//Addition of two numbers

import java.util.\*;

class Addition{

    public static void main(String args[]){

        int no1, no2, res;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter first number");

        no1 = sc.nextInt();

        System.out.println("Enter second number");

        no2 = sc.nextInt();

        res = no1 + no2;

        System.out.println("Sum of "+no1+" and "+no2+" is "+res);

    }

}

//Angstrome Number

import java.util.\*;

class Ang{

    public static void main(String[] args) {

        int num;

        try (Scanner in = new Scanner(System.in)) {

            System.out.println("Enter the number");

            num = in.nextInt();

        }

        int d=0, comp = num, temp = num;

        while(num!=0){

            d++;

            num = num / 10;

        }

        int s, sum = 0;

        while(comp!=0){

            s = comp % 10;

            sum = sum + (int)Math.pow(s, d);

            comp = comp / 10;

        }

        if(sum == temp){

            System.out.println("It is a Angstrome");

        }

        else{

            System.out.println("It is not a Angstrome number");

        }

    }

}

// Angstrom Number in range

import java.util.\*;

class AngRange {

    public static void main(String[] args) {

        int num1, num2;

        try (Scanner sc = new Scanner(System.in)) {

            System.out.println("Enter the first number ::");

            num1 = sc.nextInt();

            System.out.println("Enter the second number ::");

            num2 = sc.nextInt();

        }

        for(int number = num1 + 1; number < num2; ++number) {

            int digits = 0;

            int result = 0;

            int originalNumber = number;

            // number of digits calculation

            while (originalNumber != 0) {

              originalNumber /= 10;

              ++digits;

            }

            originalNumber = number;

            // result contains sum of nth power of its digits

            while (originalNumber != 0) {

              int remainder = originalNumber % 10;

              result += Math.pow(remainder, digits);

              originalNumber /= 10;

            }

            if (result == number) {

              System.out.print(number + " ");

            }

          }

    }

}

// 3. Factorial of a number

import java.util.Scanner;

public class Fact {

    public static void main(String[] args) {

        int number, res=1;

        Scanner in=new Scanner(System.in);

        System.out.println("Enter any number: ");

        number=in.nextInt();

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

            res = res \* i;

        }

        System.out.println(number+"! = "+res);

    }

}

// Fibonacii Series

import java.util.\*;

class Fibo{

    public static void main(String args[]){

        int n;

        try (Scanner in = new Scanner(System.in)) {

            System.out.println("Enter the last number");

            n = in.nextInt();

        }

        System.out.print("0, 1, ");

        int n1 = 0, n2 = 1;

        for(int i = 2; i<=n-1; i++, n1++,n2++){

            int sum = n1 + n2;

            System.out.print(sum+", ");

        }

    }

}

// GCD of 2 numbers

import java.util.\*;

public class GCD {

    public static void main(String[] args) {

        int num1, num2, gcd = 1;

        try (Scanner in = new Scanner(System.in)) {

            System.out.println("Enter first number");

            num1 = in.nextInt();

            System.out.println("Enter second number");

            num2 = in.nextInt();

        }

        int n1 = num1;int n2 = num2;

        while(num1!=num2){

            if(num1>num2){

                num1 = num1 - num2;

            }

            else if(num2>num1){

                num2 = num2-num1;

            }

            else{

                gcd = num1;

            }

        }

        int lcm = (n1\*n2)/gcd;

        System.out.println("GCD of " + n1 + " and " + n2 + " is " + gcd);

        System.out.println("LCM of " + n1 + " and " + n2 + " is " + lcm);

    }

}

// 5. LCM of 2 numbers

import java.util.\*;

public class LCM {

    public static void main(String[] args) {

        int num1, num2, lcm;

        try (Scanner in = new Scanner(System.in)) {

            System.out.println("Enter first number");

            num1 = in.nextInt();

            System.out.println("Enter second number");

            num2 = in.nextInt();

        }

        lcm = (num1 > num2) ? num1 : num2;

        while (true) {

            if (lcm % num1 == 0 && lcm % num2 == 0) {

                System.out.printf("The LCM of %d and %d is %d.", num1, num2, lcm);

                break;

            }

            ++lcm;

        }

    }

}

// 9. Multiplication TableCellEditor

import java.util.\*;

class MultiTable{

    public static void main(String args[]){

        int n;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the last number");

        n = in.nextInt();

        int r;

        System.out.println("");

        System.out.println("Multiplication  Table of 1 is: ");

        for(int j = 1; j<=n; j++){

            for(int i = 1; i<=10; i++){

                r = j \* i;

                System.out.println(j+" X "+i+" = "+r);

            }

            System.out.println("");

            System.out.println("Multiplication  Table of "+(j+1)+"is: ");

        }

    }

}

// partern 1

import java.util.\*;

class P1{

    public static void main(String args[]){

        int n;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the row size");

        n = in.nextInt();

        for(int i = 1; i<=n; i++){

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

                System.out.print("\* ");

            }

            System.out.println("");

        }

    }

}

// partern P2

import java.util.\*;

class P2{

    public static void main(String args[]){

        int n;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the row size");

        n = in.nextInt();

        for(int i = 1; i<=n; i++){

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

                System.out.print(j);

            }

            System.out.println("");

        }

    }

}

// partern P3

import java.util.\*;

class P3{

    public static void main(String args[]){

        int n;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the row size");

        n = in.nextInt();

        for(int i = 1; i<=n; i++){

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

                System.out.print(i);

            }

            System.out.println("");

        }

    }

}

// partern P4

import java.util.\*;

class P4 {

    public static void main(String args[]) {

        int k = 1;

        for (int r = 1; r <= 5; r++) {

            for (int c = 1; c <= r; c++) {

                System.out.print(k+" ");

                k++;

            }

            System.out.println();

        }

    }

}

// Partern 5

import java.util.Scanner;

public class P5 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter n: ");

        int n = sc.nextInt();

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

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

                if ((i + j) % 2 == 0) {

                    System.out.print("1 ");

                } else {

                    System.out.print("0 ");

                }

            }

            System.out.println();

        }

    }

}

// import java.util.\*;

// Pattern 6

public class P6{

    public static void main(String[] args){

        int n = 5;

        // Scanner in = new Scanner(System.in);

        int i, j, k, l;

        //Syste.out.println("Enter the number of row");

        for(i=1;i<=n;i++) {

            for(j=i;j<=n-1;j++) {

                System.out.print(" ");

                System.out.print(" ");

            }

            for(j=1;j<=i;j++) {

                System.out.print(j);

                System.out.print(" ");

            }

            for(j=i-1;j>=1;j--) {

                System.out.print(j);

                System.out.print(" ");

            }

            System.out.println();

        }

        k=1;

        for(i=4;i>=n-k && k<=4 ;i--,k++) {

            for(j=i;j<=n-1;j++) {

                System.out.print(" ");

                System.out.print(" ");

            }

            for(j=1;j<=i;j++) {

                System.out.print(j);

                System.out.print(" ");

            }

            for(j=i-1;j>=1;j--) {

                System.out.print(j);

                System.out.print(" ");

            }

            System.out.println();

        }

        // int i,j,x=5,l=0,k;

        // for(i=1;i<10;i++)

        // {

        //  k=1;

        //  for(j=1;j<10;j++)

        //  {

        //      if(j<x-l || j>x+l)

        //      {

        //          System.out.print("  ");

        //      }

        //      else

        //      {

        //          System.out.print(k+" ");

        //          if(j<5)

        //          {

        //              k++;

        //          }

        //          else

        //          {

        //              k--;

        //          }

        //      }

        //  }

        //  if(i<5)

        //  {

        //      l++;

        //  }

        //  else

        //  {

        //      l--;

        //  }

        //  System.out.println();

        // }

    }

}

// Pattern 7

class P7{

    public static void main(String[] args){

        int i, j, k=1, n = 5, s=0;

        for(i=1; i<=n; i++){

            if(i%2!=0){

                for(j=1; j<=i;j++,k++){

                    s = k;

                }

                for(j=1; j<=i;j++, s--){

                    System.out.print(s);

                    System.out.print(" ");

                }

            }

            else{

                for(j=1; j<=i;j++,k++){

                    System.out.print(k);

                    System.out.print(" ");

                }

            }

            System.out.println();

        }

    }

}

class P8{

    public static void main(String[] args){

        int n = 101;

        //Scanner in = new Scanner(System.in);

        int i, j, k, l;

        //Syste.out.println("Enter the number of row");

        for(i=1, k=97;i<=5;i++,k++) {

            for(j=i;j<=n-1;j++) {

                System.out.print(" ");

                System.out.print(" ");

            }

            for(j=97;j<=k;j++) {

                System.out.print((char)j);

                System.out.print(" ");

            }

            for(j=k-1;j>=97;j--) {

                System.out.print((char)j);

                System.out.print(" ");

            }

            System.out.println();

        }

    }

}

// 1. Check Palindrum or not

import java.util.Scanner;

public class Pali {

    public static void main(String args[])

    {

        int x,number, y,temp=0;

        Scanner in=new Scanner(System.in);

        System.out.println("Enter any number: ");

        number=in.nextInt();

        y = number;

        while(number!=0)

        {

            x=number%10;

            temp=temp\*10+x;

            number=number/10;

        }

        if(temp==y)

        {

            System.out.println("Number is Palindrome");

        }

        else

        {

            System.out.println("not Palindrome");

        }

    }

}

// 2. Prime number check

import java.util.Scanner;

public class Prime {

  public static void main(String[] args) {

    int num;

    try (Scanner in = new Scanner(System.in)) {

      System.out.println("Enter the number");

      num = in.nextInt();

    }

    boolean flag = false;

    for (int i = 2; i <= num / 2; ++i) {

      // condition for nonprime number

      if (num % i == 0) {

        flag = true;

        break;

      }

    }

    if (!flag)

      System.out.println(num + " is a prime number.");

    else

      System.out.println(num + " is not a prime number.");

  }

}

// 6. Prime number between a range

import java.util.Scanner;

public class PrimeRange {

    public static void main(String[] args) {

        int num1, num2;

        try (Scanner in = new Scanner(System.in)) {

            System.out.println("Enter first number");

            num1 = in.nextInt();

            System.out.println("Enter second number");

            num2 = in.nextInt();

        }

        while (num1 < num2) {

            boolean flag = false;

            for(int i = 2; i <= num1/2; ++i) {

                // condition for nonprime number

                if(num1 % i == 0) {

                    flag = true;

                    break;

                }

            }

            if (!flag && num1 != 0 && num1 != 1)

                System.out.print(num1 + " ");

            ++num1;

        }

    }

}

//iNTERFACE

import java.util.\*;

interface Shape{

    void input();

    public abstract void area();

    abstract void show();

}

class Square implements Shape{

    int side, result;

    public void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the side of the square");

        side = in.nextInt();

    }

    public void area(){

        result = side \* side;

    }

    public void show(){

        System.out.println("Area of square is "+result);

    }

}

class DemoInterface{

    public static void main(String args[]){

        Shape ob;

        ob = new Square();

        ob.input();

        ob.area();

        ob.show();

    }

}

/\*\*

Design a class named Rectangle to represent a rectangle. The class contains:

a) Two double data fields named width and height that specify the width and height of the

rectangle. The default values are 1 for both width and height .

b) A no-arg constructor that creates a default rectangle.

c) A constructor that creates a rectangle with the specified width and height .

d) A method named getArea() that returns the area of this rectangle.

e) A method named getPerimeter() that returns the perimeter

Author: Priyanshu Mallick

\*\*/

import java.util.\*;

class Rectangle{

    double width;

    double height;

    /\*Rectangle(){

        width = 1;

        height = 1;

    }

    Rectangle(double w, double h){

        width = w;

        height = h;

    }\*/

    void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the height of the rectangle");

        height = in.nextDouble();

        System.out.println("Enter the width of the rectangle");

        width = in.nextDouble();

    }

    double getArea(){

        return width\*height;

    }

    double getPerimeter(){

        return 2\*(width+height);

    }

}

class Driver{

    public static void main(String args[]){

        Rectangle ob1 = new Rectangle();

        ob1.input();

        double area = ob1.getArea();

        double perimeter = ob1.getPerimeter();

        System.out.println("The area of rectangle is "+area+" m^2 and the perimeter is "+perimeter+" m");

    }

}

/\*\*

Design a class named Rectangle to represent a rectangle. The class contains:

a) Two double data fields named width and height that specify the width and height of the

rectangle. The default values are 1 for both width and height .

b) A no-arg constructor that creates a default rectangle.

c) A constructor that creates a rectangle with the specified width and height .

d) A method named getArea() that returns the area of this rectangle.

e) A method named getPerimeter() that returns the perimeter

Author: Priyanshu Mallick

\*\*/

import java.util.\*;

class Rectangle{

    double width;

    double height;

    Rectangle(){

        width = 1;

        height = 1;

    }

    Rectangle(double w, double h){

        width = w;

        height = h;

    }

    /\*void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the height of the rectangle");

        height = in.nextDouble();

        System.out.println("Enter the width of the rectangle");

        width = in.nextDouble();

    }\*/

    double getArea(){

        return width\*height;

    }

    double getPerimeter(){

        return 2\*(width+height);

    }

}

class Driver2{

    public static void main(String args[]){

        Double height, width;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the height of the rectangle");

        height = in.nextDouble();

        System.out.println("Enter the width of the rectangle");

        width = in.nextDouble();

        Rectangle ob1 = new Rectangle();

        Rectangle ob2 = new Rectangle(width, height);

        //ob1.input();

        double Darea = ob1.getArea();

        double Dperimeter = ob1.getPerimeter();

        double area = ob2.getArea();

        double perimeter = ob2.getPerimeter();

        System.out.println("The area of default rectangle is "+Darea+" m^2 and the perimeter is "+Dperimeter+" m");

        System.out.println();

        System.out.println("The area of given rectangle is "+area+" m^2 and the perimeter is "+perimeter+" m");

    }

}

// Complex number additio without constructor

import java.util.\*;

class Complex{

    public int real;

    public int img;

    void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the real part");

        real = in.nextInt();

        System.out.println("Enter the imaginary part");

        img = in.nextInt();

    }

    void add(Complex ob1, Complex ob2){

        real = ob1.real + ob2.real;

        img = ob1.img + ob2.img;

    }

}

class Driver3{

    public static void main(String args[]){

        Complex ob1 = new Complex();

        Complex ob2 = new Complex();

        Complex ob3 = new Complex();

        ob1.input();

        ob2.input();

        ob3.add(ob1, ob2);

        System.out.println("Addition Result: "+ob3.real+" + "+ob3.img+"i");

    }

}

// Complex Number addition with constructor

import java.util.\*;

class Complex{

    public int real;

    public int img;

    /\*void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the real part");

        real = in.nextInt();

        System.out.println("Enter the imaginary part");

        img = in.nextInt();

    }\*/

    Complex(){

        real = 1;

        img = 1;

    }

    Complex(int r, int i){

        real = r;

        img = i;

    }

    void add(Complex ob1, Complex ob2){

        real = ob1.real + ob2.real;

        img = ob1.img + ob2.img;

    }

}

class Driver4{

    public static void main(String args[]){

        /\*Complex ob1 = new Complex();

        Complex ob2 = new Complex();

        Complex ob3 = new Complex();

        ob1.input();

        ob2.input();

        ob3.add(ob1, ob2);\*/

        int real1, real2, img1, img2;

        Complex ob1 = new Complex();

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the real part of num1");

        real1 = in.nextInt();

        System.out.println("Enter the imaginary of num1");

        img1 = in.nextInt();

        System.out.println("Enter the real part of num2");

        real2 = in.nextInt();

        System.out.println("Enter the imaginary of num2");

        img2 = in.nextInt();

        Complex ob2 = new Complex(real1, img1);

        Complex ob3 = new Complex(real2, img2);

        Complex ob4 = new Complex();

        ob4.add(ob2, ob3);

        System.out.println("Addition Result: "+ob4.real+" + "+ob4.img+"i");

    }

}

// Time addition without constructor

import java.util.\*;

class Time{

    public int hour;

    public int minute;

    public int second;

    public int day = 0;

    void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the hour");

        hour = in.nextInt();

        System.out.println("Enter the minute");

        minute = in.nextInt();

        System.out.println("Enter the second");

        second = in.nextInt();

    }

    void add(Time ob1, Time ob2){

        hour = ob1.hour + ob2.hour;

        minute = ob1.minute + ob2.minute;

        second = ob1.second + ob2.second;

        if(second>=60){

            minute = minute + (second/60);

            second = second % 60;

        }

        if(minute>=60){

            hour = hour + (minute/60);

            minute = minute % 60;

        }

        if(hour>=24){

            day = day + (hour/24);

            hour = hour % 24;

        }

    }

}

class Driver5{

    public static void main(String args[]){

        Time ob1 = new Time();

        Time ob2 = new Time();

        Time ob3 = new Time();

        ob1.input();

        ob2.input();

        ob3.add(ob1, ob2);

        System.out.println("Result Time is: "+ob3.day+" D "+ob3.hour+" H "+ob3.minute+" M "+ob3.second+" S");

    }

}

// Time addition with constructor

import java.util.\*;

class Time{

    public int hour;

    public int minute;

    public int second;

    public int day = 0;

    Time(){

    }

    Time(int h, int m, int s){

        hour = h;

        minute = m;

        second = s;

    }

    void add(Time ob1, Time ob2){

        hour = ob1.hour + ob2.hour;

        minute = ob1.minute + ob2.minute;

        second = ob1.second + ob2.second;

        if(second>=60){

            minute = minute + (second/60);

            second = second % 60;

        }

        if(minute>=60){

            hour = hour + (minute/60);

            minute = minute % 60;

        }

        if(hour>=24){

            day = day + (hour/24);

            hour = hour % 24;

        }

    }

}

class Driver6{

    public static void main(String args[]){

        Time ob1 = new Time(12, 20, 20);

        Time ob2 = new Time(22, 35, 63);

        Time ob3 = new Time();

        ob3.add(ob1, ob2);

        System.out.println("Result Time is: "+ob3.day+" D "+ob3.hour+" H "+ob3.minute+" M "+ob3.second+" S");

    }

}

class Employee {

    String empName;

    int empID;

    double empBasicSal;

    Employee() {

    }

    Employee(String n, int i, double bs) {

        empName = n;

        empID = i;

        empBasicSal = bs;

    }

    double grossSalary1() {

        double value = 0;

        return value;

    }

    void empDetails() {

    }

}

class Manager extends Employee {

        double bonus;

        Manager() {

        }

        Manager(String n, int i, double bs) {

            empName = n;

            empID = i;

            empBasicSal = bs;

        }

        double grossSalary2() {

            double value = 0;

            return value;

        }

        void managerDetails() {

            System.out.println(empName);

            System.out.println(empID);

            System.out.println(empBasicSal);

        }

}

public class Driver7 {

    public static void main(String[] args) {

        Manager ob = new Manager("Priyanshu Mallick", 123, 78000.00);

        ob.managerDetails();

    }

}

import java.util.Scanner;

class Employee {

    String empName;

    int empID;

    String designation;

    double grossSalary;

    Employee() {

    }

    Employee(String n, int i, String d, double gs) {

        empName = n;

        empID = i;

        designation = d;

        grossSalary = gs;

    }

    double grossSalary1() {

        double value = 0;

        return value;

    }

    void empDetails() {

        System.out.println(empName);

        System.out.println(empID);

        System.out.println(designation);

        System.out.println(grossSalary);

    }

}

public class Driver8 {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        Employee ob[] = new Employee[50];

        int i;

        for(i = 1; i<=2 ; i++){

            ob[i] = new Employee();

            System.out.println("Taking Details of emp: "+i);

            System.out.print("Name:");

            ob[i].empName = in.next();

            System.out.print("ID:");

            ob[i].empID = in.nextInt();

            System.out.print("Designation:");

            ob[i].designation = in.next();

            System.out.print("Gross Salary:");

            ob[i].grossSalary = in.nextDouble();

            System.out.println();

        }

        for(i = 1; i<=2 ; i++){

            System.out.println("Details of emp: "+i);

            ob[i].empDetails();

            System.out.println();

        }

    }

}

// Abstarct Class

import java.util.\*;

abstract class Shape{

    double side1, side2, side3;

    abstract void input();

    abstract void area();

    void show(){

        System.out.println("Base class method");

    }

}

class Rectangle extends Shape{

    double value;

    void input(){

        System.out.println("FOR RECTANGLE");

        System.out.println();

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the height");

        side1 = in.nextDouble();

        System.out.println("Base class width");

        side2 = in.nextDouble();

    }

    void area(){

        value = side1\*side2;

    }

    void show(){

        System.out.println("Area of rectangle is: "+value);

        System.out.println();

    }

}

class Square extends Shape{

    double value;

    void input(){

        System.out.println("FOR SQUARE");

        System.out.println();

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the length of the side");

        side1 = in.nextDouble();

    }

    void area(){

        value = side1 \* side1;

    }

    void show(){

        System.out.println("The area of square is "+value);

        System.out.println();

    }

}

class Triangle extends Shape{

    double value;

    void input(){

        System.out.println("FOR TRIANGLE");

        System.out.println();

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the three sides of the triangle");

        side1 = in.nextDouble();

        side2 = in.nextDouble();

        side3 = in.nextDouble();

    }

    void area(){

        double s;

        s = (side1+side2+side3)/2;

        value = Math.sqrt(s\*(s-side1)\*(s-side2)\*(s-side3));

    }

    void show(){

        System.out.println("The area of triangle is "+value);

        System.out.println();

    }

}

class Driver9{

    public static void main(String args[]){

        Shape ob;

        ob = new Square();

        ob.input();

        ob.area();

        ob.show();

        ob = new Rectangle();

        ob.input();

        ob.area();

        ob.show();

        ob = new Triangle();

        ob.input();

        ob.area();

        ob.show();

    }

}

abstract class Employee {

    String Name;

    int number;

    Employee(String n, int i) {

        Name = n;

        number = i;

    }

    abstract void printData();

}

class Manager extends Employee {

    String title, name;

    double clubDues;

    Manager(String t, double cd, String n, int i){

        super(n, i);

        title = t;

        clubDues = cd;

    }

    void printData(){

        System.out.println(Name+" "+number+" "+title+" "+clubDues);

    }

}

class Scientist extends Employee {

    String publication, name;

    Scientist(String p, String n, int i){

        super(n, i);

        publication = p;

    }

    void printData(){

        System.out.println(Name+" "+number+" "+publication);

    }

}

public class Driver10 {

    public static void main(String[] args) {

        Employee ob;

        ob = new Manager("Mr",2345,"Priyanshu",45679);

        ob.printData();

        ob = new Scientist("KP","Chandan",23456);

        ob.printData();

    }

}

abstract class Employee {

    String name;

    int empID;

    Employee(String n, int i) {

        name = n;

        empID = i;

    }

    abstract void earning();

    abstract void printData();

}

class Boss extends Employee {

    double weeklySalary;

    Boss(double ws, String n, int i){

        super(n, i);

        weeklySalary = ws;

    }

    void printData(){

        System.out.println(name+" "+empID+" "+weeklySalary);

    }

    @Override

    void earning() {

    }

}

class HourlyWorker extends Employee {

    int wage, hour;

    HourlyWorker(int w, int h, String n, int i){

        super(n, i);

        wage = w;

        hour = h;

    }

    void printData(){

        System.out.println(name+" "+empID+" "+wage+" "+hour);

    }

    @Override

    void earning() {

        // TODO Auto-generated method stub

    }

}

class CommissionWorker extends Employee {

    double salary;

    int comission;

    int quanity;

    CommissionWorker(double s, int c, int q, String n, int i){

        super(n, i);

        salary = s;

        comission = c;

        quanity = q;

    }

    void printData(){

        System.out.println(name+" "+empID+" "+salary+" "+comission+" "+quanity);

    }

    @Override

    void earning() {

        // TODO Auto-generated method stub

    }

}

class PieceWorker extends Employee {

    int wagePerPiece;

    int quanity;

    PieceWorker(int w, int q, String n, int i){

        super(n, i);

        wagePerPiece = w;

        quanity = q;

    }

    void printData(){

        System.out.println(name+" "+empID+" "+wagePerPiece+" "+quanity);

    }

    @Override

    void earning() {

        // TODO Auto-generated method stub

    }

}

public class Driver11 {

    public static void main(String[] args) {

        Employee ob;

        ob = new Boss(1200, "Badal", 1);

        ob.printData();

        ob = new CommissionWorker(20000, 2000, 20, "Chandan", 2);

        ob.printData();

        ob = new HourlyWorker(10, 6, "Kousik", 3);

        ob.printData();

        ob = new PieceWorker(30, 3, "Priyanshu", 4);

        ob.printData();

    }

}

// Hierarchical Inheritance

class Base{

    void show(){

        System.out.println("Base class");

    }

}

class Derived1 extends Base{

    void display(){

        System.out.println("Derived1 class");

    }

}

class Derived2 extends Base{

    void print(){

        System.out.println("Derived2 class");

    }

}

class HierarchicalInheritance{

    public static void main(String args[]){

        Derived1 ob1 = new Derived1();

        Derived2 ob2 = new Derived2();

        ob1.show();

        ob1.display();

        ob2.show();

        ob2.print();

    }

}

// Hierarchical Example Inheritance

import java.util.\*;

class Base{

    int no1;

}

class Derived1 extends Base{

    int no2, c;

    void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the first num");

        no1 = in.nextInt();

        System.out.println("Enter the 2nd num");

        no2 = in.nextInt();

    }

    void findLarge(){

        if(no1>no2){

            c = no1;

        }

        else if(no1<no2){

            c = no2;

        }

    }

    void display(){

        System.out.println(c+" is the larger");

    }

}

class Derived2 extends Base{

    int no3, s;

    void input(){

        Scanner in = new Scanner(System.in);

        if(no1==0){

            System.out.println("Enter the first num");

            no1 = in.nextInt();

        }

        System.out.println("Enter the 3rd num");

        no3 = in.nextInt();

    }

    void findSmall(){

        if(no1>no3){

            s = no3;

        }

        else if(no1<no3){

            s = no1;

        }

    }

    void display(){

        System.out.println(s+" is the smaller");

    }

}

class HierarchicalInheritanceEx{

    public static void main(String args[]){

        Derived1 ob1 = new Derived1();

        Derived2 ob2 = new Derived2();

        ob1.input();

        ob1.findLarge();

        ob1.display();

        ob2.input();

        ob2.findSmall();

        ob2.display();

    }

}

// Multilevel Inheritance

class Base{

    void show(){

        System.out.println("Base class");

    }

}

class Derived1 extends Base{

    void display(){

        System.out.println("Derived1 class");

    }

}

class Derived2 extends Derived1{

    void print(){

        System.out.println("Derived2 class");

    }

}

class MultiLevelInheritance{

    public static void main(String args[]){

        Derived2 ob = new Derived2();

        ob.show();

        ob.display();

        ob.print();

    }

}

// Multilevel Inheritance Example

import java.util.\*;

class Base{

    int no1;

}

class Derived1 extends Base{

    int no2;

}

class Derived2 extends Derived1{

    int no3, c, s;

    void input(){

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the first num");

        no1 = in.nextInt();

        System.out.println("Enter the 2nd num");

        no2 = in.nextInt();

        System.out.println("Enter the 3rd num");

        no3 = in.nextInt();

    }

    void findSmall(){

        if(no1<no2){

            c = no1;

        }

        else if(no1>no2){

            c = no2;

        }

        if(c<no3){

            s = c;

        }

        else if(c>no3){

            s = no3;

        }

    }

    void display(){

        System.out.println(s+" is the smallest");

    }

}

class MultiLevelInheritanceEx{

    public static void main(String args[]){

        Derived2 ob = new Derived2();

        ob.input();

        ob.findSmall();

        ob.display();

    }

}

// Single Inheritance

class Base{

    void show(){

        System.out.println("Base class");

    }

}

class Derived extends Base{

    void display(){

        System.out.println("Base class");

    }

}

class SingleInheritance{

    public static void main(String args[]){

        Derived ob = new Derived();

        ob.show();

        ob.display();

    }

}

// Circular Queue in Java

import java.util.\*;

class CiQu{

    public int SIZE = 10;

    public int q[]=new int[SIZE];

    public int front = -1;

    public int rear = -1;

    public boolean isFull(){

        if((rear == SIZE-1) && front ==0 ||(front==rear+1)){

            return true;

        }

        else{

            return false;

        }

    }

    public boolean isEmpty(){

        if(front == -1){

            return true;

        }

        else{

            return false;

        }

    }

    public void insert(int x){

        if(isFull()){

            System.out.println("Queue is full");

            System.exit(1);

        }

        else{

            if(rear == -1){

                front++;

                rear++;

            }

            else if(rear==size-1)

            {

                rear=0;

            }

            else

            {

                rear++;

            }

            q[rear] = x;

            System.out.println("The inserted element is: "+q[rear]);

        }

    }

    public int delete(){

        int de=0;

        if(isEmpty()){

            System.out.println("Queue is empty");

            System.exit(1);

        }

        else{

            de = q[front];

            if(rear==front)

            {

                front = -1;

                rear = -1;

            }

            else if(front == size-1)

            {

                front = 0;

            }

            else

            {

                front++;

            }

        }

        return de;

    }

    public void display(){

        if(isEmpty()){

            System.out.println("Queue is empty");

        }

        System.out.println("Queue is");

        if(front <=rear)

        {

            for(i=front;i<=rear;i++)

            {

                System.out.println(q[i]+", ");

            }

        }

        else

        {

            for(i=front;i<SIZE;i++)

            {

                System.out.println(q[i]+", ");

            }

            for(i=0;i<=rear;i++)

            {

                System.out.println(q[i]+", ");

            }

        }

    }

}

class CirQueueImp{

    public static void main(String args[]){

        int v, d;

        Scanner in = new Scanner(System.in);

        CiQu ob = new CiQu();

        int ch;

        while(true){

            System.out.println();

            System.out.println("Choose from following options");

            System.out.println("1. Insert");

            System.out.println("2. Delete");

            System.out.println("3. Display");

            ch = in.nextInt();

            switch(ch){

                case 1: System.out.println("Enter the value");

                        v = in.nextInt();

                        ob.insert(v);

                        break;

                case 2: d = ob.delete();

                        System.out.println("The poped element is: "+d);

                        break;

                case 3: ob.display();

                        break;

                default: System.out.println("Invalid Input");

            }

        }

    }

}

//Queue Implimenting using java

//Stack using java

import java.util.\*;

class Qu{

    public int SIZE = 10;

    public int q[]=new int[SIZE];

    public int front = -1;

    public int rear = -1;

    public boolean isFull(){

        if(rear==SIZE-1){

            return true;

        }

        else{

            return false;

        }

    }

    public boolean isEmpty(){

        if(front==-1 && front == rear +1){

            return true;

        }

        else{

            return false;

        }

    }

    public void insert(int x){

        if(isFull()){

            System.out.println("Queue is full");

            System.exit(1);

        }

        else{

            if(front == -1){

                front = 0;

            }

            rear = rear + 1;

            q[rear] = x;

            System.out.println("The inserted element is: "+q[rear]);

        }

    }

    public int delete(){

        int de=0;

        if(isEmpty()){

            System.out.println("Queue is empty");

            System.exit(1);

        }

        else{

            de = q[front];

            front = front + 1;

        }

        return de;

    }

    public void display(){

        if(isEmpty()){

            System.out.println("Queue is empty");

        }

        System.out.println("Queue is");

        for(int i=front;i<=rear;i++){

            System.out.print(q[i]+", ");

        }

        System.out.println();

    }

}

class QueueImp{

    public static void main(String args[]){

        int v, d;

        Scanner in = new Scanner(System.in);

        Qu ob = new Qu();

        int ch;

        while(true){

            System.out.println();

            System.out.println("Choose from following options");

            System.out.println("1. Insert");

            System.out.println("2. Delete");

            System.out.println("3. Display");

            ch = in.nextInt();

            switch(ch){

                case 1: System.out.println("Enter the value");

                        v = in.nextInt();

                        ob.insert(v);

                        break;

                case 2: d = ob.delete();

                        System.out.println("The poped element is: "+d);

                        break;

                case 3: ob.display();

                        break;

                default: System.out.println("Invalid Input");

            }

        }

    }

}

//Stack using java

import java.util.\*;

class Stk{

    public int SIZE = 10;

    public int stack[]=new int[SIZE];

    public int top = -1;

    public boolean isFull(){

        if(top==SIZE-1){

            return true;

        }

        else{

            return false;

        }

    }

    public boolean isEmpty(){

        if(top==-1){

            return true;

        }

        else{

            return false;

        }

    }

    public void push(int x){

        if(isFull()){

            System.out.println("Stack is full");

            System.exit(1);

        }

        else{

            top = top + 1;

            stack[top] = x;

            System.out.println("The inserted element is: "+stack[top]);

        }

    }

    public int pop(){

        int de=0;

        if(isEmpty()){

            System.out.println("Stack is empty");

            System.exit(1);

        }

        else{

            de = stack[top];

            top = top - 1;

        }

        return de;

    }

    public void display(){

        for(int i = 0; i<=top;i++){

            System.out.print(stack[i]+", ");

            //System.out.println();

        }

    }

}

class StackImp{

    public static void main(String args[]){

        int v, d;

        Scanner in = new Scanner(System.in);

        Stk ob = new Stk();

        int ch;

        while(true){

            System.out.println();

            System.out.println("Choose from following options");

            System.out.println("1. Push");

            System.out.println("2. Pop");

            System.out.println("3. Display");

            ch = in.nextInt();

            switch(ch){

                case 1: System.out.println("Enter the value");

                        v = in.nextInt();

                        ob.push(v);

                        break;

                case 2: d = ob.pop();

                        System.out.println("The poped element is: "+d);

                        break;

                case 3: ob.display();

                        break;

                default: System.out.println("Invalid Input");

            }

        }

    }

}

//Banking application with user define exception

import java.util.\*;

class Bk extends Exception{

    String str;

    Bk(String str){

        this.str = str;

    }

    public String toString(){

        return str;

    }

}

class Bank{

    float bal;

    Bank(){

    }

    Bank(float b){

        bal = b;

    }

    void display(){

        System.out.println("Balance: "+bal);

    }

    void dipo(float am){

        bal = bal + am;

    }

    void withd(float am) throws Bk{

        if(bal>am){

            bal = bal - am;

        }

        else{

            throw new Bk("Insufficient Balane in account");

        }

    }

}

class TestB{

    public static void main(String args[]){

        Bank b = new Bank(2000.00f);

        b.dipo(1000.00f);

        b.display();

        try{

            b.withd(1000.00f);

        }

        catch(Bk ob){

            System.out.println(ob);

        }

        b.display();

    }

}

// Exception implimentation

import java.util.\*;

class TestE1{

    public static void main(String args[]){

        int no1, no2;

        System.out.println("Enter numbers");

        Scanner in = new Scanner(System.in);

        try{

            no1 = in.nextInt();

            no2 = in.nextInt();

            int res = no1/no2;

            System.out.println("Result "+res);

        }

        catch(ArithmeticException ob){

            System.out.println("Cannt be devided by zero");

        }

        System.out.println("Over");

        System.out.println("Thank You");

    }

}

// Exception implimentation

import java.util.\*;

class TestE2{

    public static void main(String args[]){

        int no1, no2;

        System.out.println("Enter numbers");

        Scanner in = new Scanner(System.in);

        try{

            no1 = in.nextInt();

            no2 = in.nextInt();

            int res = no1/no2;

            System.out.println("Result "+res);

        }

        catch(ArithmeticException ob){

            System.out.println("Cannt be devided by zero");

        }

        catch(NullPointerException ob){

            System.out.println("Null found");

        }

        catch(ArrayIndexOutOfBoundsException ob){

            System.out.println("Array size ovrflow");

        }

        System.out.println("Over");

        System.out.println("Thank You");

    }

}

// Exception implimentation

import java.util.\*;

class TestE3{

    public static void main(String args[]){

        int no1, no2;

        int res;

        System.out.println("Enter numbers");

        Scanner in = new Scanner(System.in);

        try{

            no1 = in.nextInt();

            no2 = in.nextInt();

            if(no2==0){

                throw new ArithmeticException();

            }

            else{

                res = no1/no2;

                System.out.println("Result "+res);

            }

        }

        catch(ArithmeticException ob){

            System.out.println("Cannt be devided by zero");

        }

        System.out.println("Over");

        System.out.println("Thank You");

    }

}

// Exception implimentation

import java.util.\*;

class TestE4{

    public static void main(String args[]){

        int no1, no2;

        int res;

        System.out.println("Enter numbers");

        Scanner in = new Scanner(System.in);

        try{

            no1 = in.nextInt();

            no2 = in.nextInt();

            if(no2==0){

                throw new ArithmeticException();

            }

            else{

                res = no1/no2;

                System.out.println("Result "+res);

            }

        }

        catch(ArithmeticException ob){

            System.out.println("Cannt be devided by zero");

        }

        finally{

            System.out.println("Finally");

        }

        System.out.println("Over");

        System.out.println("Thank You");

    }

}

// Use of throws key word in java

class TestE5{

    void m3(){

        int no1 = 10;

        int res = no1/0;

    }

    void m2() throws ArithmeticException{

        m3();

    }

    void m1(){

        try{

            m2();

        }

        catch(ArithmeticException ob){

            System.out.println(ob);

        }

    }

    public static void main(String args[]){

        TestE5 ob = new TestE5();

        ob.m1();

    }

}

// User Defined Exception

import java.util.\*;

class VotingApplication extends Exception{

    String str;

    VotingApplication(String str){

        this.str = str;

    }

    public String toString(){

        return str;

    }

}

class TestE6{

    static void validate(int age) throws VotingApplication{

        if(age>=18){

            System.out.println("Welcome to Voting");

        }

        else{

            throw new VotingApplication("Not Allowed");

        }

    }

    public static void main(String args[]){

        int age;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter your age");

        age = in.nextInt();

        try{

            validate(age);

        }

        catch(VotingApplication ob){

            System.out.println("As age is less than 18 "+ob);

        }

    }

}