private JRadioButton r2;
    private JRadioButton r3;
    private Color red_c;
    private Color green_c;
    private Color yellow_c;

    public TrafficLight() {
        setBounds(0, 0, 500, 480);
        r1 = new JRadioButton("Red");
        r2 = new JRadioButton("Yellow");
        r3 = new JRadioButton("Green");
        red_c = getBackground();
        yellow_c = getBackground();
        green_c = getBackground();
        ButtonGroup gp = new ButtonGroup();
        gp.add(r1);
        gp.add(r2);
        gp.add(r3);
        add(r1);
        add(r2);
        add(r3);
        r1.addActionListener(this);
        r2.addActionListener(this);
        r3.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (r1.isSelected()) {
            red_c = Color.red;
            yellow_c = getBackground();
            green_c = getBackground();
        } else if (r2.isSelected()) {
            red_c = getBackground();
            yellow_c = Color.yellow;
            green_c = getBackground();
        } else if (r3.isSelected()) {
            red_c = getBackground();
            yellow_c = getBackground();
            green_c = Color.green;
        }
        repaint();
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
```

```
       g.drawOval(50, 50, 50, 50);
        g.drawOval(50, 110, 50, 50);
        g.drawOval(50, 170, 50, 50);
        g.setColor(red_c);
        g.fillOval(50, 50, 50, 50);
        g.setColor(yellow_c);
        g.fillOval(50, 110, 50, 50);
        g.setColor(green_c);
        g.fillOval(50, 170, 50, 50);
    }
}

class TrafficLightSimulator {
    public static void main(String[] args) {
        JFrame f = new JFrame();
        f.setVisible(true);
        f.setSize(500, 480);
        f.setLayout(null);
        TrafficLight t = new TrafficLight();
        f.add(t);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```
//QUICK SORT

import java.util.*;

class Quicksort{
    static void quickSort(String names[],int low,int high){
        if(low<high){
            int l=low,h=high,p=low;
            String temp;
            while(l<h){
                while( (names[l].compareTo(names[p]) <=0)&& l<h){
                    l++;
                }
                while(names[h].compareTo(names[p]) > 0){
                    h--;
                }
                if(l<h){
                    temp = names[l];
                    names[l] = names[h];
                    names[h] = temp;
                }
                temp = names[h];
                names[h] = names[p];
                names[p] = temp;

                quickSort(names,0,h-1);
                quickSort(names,h+1,high);
            }
        }

    }

    public static void main(String arr[]){

        Scanner sc = new Scanner(System.in);
        System.out.println("How many names do you have: ");
        int n = sc.nextInt();
        String names[] = new String[n];

        sc.nextLine();
        for(int i=0;i<n;i++){

            System.out.println("Enter name "+(i+1)+": ");
            names[i] = sc.nextLine();
        }

        quickSort(names,0,n-1);

        System.out.println("Sorted list:");
        for(String i : names){
            System.out.println(i);
        }
    }}
```

OUTPUT

```
How many names do you have: 5
Enter name 1:
Ashwin
Enter name 2:
Aarav
Enter name 3:
Anandhu
Enter name 4:
Abin
Enter name 5:
Rohit
Sorted list:
Aarav
Abin
Anandhu
Ashwin
Rohit
```

```
//BINARY SEARCH

import java.util.*;
class BinarySearch{
    public static void main(String arg[]){
        Scanner sc = new Scanner(System.in);
        int i,j,limit,beg,mid,end,item,temp;

        System.out.println("Enter size of the array: ");
        limit = sc.nextInt();
        int arr[] = new int[limit];
        System.out.println("Enter "+limit+" numbers : \n");
        for(i=0;i<limit;i++){
            arr[i] = sc.nextInt();
        }

        for(j=limit-1;j>=0;j--){
            for(i=0;i<j;i++ ){
                if(arr[i]>arr[i+1]){
                    temp = arr[i];
                    arr[i] = arr[i+1];
                    arr[i+1] = temp;
                }
            }
        }
        System.out.println("Sorted array: \n");
        for(i=0;i<limit;i++){
            System.out.println("\t"+arr[i]);
        }
        System.out.println("Enter item to search");
        item = sc.nextInt();
        beg = 0;
        end = limit-1;
        mid = (beg+end)/2;
        while((arr[mid] != item) && (beg<=end)){
            if(item < arr[mid]){
                end = mid-1;
                mid = (beg+end)/2;
            }
            else{
                beg = mid+1;
                mid = (beg+end)/2;
            }
        }
        if(arr[mid] == item){
            System.out.println("Element found at : "+ (mid+1));
        }
        else{
            System.out.println("Element not found\n");
        }

    }
}
```

OUTPUT

```
Enter size of the array:
5
Enter 5 numbers :

2
6
3
4
1
Sorted array:

        1
        2
        3
        4
        6
Enter item to search
4
Element found at : 4
```

```
//DOUBLY LINKED LIST

import java.util.Scanner;

public class DLinkedList {
    class Node {
        int data;
        Node next;
        Node prev;

        Node(int data) {
            this.data = data;
            this.prev = null;
            this.next = null;
        }
    }

    public Node head;

    public void addNode(int data) {
        Node newNode = new Node(data);
        if (head == null)
            head = newNode;
        else {
            Node temp = head;
            while (temp.next != null)
                temp = temp.next;
            temp.next = newNode;
            newNode.prev = temp;
        }
    }

    public void removeNode(int data) {
        if (head == null) {
            System.out.println("List empty");
            return;
        }
        if (head.data == data) {
            if (head.next != null)
                head.next.prev = null;
            head = head.next;
            return;
        }
        Node temp = head;
        while (temp != null) {
            if (temp.data == data)
                break;
            temp = temp.next;
        }
        if (temp == null) {
            System.out.println("data not found");
            return;
        }
if (temp.next != null)
            temp.next.prev = temp.prev;
        temp.prev.next = temp.next;
    }

    public void display() {
        if (head == null) {
            System.out.println("Empty List ");
            return;
        }
        Node temp = head;
        System.out.print("List :");
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.next;
        }
    }

    public static void main(String args[]) {
        DLinkedList list = new DLinkedList();
        while (true) {
            System.out.println("\nEnter \n1. add Node\n2 remove Node\n3 exit");
            Scanner sc = new Scanner(System.in);
            char ch = sc.nextLine().charAt(0);
            switch (ch) {
                case '1':
                    System.out.print("enter the data :");
                    list.addNode(sc.nextInt());
                    list.display();
                    break;
                case '2':
                    System.out.print("enter the data to be delete:");
                    list.removeNode(sc.nextInt());
                    list.display();
                    break;
                case '3':
                    return;
            }
        }
    }
}
```

OUTPUT

```
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :3
List :3              1
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :2
List :3 2
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :8
List :3 2 8
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :21
List :3 2 8 21
Enter
1. add Node
2 remove Node
3 exit
1
enter the data :4
List :3 2 8 21 4
Enter
1. add Node
2 remove Node
3 exit
2
enter the data to be delete:8
List :3 2 21 4
Enter
1. add Node
2 remove Node
3 exit
2
enter the data to be delete:21
List :3 2 4
          Enter
1. add Node
2 remove Node
3 exit

enter the data :5
List :3 2 4 5
          Enter
1. add Node
2 remove Node
3 exit
3
```