Java practical

Q1

import java.awt.\*;

import java.awt.event.\*;

public class MultiplicationTable extends Frame implements ActionListener {

TextField tf;

Button b;

TextArea resultArea;

MultiplicationTable() {

Label label = new Label("Enter a number:");

tf = new TextField();

b = new Button("Generate Table");

resultArea = new TextArea();

label.setBounds(30, 40, 120, 20);

tf.setBounds(160, 40, 80, 20);

b.setBounds(30, 80, 120, 30);

resultArea.setBounds(30, 120, 200, 150);

add(label);

add(tf);

add(b);

add(resultArea);

b.addActionListener(this);

setSize(250, 300);

setLayout(null);

setVisible(true);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == b) {

try {

int number = Integer.parseInt(tf.getText());

generateTable(number);

} catch (NumberFormatException ex) {

resultArea.setText("Invalid input. Please enter a valid number.");

}

}

}

private void generateTable(int number) {

StringBuilder tableResult = new StringBuilder();

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

int result = number \* i;

tableResult.append(number).append(" \* ").append(i).append(" = ").append(result).append("\n");

}

resultArea.setText(tableResult.toString());

}

public static void main(String args[]) {

new MultiplicationTable();

}

}

Q2

import java.util.Scanner;

public class BankAc {

String DepositorName, AcType;

int AcNo;

double AvlBal = 500;

public void AcInfo() {

Scanner s = new Scanner(System.in);

System.out.print("Enter name of depositor: ");

DepositorName = s.nextLine();

System.out.print("Enter account number: ");

AcNo = s.nextInt();

System.out.print("Enter account type (Saving/Current): ");

AcType = s.nextLine();

System.out.println("Depositor Name: " + DepositorName + " " + "Account No: " + AcNo + " " + "Account type: " + AcType);

}

public void Deposite() {

double DepAmt;

Scanner s1 = new Scanner(System.in);

System.out.print("Enter amount to deposit: ");

DepAmt = s1.nextDouble();

AvlBal += DepAmt;

System.out.println("Available balance after depositing amount is: " + AvlBal);

}

public void Withdraw() {

double WithAmt;

Scanner s2 = new Scanner(System.in);

System.out.print("Enter amount to withdraw: ");

WithAmt = s2.nextDouble();

if (AvlBal - WithAmt >= 500) {

AvlBal -= WithAmt;

System.out.println("Withdraw successful");

} else {

System.out.println("Insufficient balance in account");

}

}

public void DisplayBal() {

System.out.println("Account balance: " + AvlBal);

}

public static void main(String args[]) {

BankAc b = new BankAc();

b.AcInfo();

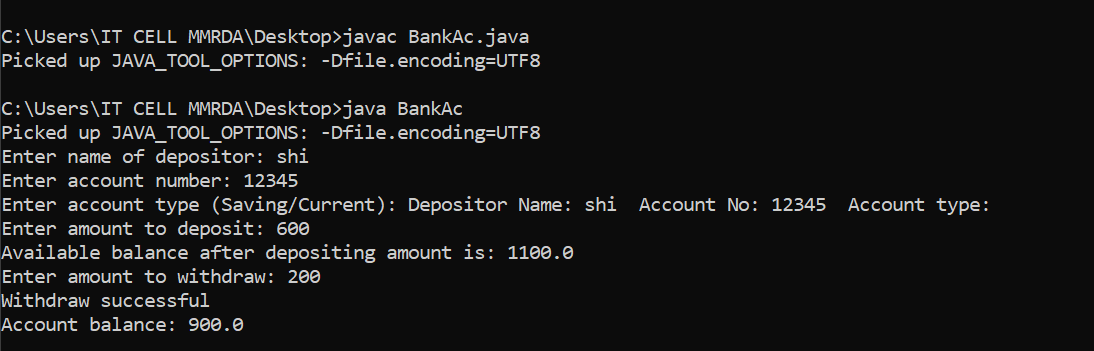
b.Deposite();

b.Withdraw();

b.DisplayBal();

}

}



Q3 triangle

import java.util.\*;

class triangle

{

public static void main (String s1[]){

int base,height;

Scanner s=new Scanner(System.in);

System.out.println("enter a base");

base=s.nextInt();

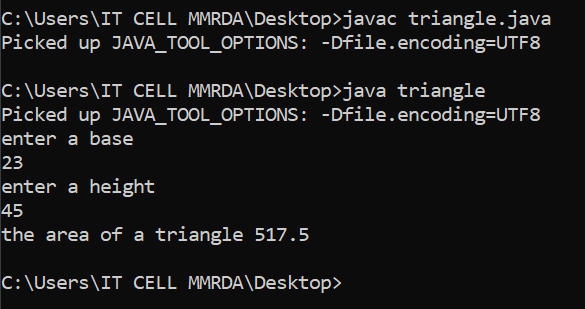
System.out.println("enter a height");

height=s.nextInt();

System.out.println("the area of a triangle " +(0.5\*base\*height));

}

}



Q4factorial

import java.awt.\*;

import java.awt.event.\*;

import java.awt.event.WindowListener;

class Factorial extends Frame implements ActionListener

{

TextField tf;

Button b;

Label n, l, r;

Factorial()

{

n = new Label("AWT Factorial Program");

l = new Label("Enter number");

r = new Label();

tf = new TextField();

b = new Button("Factorial");

n.setBounds(30, 40, 200, 20);

l.setBounds(30, 70, 150, 20);

r.setBounds(30, 170, 200, 20);

tf.setBounds(30, 90, 190, 30);

b.setBounds(30, 130, 190, 30);

b.setBackground(Color.BLUE);

b.setForeground(Color.WHITE);

add(n);

add(l);

add(r);

add(tf);

add(b);

setSize(300, 300);

setLayout(null);

setVisible(true);

b.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

dispose();

}

});

}

public void actionPerformed(ActionEvent e)

{

if (e.getSource() == b)

{

int num = Integer.parseInt(tf.getText());

r.setText("Factorial of " + num + " is " + getFactorial(num));

}

}

public int getFactorial(int x)

{

int a = 1;

for (int i = 1; i <=x; i++)

{

a = a \* i;

}

return (a);

}

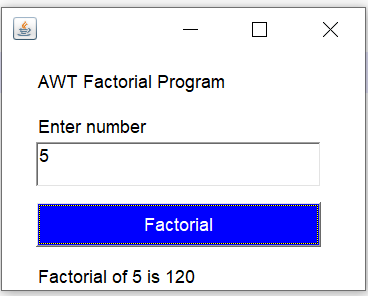
public static void main(String args[])

{

Factorial factorial = new Factorial();

}

}



Q5

Q6even or not

import java.awt.\*;

import java.awt.event.\*;

class Even extends Frame implements ActionListener {

TextField tf;

Button b;

Label n, l, r;

Even() {

n = new Label("AWT Even Odd Program");

l = new Label("Enter number");

r = new Label();

tf = new TextField();

b = new Button("Result");

n.setBounds(30, 40, 200, 20);

l.setBounds(30, 70, 150, 20);

r.setBounds(30, 170, 200, 20);

tf.setBounds(30, 90, 190, 30);

b.setBounds(30, 130, 190, 30);

b.setBackground(Color.BLUE);

b.setForeground(Color.WHITE);

add(n);

add(l);

add(r);

add(tf);

add(b);

setSize(250, 210);

setLayout(null);

setVisible(true);

b.addActionListener(this);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

}

public void getEvenOdd(int x) {

if (x % 2 == 0) {

r.setText("The number " + x + " is even");

} else {

r.setText("The number " + x + " is odd");

}

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == b) {

int x = Integer.parseInt(tf.getText());

getEvenOdd(x);

}

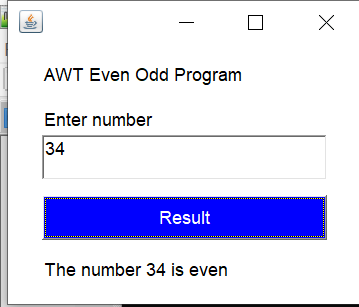
}

public static void main(String[] args) {

Even f = new Even();

}

}



Q7 interface

Done nbk

Q8reverse

import java.awt.\*;

import java.awt.event.\*;

public class Reverse extends Frame implements ActionListener {

TextField tf;

Button b;

Label r;

Reverse() {

Label label = new Label("Enter a number:");

tf = new TextField();

b = new Button("Reverse Number");

r = new Label();

label.setBounds(30, 40, 120, 20);

tf.setBounds(160, 40, 80, 20);

b.setBounds(30, 80, 120, 30);

r.setBounds(30, 120, 200, 20);

add(label);

add(tf);

add(b);

add(r);

b.addActionListener(this);

setSize(250, 180);

setLayout(null);

setVisible(true);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == b) {

int number = Integer.parseInt(tf.getText());

int x = Reverse(number);// to store the reversed value

r.setText("Reversed number: " + x);

}

}

public int Reverse(int number) {

int x = 0;

while (number != 0) {

int digit = number % 10;

x = x \* 10 + digit;

number /= 10;

}

return x;

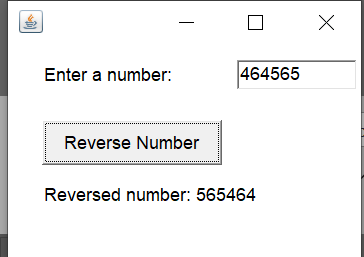
}

public static void main(String args[]) {

Reverse m= new Reverse();

}

}



Q9package

done

Q10mouselistener and mousemotion

import java.awt.\*;

import java.awt.event.\*;

class mouse extends Frame implements MouseListener {

Label l;

mouse() {

l = new Label();

l.setBounds(30, 70, 300, 20); // Increase the width to display longer messages

add(l);

setSize(400, 300); // Increase the width of the frame

setLayout(null);

setVisible(true);

addMouseListener(this);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

}

public void mousePressed(MouseEvent e) {

l.setText("Mouse pressed event generated");

}

public void mouseClicked(MouseEvent e) {

l.setText("Mouse clicked event generated");

}

public void mouseReleased(MouseEvent e) {

l.setText("Mouse released event generated");

}

public void mouseEntered(MouseEvent e) {

l.setText("Mouse entered event generated");

}

public void mouseExited(MouseEvent e) {

l.setText("Mouse exited event generated");

}

public static void main(String args[]) {

mouse al = new mouse();

}

}

Q11salary

import java.util.\*;

class sal

{

public static void main(String args[])

{

double basicSal, netSal,increasedSal;

double increment=0;

Scanner s=new Scanner(System.in);

System.out.println("Enter basic sal");

basicSal=s.nextDouble();

if(basicSal<3000)

{

increment=2;

}

else if(basicSal>=3000&&basicSal<5000)

{

increment=5;

}

else if(basicSal>=5000&&basicSal<10000)

{

increment=10;

}

else if(basicSal>10000){

increment=20;

}

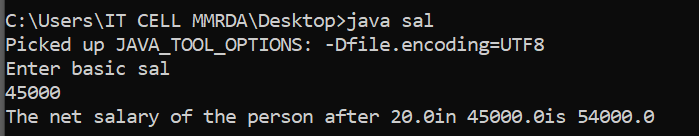
increasedSal=basicSal\*(increment/100);

netSal=(basicSal+increasedSal);

System.out.println("The net salary of the person after " +increment+ "in " +basicSal+ "is "+ netSal);

}

}



Q12prime or not

import java.awt.\*;

import java.awt.event.\*;

class PrimeNumber extends Frame implements ActionListener {

TextField tf;

Button b;

Label n, l, r;

PrimeNumber() {

n = new Label("AWT Prime Number Program");

l = new Label("Enter number");

r = new Label();

tf = new TextField();

b = new Button("Check Prime");

n.setBounds(30, 40, 200,20);

l.setBounds(30, 70, 150, 20);

r.setBounds(30, 170, 200, 20);

tf.setBounds(30, 90, 190, 30);

b.setBounds(30, 130, 190, 30);

b.setBackground(Color.BLUE);

b.setForeground(Color.WHITE);

add(n);

add(l);

add(r);

add(tf);

add(b);

setSize(300, 300);

setLayout(null);

setVisible(true);

b.addActionListener(this);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == b) {

int num = Integer.parseInt(tf.getText());

if (isPrime(num)) {

r.setText(num + " is a Prime Number.");

} else {

r.setText(num + " is not a Prime Number.");

}

}

}

public boolean isPrime(int x) {

if (x <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(x); i++) {

if (x % i == 0) {

return false;

}

}

return true;

}

public static void main(String args[]) {

PrimeNumber primeNumber = new PrimeNumber();

}

}

Q13palindrome

Q14user defined exception

Q15

class player

{

public player()

{

String pname;

int innings,runs,notouts;

}

public void showdata(String pname, int innings, int runs, int notouts)

{

System.out.println("Player name: "+pname);

System.out.println("Innings: "+innings);

System.out.println("Runs scored: "+runs);

System.out.println("Notouts: "+notouts);

}

public void calcAvg(int runs, int innings, int notouts)

{

int BatAvg=runs/(innings-notouts);

System.out.println("Batting average of player is: "+BatAvg);

}

public static void main(String args[])

{

player p=new player();

p.showdata("nidhi",10,100,4);

p.calcAvg(100,10,4);

}

}

Q16 key listener

import java.awt.\*;

import java.awt.event.\*;

import java.awt.event.KeyListener;

import java.awt.event.WindowListener;

class awt extends Frame implements KeyListener

{

Label l;

TextField t;

awt()

{

l=new Label();

l.setBounds(30,70,100,20);

t=new TextField();

t.setBounds(70,100,100,30);

add(t);

add(l);

t.addKeyListener(this);

addWindowListener(new WindowAdapter(){

public void windowClosing (WindowEvent e){

dispose ();

}

});

setSize(300,300);

setLayout(null);

setVisible(true);

}

public void keyPressed (KeyEvent e1)

{

l.setText("keypressed event generated");

}

public void keyTyped (KeyEvent e1)

{

l.setText("key typed event generated ");

}

public void keyReleased (KeyEvent e1)

{

l.setText("key released event generated ");

}

public static void main (String args[]){

awt al =new awt();

}

}



Q17

import java.util.Scanner;

public class CreditCard {

public static void main(String[] args) {

String cardType;

double shopAmt, discount = 0, discountAmt, netPrice;

Scanner s = new Scanner(System.in);

System.out.println("Enter credit card type (VISA, XYZ, ABC): ");

cardType = s.next();

System.out.println("Enter Shopping Amount: ");

shopAmt = s.nextDouble();

switch (cardType) {

case "VISA":

if (shopAmt < 5000) {

discount = 10;

} else {

discount = 20;

}

break;

case "XYZ":

if (shopAmt < 10000) {

discount = 15;

} else {

discount = 25;

}

break;

case "ABC":

if (shopAmt < 800) {

discount = 12;

} else {

discount = 15;

}

break;

default:

System.out.println("Invalid card type");

return;

}

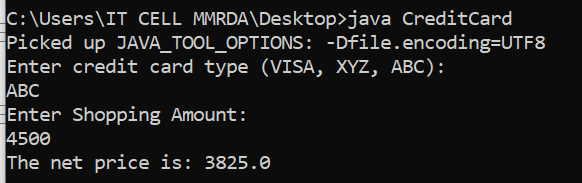
discountAmt = shopAmt \* (discount / 100);

netPrice = shopAmt - discountAmt;

System.out.println("The net price is: " + netPrice);

}

}



Q18arithmetic operator ,Q30,Q31 SAME

import java.awt.\*;

import java.awt.event.\*;

public class ArithmeticDemo extends Frame implements ItemListener {

TextField tf, tf1, rf;

Choice c;

public ArithmeticDemo() {

setLayout(new BorderLayout());

// Create text fields

tf = new TextField();

tf1 = new TextField();

rf = new TextField();

// Create choice for arithmetic operation

c = new Choice();

c.add("+");

c.add("-");

c.add("\*");

c.add("/");

c.add("%");

// Create panels for organizing components

Panel p = new Panel();

Panel rp = new Panel();

// Add components to the panels

p.setLayout(new GridLayout(3, 2));

p.add(new Label("Number 1:"));

p.add(tf);

p.add(new Label("Number 2:"));

p.add(tf1);

p.add(new Label("Operation:"));

p.add(c);

rp.setLayout(new GridLayout(2, 1));

rp.add(new Label("Result:"));

rp.add(rf);

// Add panels to the frame

add(p, BorderLayout.NORTH);

add(rp, BorderLayout.CENTER);

// WindowListener to handle closing event

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

// Register event listeners

c.addItemListener(this);

// Set frame properties

setSize(300, 200);

setVisible(true);

}

private void performOperation() {

try {

int num1 = Integer.parseInt(tf.getText());

int num2 = Integer.parseInt(tf1.getText());

String selectedOperation = c.getSelectedItem();

switch (selectedOperation) {

case "+":

rf.setText(String.valueOf(num1 + num2));

break;

case "-":

rf.setText(String.valueOf(num1 - num2));

break;

case "\*":

rf.setText(String.valueOf(num1 \* num2));

break;

case "/":

rf.setText(String.valueOf((double) num1 / num2));

break;

case "%":

rf.setText(String.valueOf(num1 % num2));

break;

}

} catch (NumberFormatException ex) {

rf.setText("Invalid input");

}

}

public void itemStateChanged(ItemEvent e) {

performOperation();

}

public static void main(String[] args) {

new ArithmeticDemo();

}

}

Q19

Q20 array

public class ArrLarSmall{

public static void main(String[] args) {

int numbers[] = new int[]{55,32,45,98,82,11,9,39,50};

int smallest = numbers[0];

int largest = numbers[0];

for (int i = 1; i < numbers.length; i++) {

if (numbers[i] > largest)

largest = numbers[i];

else if (numbers[i] < smallest)

smallest = numbers[i];

}

System.out.println("Largest Number is : " + largest);

System.out.println("Smallest Number is : " + smallest);

}

}

Q21

Q22cube

import java.awt.\*;

import java.awt.event.\*;

import java.awt.event.WindowListener;

class Cube extends Frame implements ActionListener

{

TextField tf;

Button b;

Label n, l, r;

Cube()

{

n = new Label("AWT Cube Program");

l = new Label("Enter number");

r = new Label();

tf = new TextField();

b = new Button("Result");

n.setBounds(30, 40, 200, 20);

l.setBounds(30, 70, 150, 20);

r.setBounds(30, 170, 200, 20);

tf.setBounds(30, 90, 190, 30);

b.setBounds(30, 130, 190, 30);

b.setBackground(Color.BLUE);

b.setForeground(Color.WHITE);

add(n);

add(l);

add(r);

add(tf);

add(b);

setSize(250,210);

setLayout(null);

setVisible(true);

b.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

dispose();

}

});

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b)

{

int num = Integer.parseInt(tf.getText());

r.setText("Cube of " + num + " is " + (num\*num\*num));

}

}

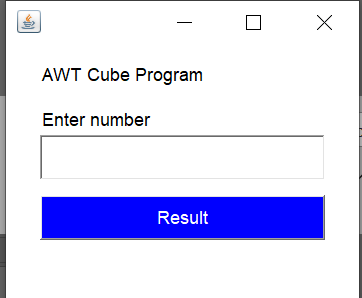
public static void main(String args[])

{

Cube t = new Cube();

}

}



Q23

import java.util.Scanner;

public class calculation {

int a, b;

public void inputNumbers() {

Scanner s = new Scanner(System.in);

System.out.print("Enter first number: ");

a = s.nextInt();

System.out.print("Enter second number: ");

b = s.nextInt();

}

public void sum() {

System.out.println("The sum is " + a + " + " + b + " = " + (a + b));

}

public void difference() {

System.out.println("The difference is " + a + " - " + b + " = " + (a - b));

}

public void multiply() {

System.out.println("The product is " + a + " \* " + b + " = " + (a \* b));

}

public void divide() {

if(b!=0)

{

System.out.println("The division is " + a + " / " + b + " = " + (a / b));

}

else

{

System.out.println("Cannot divide by zero");

}

}

public void average() {

System.out.println("The average of " + a + " + " + b + " = " + ((a + b) / 2));

}

public void maximum() {

System.out.println("The maximum number is :" + Math.max(a, b));

}

public void minimum() {

System.out.println("The minimum number is :" + Math.min(a, b));

}

public static void main(String args[]) {

calculation c = new calculation();

c.inputNumbers();

c.sum();

c.difference();

c.multiply();

c.divide();

c.average();

c.maximum();

c.minimum();

}

}

Q24

import java.awt.\*;

import java.awt.event.\*;

public class ArithmeticDemo extends Frame implements ItemListener {

TextField tf, tf1, rf;

Choice c;

public ArithmeticDemo() {

setLayout(new BorderLayout());

// Create text fields

tf = new TextField();

tf1 = new TextField();

rf = new TextField();

// Create choice for arithmetic operation

c = new Choice();

c.add("+");

c.add("-");

c.add("\*");

c.add("/");

c.add("%");

// Create panels for organizing components

Panel p = new Panel();

Panel rp = new Panel();

// Add components to the panels

p.setLayout(new GridLayout(3, 2));

p.add(new Label("Number 1:"));

p.add(tf);

p.add(new Label("Number 2:"));

p.add(tf1);

p.add(new Label("Operation:"));

p.add(c);

rp.setLayout(new GridLayout(2, 1));

rp.add(new Label("Result:"));

rp.add(rf);

// Add panels to the frame

add(p, BorderLayout.NORTH);

add(rp, BorderLayout.CENTER);

// WindowListener to handle closing event

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

// Register event listeners

c.addItemListener(this);

// Set frame properties

setSize(300, 200);

setVisible(true);

}

private void performOperation() {

try {

int num1 = Integer.parseInt(tf.getText());

int num2 = Integer.parseInt(tf1.getText());

String selectedOperation = c.getSelectedItem();

switch (selectedOperation) {

case "+":

rf.setText(String.valueOf(num1 + num2));

break;

case "-":

rf.setText(String.valueOf(num1 - num2));

break;

case "\*":

rf.setText(String.valueOf(num1 \* num2));

break;

case "/":

rf.setText(String.valueOf((double) num1 / num2));

break;

case "%":

rf.setText(String.valueOf(num1 % num2));

break;

}

} catch (NumberFormatException ex) {

rf.setText("Invalid input");

}

}

public void itemStateChanged(ItemEvent e) {

performOperation();

}

public static void main(String[] args) {

new ArithmeticDemo();

}

}

Q25

Q26

Q27

Q28

Q29 2 pakages primeFact

Q31

Q32

Q34

Q35Areacompute

interface Area{

void Areacompute(float x, float y);

}

class Rectangle implements Area

{

Rectangle(){

System.out.println("Rectangle constructor called");

}

public void Areacompute(float x, float y){

System.out.println(x\*y);

}

}

class circle implements Area{

circle(){

System.out.println("circle constructor called");

}

public void Areacompute(float x, float y){

System.out.println(3.14\*x\*x);

}

}

public class Main{

public static void main(String s1[])

{

Rectangle r = new Rectangle();

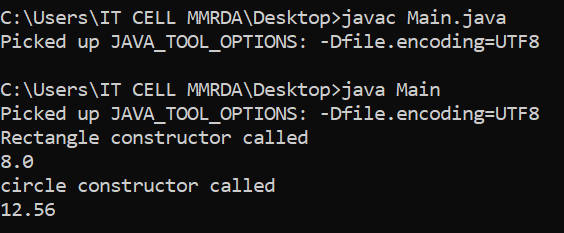
r.Areacompute(2,4);

circle d = new circle();

d.Areacompute(2,0);

}

}



Q36

Q37