**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] + " ");

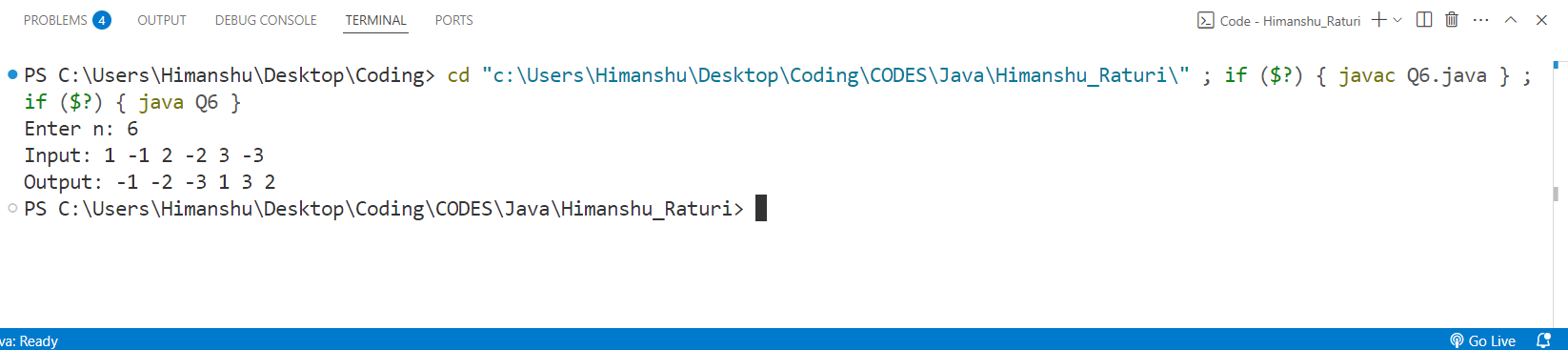
        }

        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;

            }

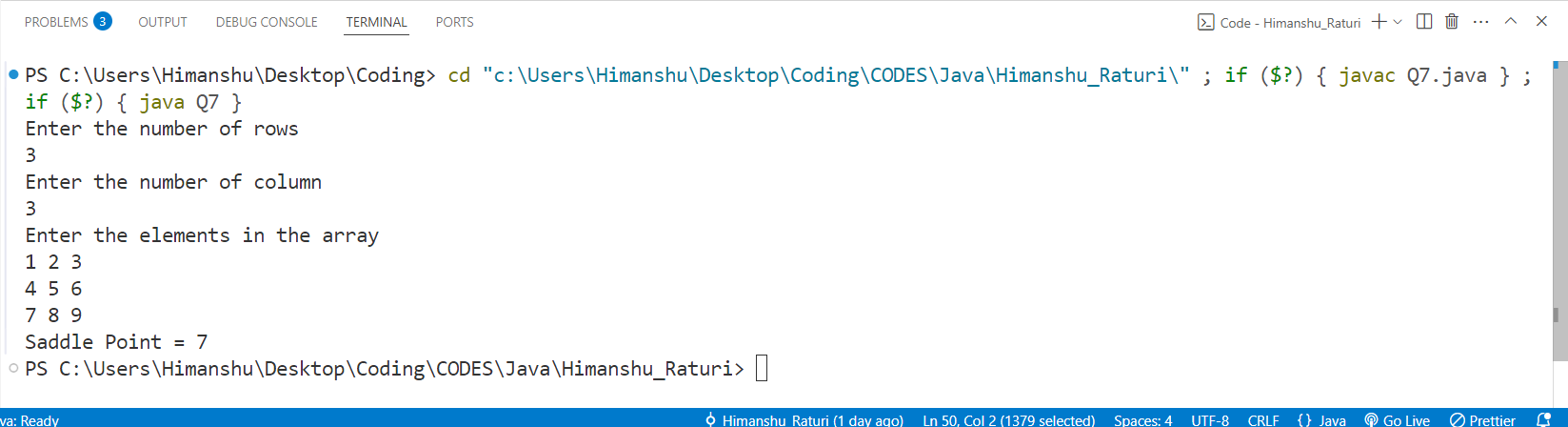
        }

        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++;

            }

        }

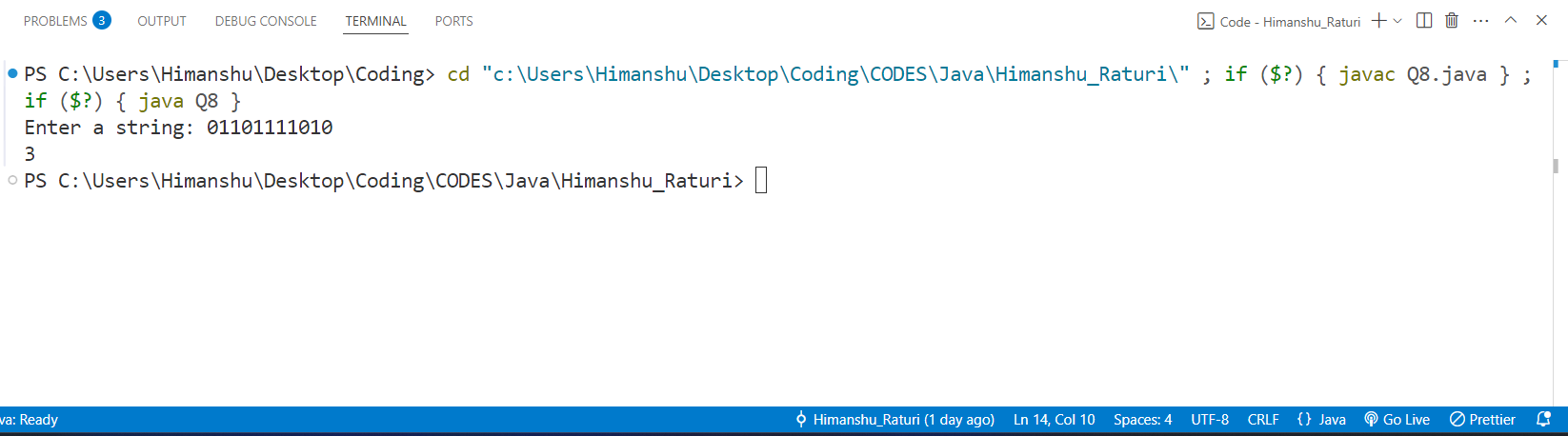
        System.out.println(count);

        sc.close();

    }

}

**Output:**



**Practical No. 8:** 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:**



