**Practical No. 1:**

Write a java program to take input as a command line argument. Your name, course, university rollno and semester. Display the information.

Name:

University Roll No:

Course:

Semester:

**Source Code:**

public class Q1 {

  public static void main(String[] args) {

    String name = args[0];

    int RollNo = Integer.parseInt(args[1]);

    String course = args[2];

    int sem = Integer.parseInt(args[3]);

    System.out.println("Name: " + name);

    System.out.println("University Roll No: " + RollNo);

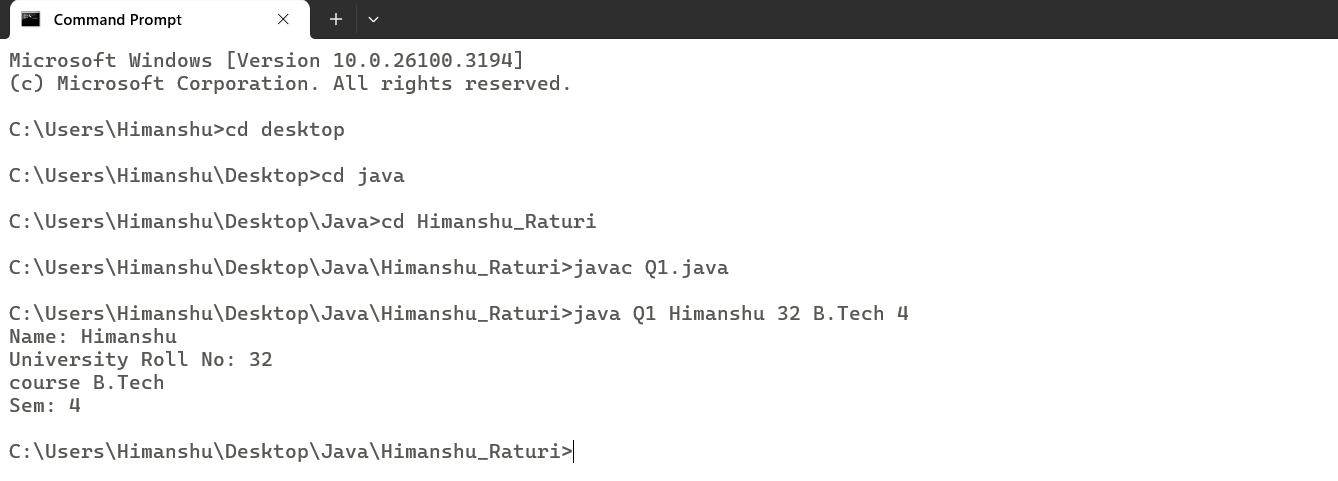
    System.out.println("course " + course);

    System.out.println("Sem: " + sem);

  }

}

**Output:**



**Practical No. 2:**

Using the switch statement, write a menu-driven program to calculate the maturity amount of a bank deposit.

The user is (i) Term Deposit (ii) Recurring Deposit

For option (i) accept Principal (p), rate of interest (r) and time period in years (n). Calculate and output the maturity amount (a) receivable using the formula a = p[1 + r / 100]n.

For option (ii) accept monthly installment (p), rate of interest (r) and time period in months (n). Calculate and output the maturity amount (a) receivable using the formula a = p \* n + p \* n(n + 1) / 2 \* r / 100 \* 1 / 12. For an incorrect option, an appropriate error message should be displayed.

**Source Code:**

import java.util.Scanner;

//import java.lang.\*;

public class Q2

{

    public static void main(String args[])

    {

        System.out.println("The user is\n1.Term Deposit\n2.Recurring Deposit");

        Scanner sc = new Scanner(System.in);

        int Choice = sc.nextInt();

        switch(Choice)

        {

            case 1:

            {

                System.out.println("Enter Principle(P): ");

                int Princ = sc.nextInt();

                System.out.println("Enter rate of interest(r): ");

                float Rate = sc.nextFloat();

                System.out.println("Enter Time Period in years(t): ");

                int t = sc.nextInt();

                double Maturity = Princ\*Math.pow(1+Rate/100,t);

                System.out.println("Maturity Amount is: " + Maturity);

                break;

            }

            case 2:

            {

                System.out.println("Enter Monthly installment(P): ");

                double p = sc.nextInt();

                System.out.println("Enter Rate of Interest(r): ");

                double r = sc.nextInt();

                System.out.println("Enter Time period(yrs): ");

                double t = sc.nextInt();

                double Maturity =  p \* t + p \* t\*(t + 1) / 2 \* r / 100 \* 1 / 12;

                System.out.println("Maturity is: " + Maturity);

                break;

            }

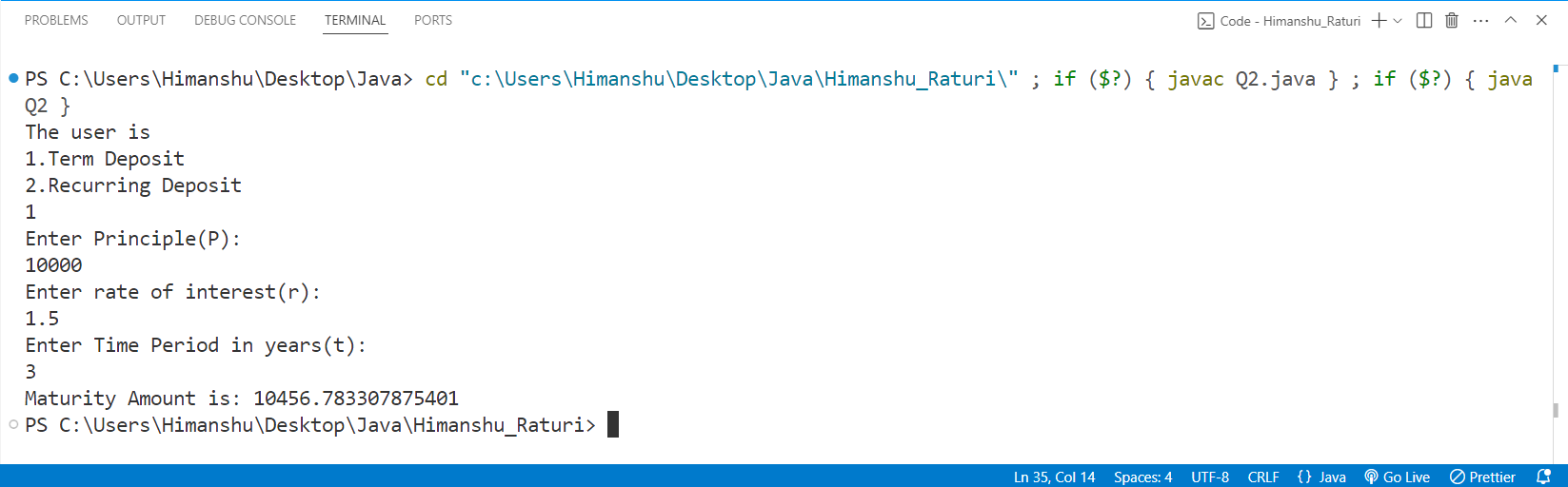
        }

        sc.close();

    }

}

**Output:**



**Practical No. 3:**

Program to find if the given numbers are Friendly pair or not (Amicable or not). Friendly Pair are two or more numbers with a common abundance

**Source Code:**

import java.util.Scanner;

public class Q3 {

  public static void main(String args[]) {

    Scanner sc = new Scanner(System.in);

    System.out.println("Enter Number 1: ");

    int num1 = sc.nextInt();

    System.out.println("Enter Number 2: ");

    int num2 = sc.nextInt();

    int sum1 = 0, sum2 = 0;

    for (int i = 1; i <= num1 / 2; i++)

{

      if (num1 % i == 0)

{

        sum1 += i;

      }

    }

    for (int i = 1; i <= num2 / 2; i++) {

      if (num2 % i == 0) {

        sum2 += i;

      }

    }

    if ((sum1 == num1) && (sum2 == num2)) {

      System.out.println("Abundant Numbers");

    } else {

      System.out.println("Not Abundant Numbers");

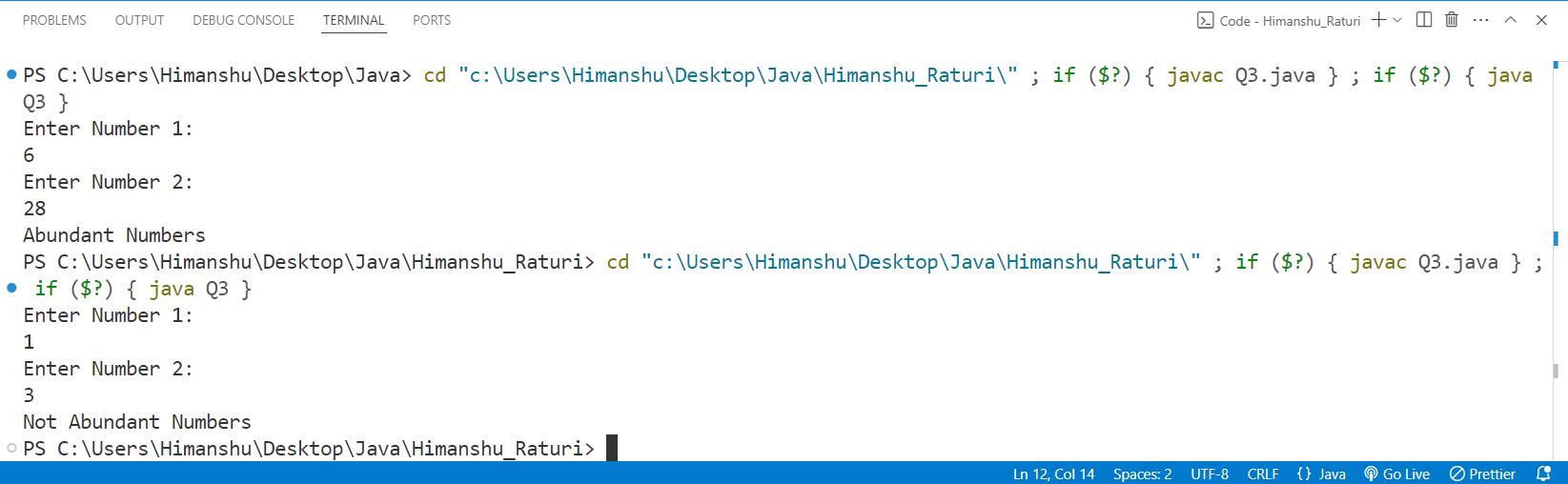
    }

    sc.close();

  }

}

**Output:**



**Practical No. 4:**

Program to replace all 0's with 1 in a given integer. Given an integer as an input, all the 0's in the number has to be replaced with 1.

**Source Code:**

import java.util.Scanner;

public class Q4 {

  public static void main(String args[]) {

    Scanner sc = new Scanner(System.in);

    System.out.print("Input: ");

    int num = sc.nextInt();

    if (num == 0) {

      num = 1;

    }

    int x = 0;

    while (num > 0) {

      int val = num % 10;

      if (val == 0) {

        val = 1;

      }

      x = x \* 10 + val;

      num = num / 10;

    }

    int ans = 0;

    while (x > 0) {

      int val = x % 10;

      ans = ans \* 10 + val;

      x = x / 10;

    }

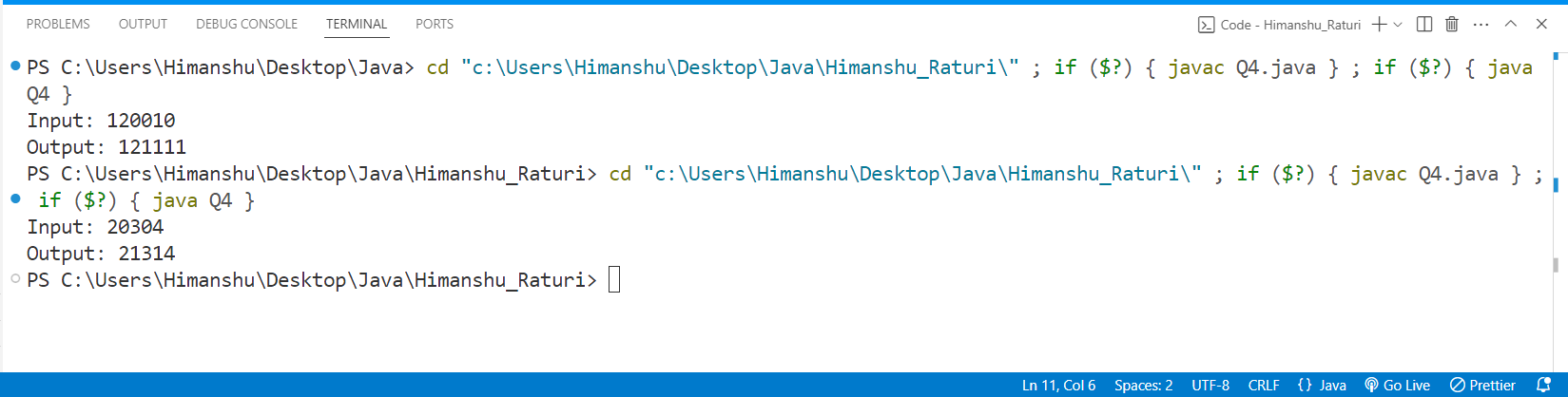
    System.out.print("Output: " + ans);

    sc.close();

  }

}

**Output:**



**Practical No. 5:**

Printing an array into Zigzag fashion. Suppose you were given an array of integers, and you are told to sort the integers in a zigzag pattern. In general, in a zigzag pattern, the first integer is less than the second integer, which is greater than the third integer, which is less than the fourth integer, and so on. Hence, the converted array should be in the form of e1 < e2 > e3 < e4 > e5 < e6.

**Source Code:**

import java.util.Scanner;

public class java5 {

    public static void main(String[] args) {

        {

            Scanner in=new Scanner(System.in);

            int n;

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

            n=in.nextInt();

            int arr[]=new int[n];

            int temp;

            System.out.println("Enter the element in the array ");

            for(int i=0;i<n;i++)

            {

                arr[i]=in.nextInt();

            }

            for(int i=0;i<n-1;i++)

            {

                if(i%2==0)

                {

                    if(arr[i]>arr[i+1])

                    {

                        temp=arr[i];

                        arr[i]=arr[i+1];

                        arr[i+1]=temp;

                    }

                }

                else

                {

                    if(arr[i]<arr[i+1])

                    {

                        temp=arr[i];

                        arr[i]=arr[i+1];

                        arr[i+1]=temp;

                    }

                }

            }

            System.out.println("Array after change ");

            for(int i=0;i<n;i++)

            {

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

            }

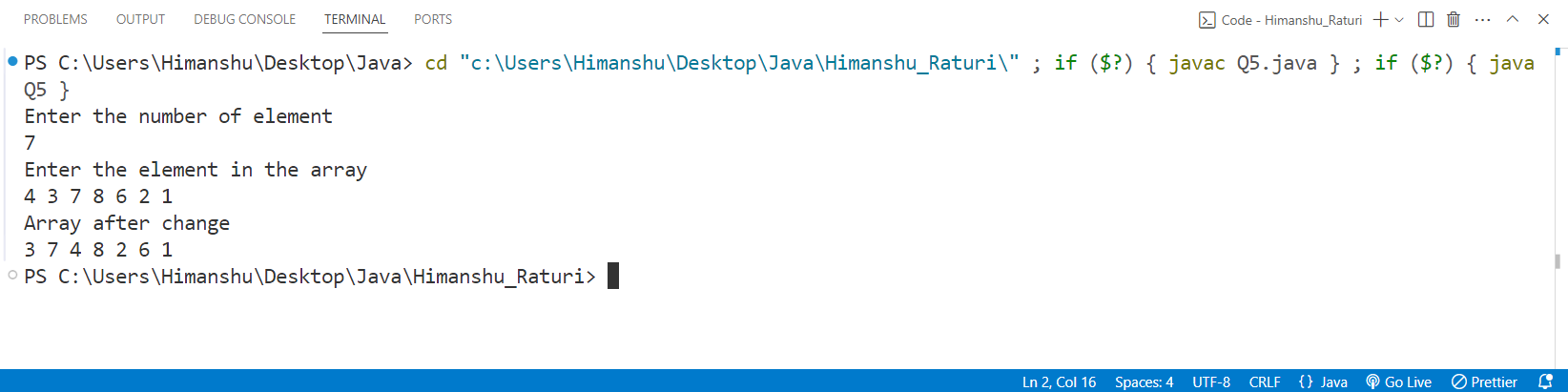
            in.close();

        }

    }

}

**Output:**



**Practical No. 6:** The problem to rearrange positive and negative numbers in an array . Method: This approach moves all negative numbers to the beginning and positive numbers to the end but changes the order of appearance of the elements of the array.

**Source Code:**

package CODES.Java.Himanshu\_Raturi;

import java.util.Scanner;

public class Q6

{

    public static void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.print("Input: ");

        for(int i = 0 ; i < n ; i++)

        {

            arr[i] = sc.nextInt();

        }

        int first = 0;

        for(int i = 0; i < n ; i++)

        {

            if(arr[i] < 0 )

            {

                int temp = arr[i];

                arr[i] = arr[first];

                arr[first] = temp;

                first++;

            }

        }

        System.out.print("Output: ");

        for(int i = 0 ; i < n ; i++)

        {

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

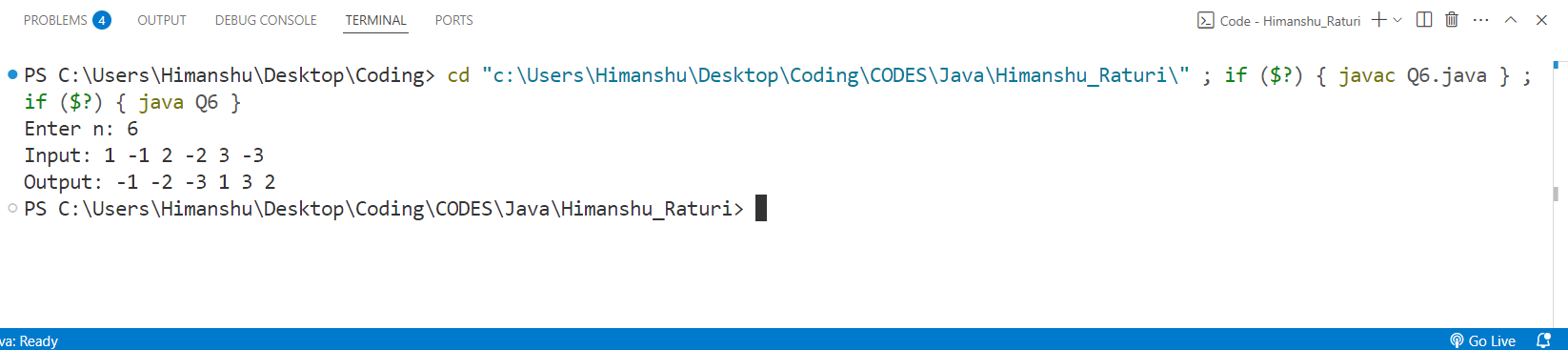
        }

        sc.close();

    }

}

**Output:**



**Practical No. 7:** Program to find the saddle point coordinates in a given matrix. A saddle point is an element of the matrix, which is the minimum element in its row and the maximum in its column.

**Source Code:**

package CODES.Java.Himanshu\_Raturi;

import java.util.Scanner;

public class Q7 {

    public static void main(String args[])

    {

        Scanner sc=new Scanner(System.in);

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

        int rows=sc.nextInt();

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

        int columns=sc.nextInt();

        int arr[][]=new int[rows][columns];

        System.out.println("Enter the elements in the array ");

        for(int i=0;i<rows;i++)

        {

            for(int j=0;j<columns;j++)

            {

                arr[i][j]=sc.nextInt();

            }

        }

        for(int i=0;i<rows;i++)

        {

            int min=arr[i][0];

            int colindex=0;

            for(int j=0;j<columns;j++)

            {

                if(arr[i][j]<min)

                {

                    min=arr[i][j];

                    colindex=j;

                }

            }

            int max=arr[0][colindex];

            for(int k=0;k<rows;k++)

            {

                if(arr[k][colindex]>max)

                {

                    max=arr[k][colindex];

                }

            }

            if(min==max)

            {

                System.out.println("Saddle Point = "+min );

                break;

            }

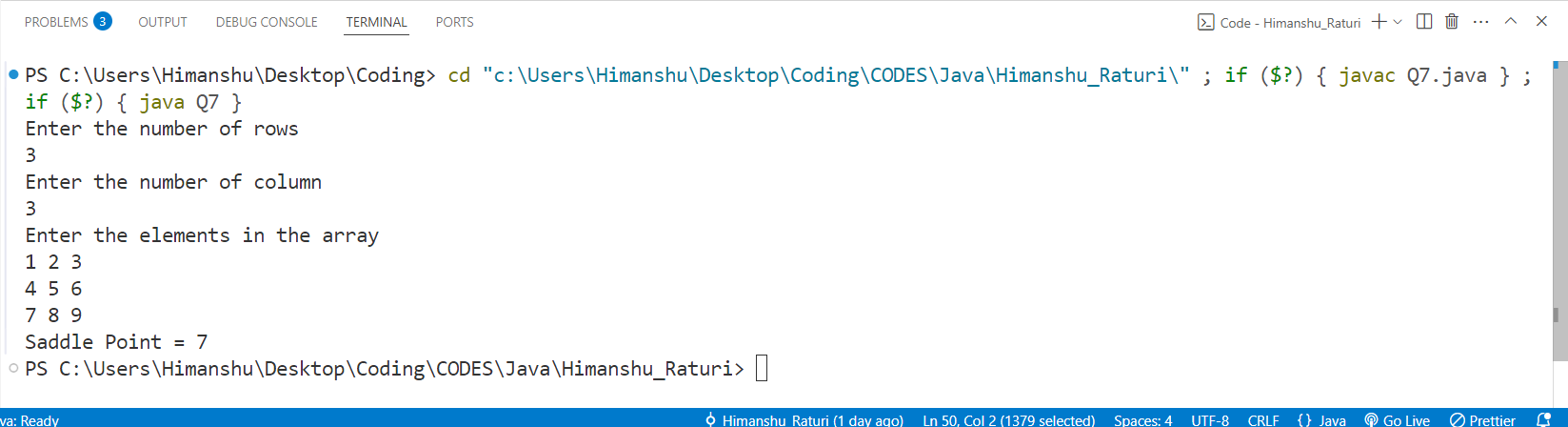
        }

        sc.close();

    }

}

**OUTPUT:**

**Practical No. 8:** Program to find all the patterns of 0(1+)0 in the given string. Given a string containing 0's and 1's, find the total number of 0(1+)0 patterns in the string and output it.   
0(1+)0 - There should be at least one '1' between the two 0's.

**Source Code:**

package CODES.Java.Himanshu\_Raturi;

import java.util.Scanner;

public class Q8

{

    public static void main(String args[])

    {

        String str;

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a string: ");

        str = sc.nextLine();

        //String str = new String("01101111010");

        int count = 0 ;

        for(int i =0 ; i < str.length() - 1; i++)

        {

            if(str.charAt(i) == '0' && str.charAt(i+1) == '1')

            {

                count++;

            }

        }

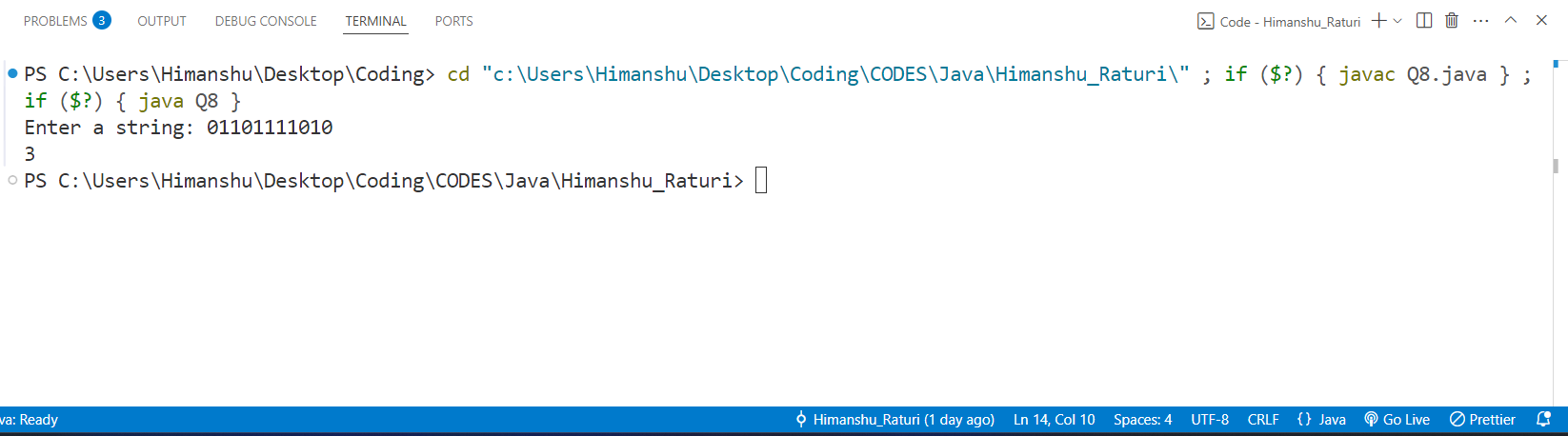
        System.out.println(count);

        sc.close();

    }

}

**Output:**



**Practical No. 9:** Write a java program to create a class named 'Bank '   
with the following data members:   
 Name of depositor   
 Address of depositor   
 Account Number   
 Balance in account   
   
Class 'Bank' has a method for each of the following:   
1 - Generate a unique account number for each   
depositor   
For first depositor, account number will be 1001, for   
second depositor it will be 1002 and so on   
2 - Display information and balance of depositor   
3 - Deposit more amount in balance of any depositor   
4 - Withdraw some amount from balance deposited   
5 - Change address of depositor   
   
**Source code:**

package CODES.Java.Himanshu\_Raturi;

import java.util.Scanner;

public class Q9\_Bank

{

String name , address ;

int accno;

double balance;

void setName(String name)

{

    this.name = name;

}

void setAddress(String address)

{

    this.address = address;

}

void setAccno(int i)

{

    this.accno = accno + 1001 + i;

}

void setBalance(double balance)

{

    this.balance = balance;

}

int getacc()

{

    return accno;

}

void display() {

System.out.println("Name: " + name + "\n" +

                    "Address: " + address + "\n" +

                    "Account Number: " + accno + "\n" +

                    "Balance:INR " + balance);

}

void deposit(int amt)

{

    balance += amt;

    System.out.println("INR " + amt+" has been successfully deposited.\n" + "Total amount is: "+balance);

}

void withdraw(int amt)

{

    if(amt > balance)

    {

        System.out.println("Insufficient Balance availble.");

    }else

    {

        balance -= amt;

        System.out.println(amt+" has been successfully withdrawed.\n" + "Total amount is: "+balance);

    }

}

void changeAddress(String add)

{

    System.out.println("Address has been successfully changed from "+ address + " to " + add);

    address = add;

}

public static void main(String args[])

{

    int n;

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter Number of depositors: ");

    n = sc.nextInt();

    Q9\_Bank depositors[] = new Q9\_Bank[n];

    for(int i = 0 ; i < n ; i++)

    {

        depositors[i] = new Q9\_Bank();

    }

    for(int i = 0 ; i < n; i++)

    {

        String name , address ;

        double balance;

        System.out.println("Enter Details of "  + " user:- " +(1001+i)+":" );

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

        name = sc.next();

        depositors[i].setName(name);

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

        address = sc.next();

        depositors[i].setAddress(address);

        depositors[i].setAccno(i);

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

        balance = sc.nextDouble();

        depositors[i].setBalance(balance);

    }

    int choice;

    int accno;

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

    accno = sc.nextInt();

    do

    {

    System.out.println("Press:\n" + "1 to Deposit Money\n" + "2 to withdraw money\n" + "3 to Change addres\n"+ "4 to display Information\n"+ "5 to exit.");

    choice = sc.nextInt();

    switch(choice)

    {

        case 1:

                {

                    int amt;

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

                    amt = sc.nextInt();

                    for(int i = 0 ; i < n ;i++)

                    {

                        if(depositors[i].getacc() == accno)

                        {

                            depositors[i].deposit(amt);

                            break;

                        }

                    }

                    break;

                }

        case 2:

                {

                    int amt;

                    System.out.println("Enter amount to Withdraw: ");

                    amt = sc.nextInt();

                    for(int i = 0 ; i < n ;i++)

                    {

                        if(depositors[i].getacc() == accno)

                        {

                            depositors[i].withdraw(amt);

                            break;

                        }

                    }

                    break;

                }

        case 3:

                {

                    String add;

                    System.out.println("Enter New address: ");

                    add = sc.next();

                    for(int i = 0 ; i < n ;i++)

                    {

                        if(depositors[i].getacc() == accno)

                        {

                            depositors[i].changeAddress(add);

                            break;

                        }

                    }

                    break;

                }

        case 4:

                {

                    for(int i = 0 ; i < n ;i++)

                    {

                        if(depositors[i].getacc() == accno)

                        {

                            depositors[i].display();

                            break;

                        }

                    }

                    break;

                }

    }

}while(choice != 5);

System.out.println("Exiting System.Thank you......");

sc.close();

}

}

**Output:**



