```java
import java.util.Scanner;
public class Binary {
    public static void main(String[] args) {
        int i, n, m, item, loc = -1;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no. of elements in the array:");
        n = sc.nextInt();
        int[] a = new int[n];
        System.out.println("Enter array elements:");
        for (i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }
        System.out.println("Enter the element to be searched:");
        item = sc.nextInt();
        sc.close();
        int f = 0, l = n - 1;
        while (f <= l) {
            m = (f + l) / 2;
            if (a[m] == item) {
                loc = m;
                break;
            } else if (item < a[m]) {
                l = m - 1;
            } else {
                f = m + 1;
            }
        }
        if (loc != -1) {
            System.out.println("Element found at position = " + (loc + 1));
        } else {
            System.out.println("Element not found!!!!");
        }
    }
}
```

```
Enter the no. of elements in the array:
7
Enter array elements:
1 2 3 4 5 6 7
Enter the element to be searched:
4
Element found at position = 4
```

```java
import java.util.Scanner;
import java.lang.System;
public class DoublyLinkedList {
    static class Node {
        int data;
        Node next;
        Node prev;
        Node(int data) {
            this.data = data;
        }
    }
    static Node head = null;
    static Node tail = null;
    static void addNode(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
            tail = newNode;
        } else {
            tail.next = newNode;
            newNode.prev = tail;
            tail = newNode;
        }
    }
    static void deleteNode(int data) {
        Node current = head;
        while (current != null) {
            if (current.data == data) {
                if (current == head) {
                    head = head.next;
                    head.prev = null;
                } else if (current == tail) {
                    tail = tail.prev;
                    tail.next = null;
                } else {
                    current.prev.next = current.next;
                    current.next.prev = current.prev;
                }
                break;
            }
            current = current.next;
        }
    }
    static void displayList() {
        Node current = head;
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.next;
```

```
        }
        System.out.println();
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int c;
        System.out.println("Enter the number of elements to add to the list:");
        int numElements = sc.nextInt();

        System.out.println("Enter the elements:");
        for (int i = 0; i < numElements; i++) {
            int element = sc.nextInt();
            addNode(element);
        }
        System.out.println("Original list:");
        displayList();
        while (true) {
System.out.print("\nEnter the choice\n1.Delete an element\n2.display\nPress any other key toexit!\n");
            c = sc.nextInt();
            switch (c) {
                case 1:
                    System.out.println("Enter the element to delete:");
                    int elementToDelete = sc.nextInt();
                    deleteNode(elementToDelete);
                    break;
                case 2:
                    System.out.println("List after deleting the element:");
                    displayList();
                    break;
                default:
                    System.exit(0);
                    break;
            }
        }
    }
}
```

```
Original list:
1 2 3 4 5

Enter the choice
1.Delete an element
2.display
Press any other key to exit!
1
Enter the element to delete:
4

Enter the choice
1.Delete an element
2.display
Press any other key to exit!
2
List after deleting the element:
1 2 3 5
```

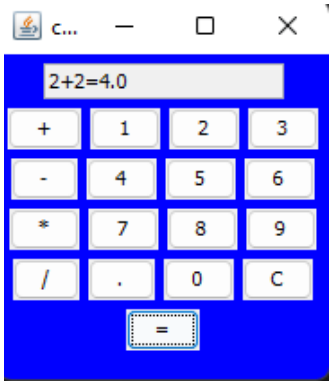
```
Enter the no. of names
4
Enter the strings
lloris alice oliver pepe
[alice, lloris, oliver, pepe]
```

```java
import java.util.Arrays;
import java.util.Scanner;
public class QuickSort {
    public static void quickSort(String[] names, int low, int high) {
        if (low >= high) {
            return;
        }
        int pivotIndex = (low + high) / 2;
        int partitionIndex = partition(names, low, high, pivotIndex);
        quickSort(names, low, partitionIndex - 1);
        quickSort(names, partitionIndex + 1, high);
    }
    private static int partition(String[] names, int low, int high, int
pivotIndex) {
        swap(names, pivotIndex, high);
        int partitionIndex = low;
        for (int i = low; i < high; i++) {
            if (names[i].compareTo(names[high]) < 0) {
                swap(names, i, partitionIndex);
                partitionIndex++;
            }
        }
        swap(names, partitionIndex, high);

        return partitionIndex;
    }
    private static void swap(String[] names, int i, int j) {
        String temp = names[i];
        names[i] = names[j];
        names[j] = temp;
    }
    public static void main(String[] args) {
        int n;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no. of names");
        n = sc.nextInt();
        String[] names = new String[n];
        System.out.println("Enter the strings");
        for (int i = 0; i < n; i++) {
            names[i] = sc.next();
        }
        sc.close();
        quickSort(names, 0, names.length - 1);
        System.out.println(Arrays.toString(names));
    }
}
```

```
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;
class MyCalculator extends JFrame implements ActionListener {
    static JFrame f;
    static JTextField l;
    String s0, s1, s2;
    MyCalculator() {
        s0 = s1 = s2 = "";
    }
    public static void main(String args[]) {
        f = new JFrame("calculator");
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            System.err.println(e.getMessage());
        }
        MyCalculator c = new MyCalculator();
        l = new JTextField(16);
        l.setEditable(false);
        JButton ba, bs, bd, bm, be, beq, beq1;
        JButton b[] = new JButton[10];
        for (int i = 0; i < 10; i++) {
            String num = Integer.toString(i);
            b[i] = new JButton(num);
        }
        beq1 = new JButton("=");
        ba = new JButton("+");
        bs = new JButton("-");
        bd = new JButton("/");
        bm = new JButton("*");
        beq = new JButton("C");
        be = new JButton(".");
        JPanel p = new JPanel();
        bm.addActionListener(c);
        bd.addActionListener(c);
        bs.addActionListener(c);
        ba.addActionListener(c);
        be.addActionListener(c);
        beq.addActionListener(c);
        beq1.addActionListener(c);
        for (int i = 0; i < 10; i++) {
            b[i].addActionListener(c);
        }
        p.add(l);
        p.add(ba);
        p.add(b[1]);
        p.add(b[2]);
```

```
        p.add(b[3]);
        p.add(bs);
        p.add(b[4]);
        p.add(b[5]);
        p.add(b[6]);
        p.add(bm);
        p.add(b[7]);
        p.add(b[8]);
        p.add(b[9]);
        p.add(bd);
        p.add(be);
        p.add(b[0]);
        p.add(beq);
        p.add(beq1);
        p.setBackground(Color.blue);
        f.add(p);
        f.setSize(200, 220);
        f.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        String s = e.getActionCommand();
        if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {
            if (!s1.equals(""))
                s2 = s2 + s;
            else
                s0 = s0 + s;
            l.setText(s0 + s1 + s2);
        } else if (s.charAt(0) == 'C') {
            s0 = s1 = s2 = "";
            l.setText(s0 + s1 + s2);
        } else if (s.charAt(0) == '=') {
            double te;
            if (s1.equals("+"))
                te = (Double.parseDouble(s0) + Double.parseDouble(s2));
            else if (s1.equals("-"))
                te = (Double.parseDouble(s0) - Double.parseDouble(s2));
            else if (s1.equals("/"))
                te = (Double.parseDouble(s0) / Double.parseDouble(s2));
            else
                te = (Double.parseDouble(s0) * Double.parseDouble(s2));
            l.setText(s0 + s1 + s2 + "=" + te);
            s0 = Double.toString(te);
            s1 = s2 = "";
        } else {
            if (s1.equals("") || s2.equals(""))
                s1 = s;
            else {
```

```
                double te;
                if (s1.equals("+"))
                    te = (Double.parseDouble(s0) + Double.parseDouble(s2));
                else if (s1.equals("-"))
                    te = (Double.parseDouble(s0) - Double.parseDouble(s2));
                else if (s1.equals("/"))
                    te = (Double.parseDouble(s0) / Double.parseDouble(s2));
                else
                    te = (Double.parseDouble(s0) * Double.parseDouble(s2));
                s0 = Double.toString(te);
                s1 = s;
                s2 = "";
            }
            l.setText(s0 + s1 + s2);
        }
    }
}
```




```
P of obj1 = 5
P of obj2 = 5

P of obj1 = 8
P of obj2 = 9
```

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class TrafficSignal
```

```
    extends JFrame implements ItemListener {
JRadioButton jr1;
JRadioButton jr2;
JRadioButton jr3;
JTextField j1 = new JTextField(10);
ButtonGroup b = new ButtonGroup();
String msg = " ";
int x = 0, y = 0, z = 0;
public TrafficSignal(String msg) {
    super(msg);
    setLayout(new FlowLayout());
    jr1 = new JRadioButton("Red");
    jr2 = new JRadioButton("Yellow");
    jr3 = new JRadioButton("Green");
    jr1.addItemListener(this);
    jr2.addItemListener(this);
    jr3.addItemListener(this);
    add(jr1);
    add(jr2);
    add(jr3);
    b.add(jr1);
    b.add(jr2);
    b.add(jr3);
    add(j1);
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
}
public void itemStateChanged(ItemEvent ie) {
    if (ie.getSource() == jr1) {
        if (ie.getStateChange() == 1) {
            msg = "Stop!";
            x = 1;
            repaint();
        } else {
            msg = "";
        }
    }
    if (ie.getSource() == jr2) {
        if (ie.getStateChange() == 1) {
            msg = "Get Ready to go!";
            y = 1;
            repaint();
        } else {
            msg = "";
        }
```

```
        }
        if (ie.getSource() == jr3) {
            if (ie.getStateChange() == 1) {
                msg = "Go!!";
                z = 1;
                repaint();
            } else {
                msg = "";
            }
        }
        j1.setText(msg);
    }
    public void paint(Graphics g) {
        g.drawRect(100, 105, 110, 270);
        g.drawOval(120, 150, 60, 60);
        g.drawOval(120, 230, 60, 60);
        g.drawOval(120, 300, 60, 60);
        if (x == 1) {
            g.setColor(Color.RED);
            g.fillOval(120, 150, 60, 60);
            g.setColor(Color.WHITE);
            g.fillOval(120, 230, 60, 60);
            g.setColor(Color.WHITE);
            g.fillOval(120, 300, 60, 60);
            x = 0;
        }
        if (y == 1) {
            g.setColor(Color.WHITE);
            g.fillOval(120, 150, 60, 60);
            g.setColor(Color.YELLOW);
            g.fillOval(120, 230, 60, 60);
            g.setColor(Color.WHITE);
            g.fillOval(120, 300, 60, 60);
            y = 0;
        }
        if (z == 1) {
            g.setColor(Color.WHITE);
            g.fillOval(120, 150, 60, 60);
            g.setColor(Color.WHITE);
            g.fillOval(120, 230, 60, 60);
            g.setColor(Color.GREEN);
            g.fillOval(120, 300, 60, 60);
            z = 0;
        }
    }
    public static void main(String args[]) {
        JFrame jf = new TrafficSignal("Traffic Light");
        jf.setSize(500, 500);
```

```
        jf.setVisible(true);
    }
}
```

```
public class C31
{
    static void fun() throws IllegalAccessException
    {
        System.out.println("Inside fun(). ");
        throw new IllegalAccessException("demo");
    }

    public static void main(String[] args) {

        try
        {
            fun();
        }
        catch(IllegalAccessException e)
        {
            System.out.println("Exception caught in main.");
        }
        finally{
            System.out.println("Code Ended.....");
        }

    }
}
```

```
Exception caught in main.
Code Ended.....
```

```
87 658503

65 274625

4 16

35 42875

72 5184

83 571787
```

```java
class MyException extends Exception{
    public MyException(String s)
    {
        super(s);
    }
}
public class Userexc{
    public static void main(String[] args) {
        try
        {
            throw new MyException("Abhiram");
        }
        catch (MyException ex) {
            System.out.println("Caught");
            System.out.println(ex.getMessage());
        }
        finally{
            System.out.println("Code Ended.....");
        }
    }
}
```

```
Caught
Abhiram
Code Ended.....
```

```java
public class example1 extends Thread {
    int i;
    static int k = 0;

    public static void main(String[] args) {
        int i, n;
        try {
            for (i = 5; i >= 0; i--) {
                n = (int) Math.floor(Math.random() * 100);
                k = n;
                System.out.print(n + " ");
                Even E = new Even();
                Odd O = new Odd();
                E.start();
                O.start();
                Thread.sleep(1000);
                System.out.println('\n');
            }
        } catch (InterruptedException e) {
```

```
            System.out.println(e);
        }
    }
}
class Even extends Thread {
    example1 ob = new example1();
    int num;
    public void run() {
        try {
            int num = ob.k;
            if (num % 2 == 0) {
                System.out.print(num * num + " ");
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}
class Odd extends Thread {
    example1 ob = new example1();
    int num;
    public void run() {
        try {
            num = ob.k;
            if (num % 2 != 0) {
                System.out.print(num * num * num + " ");
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}
```

```
import java.util.*;

class multiple {
    synchronized void printMultiple(int n) {
        for (int i = 1; i < 4; i++) {
            System.out.print(i * n + "\t");
        }
    }

}
class ThreadOne extends Thread {
    multiple m;
```

```
    ThreadOne(multiple m) {
        this.m = m;
    }
    public void run() {
        m.printMultiple(2);
    }
}
class ThreadTwo extends Thread {
    multiple m;
    ThreadTwo(multiple m) {
        this.m = m;
    }
    public void run() {
        m.printMultiple(50);
    }
}
public class threadsync {
    public static void main(String[] args) {
        multiple m = new multiple();
        ThreadOne t1 = new ThreadOne(m);
        ThreadTwo t2 = new ThreadTwo(m);
        t1.start();
        t2.start();
    }
}
```

```
import java.util.*;
class multiple {
    synchronized void printMultiple(int n) {
        if (n == 1) {
            System.out.println("ODD B/W 1 and 100 : ");
            for (int i = 1; i < 100; i++) {
                if (i % 2 != 0) {
                    System.out.print(i + "\t");
                }

            }
            System.out.println(" ");
        } else {
            System.out.println("EVEN B/W 1 and 100 : ");
            for (int j = 1; j < 100; j++) {
                if (j % 2 == 0) {
                    System.out.print(j + "\t");
                }

            }
            System.out.println(" ");
        }
```

```
    }
}
class ThreadOne extends Thread {
    multiple m;
    ThreadOne(multiple m) {
        this.m = m;
    }
    public void run() {
        m.printMultiple(1);
    }
}
class ThreadTwo extends Thread {
    multiple m;
    ThreadTwo(multiple m) {
        this.m = m;
    }
    public void run() {
        m.printMultiple(2);
    }
}
public class threadNum {
    public static void main(String[] args) {
        multiple m = new multiple();
        ThreadOne t1 = new ThreadOne(m);
        ThreadTwo t2 = new ThreadTwo(m);
        t1.start();
        t2.start();
    }
}
```

```
import java.util.*;
class threadP extends Thread {
    public void run() {
        System.out.println("BLABLABLABLA...");
    }
}
class threadQ extends Thread {
    public void run() {
        System.out.println("BLABLABLABLA...");
    }
}
public class threadPrior {
    public static void main(String[] args) {
        threadP obj1 = new threadP();
        threadQ obj2 = new threadQ();
        System.out.println("P of obj1 = " + obj1.getPriority());
        System.out.println("P of obj2 = " + obj2.getPriority());
        System.out.println(" ");
```

```
        obj1.setPriority(8);
        obj2.setPriority(9);
        System.out.println("P of obj1 = " + obj1.getPriority());
        System.out.println("P of obj2 = " + obj2.getPriority());
    }
}
```

```
import java.io.*;
import java.util.*;
class file {
    public static void main(String[] args) throws IOException {
        int c, d = 1;
        while (d != 0) {
            System.out.print("Select Choice :\n 1.Write on file \n 2.Read from
file\n 3.Exit\n");
            System.out.println(" ");
            Scanner sc = new Scanner(System.in);
            c = sc.nextInt();
            System.out.println(" ");
            switch (c) {
                case 1:
                    System.out.print("Enter Text : ");
                    Scanner ss = new Scanner(System.in);
                    String word = ss.nextLine();
                    FileWriter fw = new FileWriter("output.txt");
                    for (int i = 0; i < word.length(); i++) {
                        fw.write(word.charAt(i));
                    }
                    fw.close();
                    System.out.println("writing Completed !!");
                    System.out.println(" ");
                    break;
                case 2:
                    File txt = new File("output.txt");
                    FileReader fr = new FileReader(txt);
                    char[] content = new char[100];
                    fr.read(content);
                    System.out.println(content);
                    System.out.print("--END--");
                    System.out.println(" ");
                    break;
                case 3:
                    d = 0;
                    break;
                default:
                    System.out.println("INVALID INPUT");
            }
        }
```

```
    }
}
```

```
import java.io.*;
import java.util.*;

class lineno {
    public static void main(String[] args) throws IOException {
        int i = 0, l = 2;
        File txt = new File("line.txt");
        FileReader fr = new FileReader(txt);
        char[] content = new char[100];
        fr.read(content);
        if (content[0] != '\n')
            System.out.print("1. ");
        else
            System.out.println("FIle Empty");
        while (content[i] != '\0') {
            if (content[i] != '\n') {
                System.out.print(content[i]);
            }
            if (content[i] == '\n') {
                System.out.print("\n" + l + ". ");
                l++;
            }
            i = i + 1;

        }

        fr.close();
        System.out.println(" ");
    }
}
```

```
import java.io.*;
import java.util.*;
class freq {
    public static void main(String[] args) throws IOException {
        int i = 0;
        int c = 0, line = 0, words = 0;
        File txt = new File("line.txt");
        FileReader fr = new FileReader(txt);
        char[] content = new char[100];
        fr.read(content);
        if (content[0] != '\n')
            System.out.println("");
```

```
        else
            System.out.println("FIle Empty");
        while (content[i] != '\0') {
            if (Character.isWhitespace(content[i])) {
                words++;
            }
            if (content[i] == '\n' && content[i + 1] != '\n') {
                line++;
            }
            if (content[i] != ' ' && content[i] != '\n' && content[i] != '\0')
{
                c++;
            }
            i = i + 1;
        }
        fr.close();
        System.out.println("Characters = " + c + "\nWords = " + (words + 1) +
"\nlines = " + (line + 1));
        System.out.println(" ");
    }
}
```

```
import java.io.*;
import java.util.*;
class Rdwrt{
    public static void main(String[] args) throws IOException
    {
        int i=0,l=2;
        File txt = new File("red.txt");
                FileReader fr = new FileReader(txt);
                char[] content = new char[100];
                fr.read(content);
                if(content[0]!='\n') System.out.print(" ");
                else System.out.println("FIle Empty");
                FileWriter fw = new FileWriter("wrt.txt");
                while(content[i]!='\0'){
                    fw.write(content[i]);
                    i++;
                }
            fr.close();
            fw.close();
            System.out.println("Completed!");
    }
}

import java.util.*;
public class tknizer {
    public static void main(String args[])
```

```java
    {
        int sum=0;
        System.out.print("Enter numbers : ");
        Scanner sc = new Scanner(System.in);
        String Str = sc.nextLine();
        StringTokenizer st = new StringTokenizer(Str);
        while(st.hasMoreTokens())
        {
            String num = st.nextToken();
            int a = Integer.parseInt(num);
            sum=sum+a;
        }
        System.out.println("SUM = " + sum );
        sc.close();
    }
}
```

```
PS D:\Java-Lab\C4> javac threadsync.java
PS D:\Java-Lab\C4> java threadsync
2       4       6       50      100     150
```

```
Enter numbers : 1 2 3 4 5 6
SUM = 21
```

```
Completed!
```

```
Characters = 43
Words = 11
lines = 5
```

```
ODD B/W 1 and 100 :
1       3       5       7       9       11      13      15      17      19      21
  23      25      27      29      31      33      35  37      39      41      43
 45      47      49      51      53      55      57      59      61      63      65
   67      69      71  73      75      77      79      81      83      85      87
  89      91      93      95      97      99
EVEN B/W 1 and 100 :
2       4       6       8       10      12      14      16      18      20      22
  24      26      28      30      32      34      36  38      40      42      44
 46      48      50      52      54      56      58      60      62      64      66
   68      70      72  74      76      78      80      82      84      86      88
  90      92      94      96      98
```

```
Select Choice :
 1.Write on file
 2.Read from file
 3.Exit

1

Enter Text : Hello everyone
writing Completed !!

Select Choice :
 1.Write on file
 2.Read from file
 3.Exit

2

Hello everyone
--END--
```

```
1. hello hello hello
2. helloo
3.
4.
5.
6.
7. hello
8. hellooo
9. hellooooo
```

```java
interface Print {
    void print();
}
interface Show extends Print {
    void show();
}
class Test implements Show {
    public void print() {
        System.out.println("Hello");
    }
    public void show() {
        System.out.println("Welcome");
    }
}
public class Interface1 {
    public static void main(String[] args) {
        Test t = new Test();
        t.print();
        t.show();
    }
}
```

```java
abstract class Shape {
    abstract void noOfSides();
}
class Rectangle extends Shape {
    void noOfSides() {
        System.out.println("No. of sides = 4");
    }
}
class Triangle extends Shape {
    void noOfSides() {
        System.out.println("No. of sides = 3");
    }
}
class Hexagon extends Shape {
    void noOfSides() {
        System.out.println("No. of sides = 6");
    }
}
public class Polymorphism {
    public static void main(String[] args) {
        Hexagon h = new Hexagon();
        Triangle t = new Triangle();
        Rectangle r = new Rectangle();
        h.noOfSides();
        t.noOfSides();
        r.noOfSides();    }}
```

```java
public class TestGC {
    public void finalize() {
        System.out.println("Garbage is collected");
    }
    public static void main(String args[]) {
        TestGC o1 = new TestGC();
        TestGC o2 = new TestGC();
        o1 = null;
        o2 = null;
        System.gc();
    }
}
```

```
Completed!
```

```
No. of sides = 6
No. of sides = 3
No. of sides = 4
```

```
Hello
Welcome
```

```
Garbage is collected
Garbage is collected
```