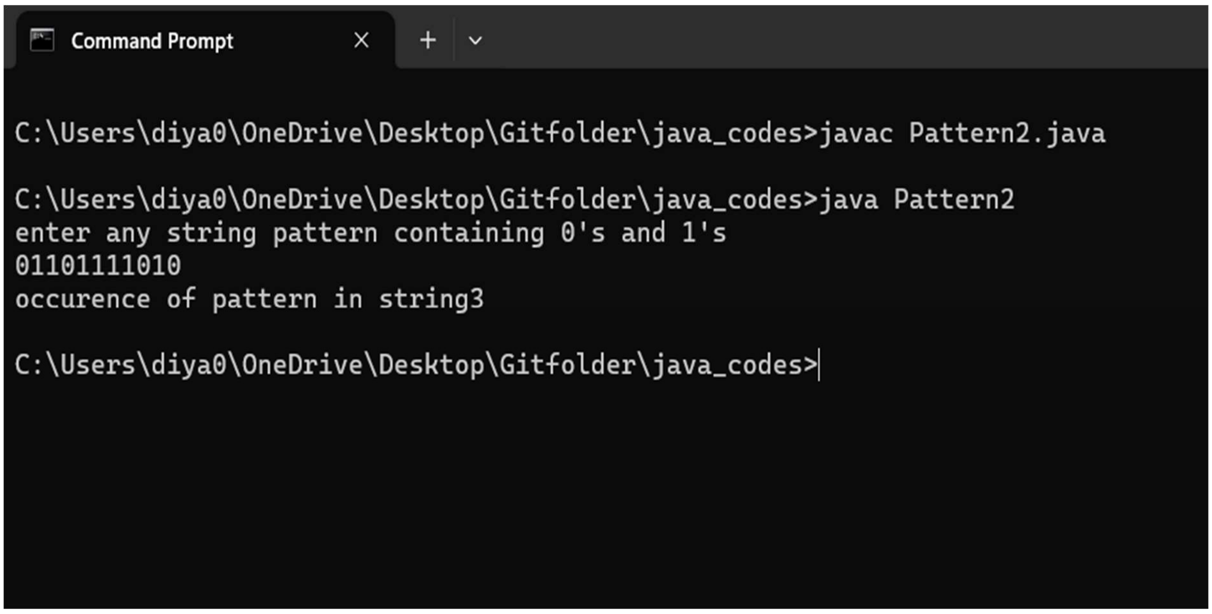
```java
/*Q23.Program to find all the patterns of 0(1+)0 in the given string. Given a string containing
0's and 1's, find the total number of 0(1+)0 patterns in the string and output it.

0(1+)0 - There should be at least one '1' between the two 0's.

For example, consider the following string.

Input: 01101111010

Output: 3

Explanation: 01101111010 - count = 1

*/


import java.util.*;

class Pattern2{

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        int count=0;

        System.out.println("enter any string pattern containing 0's and 1's");

        String s=sc.next();

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

        {

            if(s.charAt(i)=='0')

            {

                int j=i+1;

                while(j<s.length()&&s.charAt(j)!='0')

                {

                    j++;

                }

                if(j<s.length()&&j!=i+1)

                {

                    count++;

                }

                i=j;

            }
            else
            {
               i++;
            }
        }
        System.out.println("occurence of pattern in string"+count);


    }
}
```

Output: 3

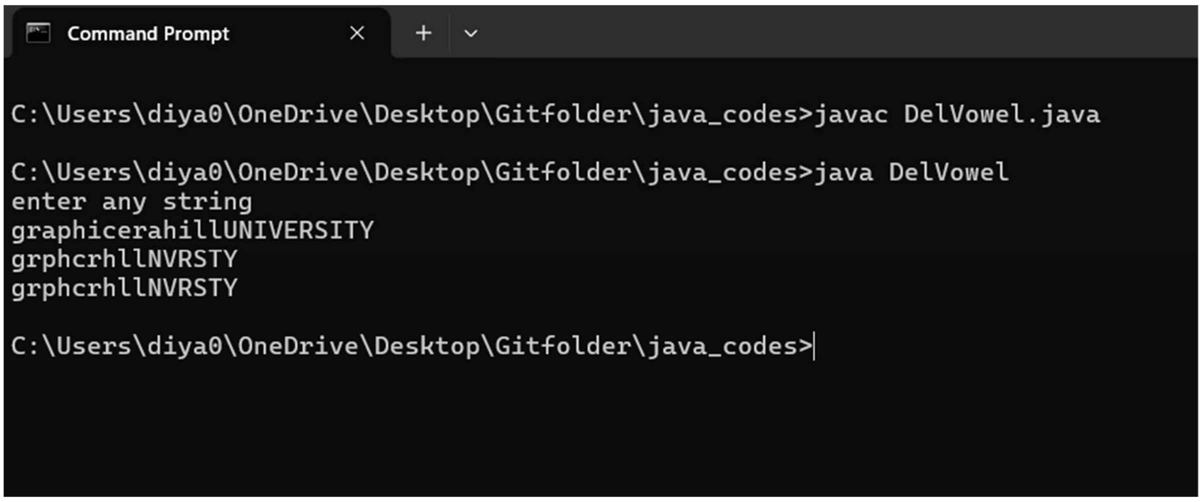


Command Prompt

```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac Pattern2.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java Pattern2
enter any string pattern containing 0's and 1's
01101111010
occurence of pattern in string3

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
//Q24.Write a java program to delete vowels using StringBuffer class.

import java.util.Scanner;
public class DelVowel {
   public static void main(String[] args) {
     Scanner sc=new Scanner(System.in);
     System.out.println("enter any string");
     StringBuffer sb=new StringBuffer(sc.nextLine());
     delVowel(sb);
   }
   public static void delVowel(StringBuffer sb){
       for(int i=0;i<sb.length();i++)
       {
          char ch=sb.charAt(i);
          if(isVowel(ch)){
             sb.deleteCharAt(i);
             i--;
          }
       }
       System.out.println(sb);
       System.out.println(sb.toString());//to show content as string.

   }
   public static boolean isVowel(char ch){
       return "aeiouAEIOU".indexOf(ch)!=-1;
   }
}
```

Output: grphcrhllNVRSTY



```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac DelVowel.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java DelVowel
enter any string
graphicerahillUNIVERSITY
grphcrhllNVRSTY
grphcrhllNVRSTY

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

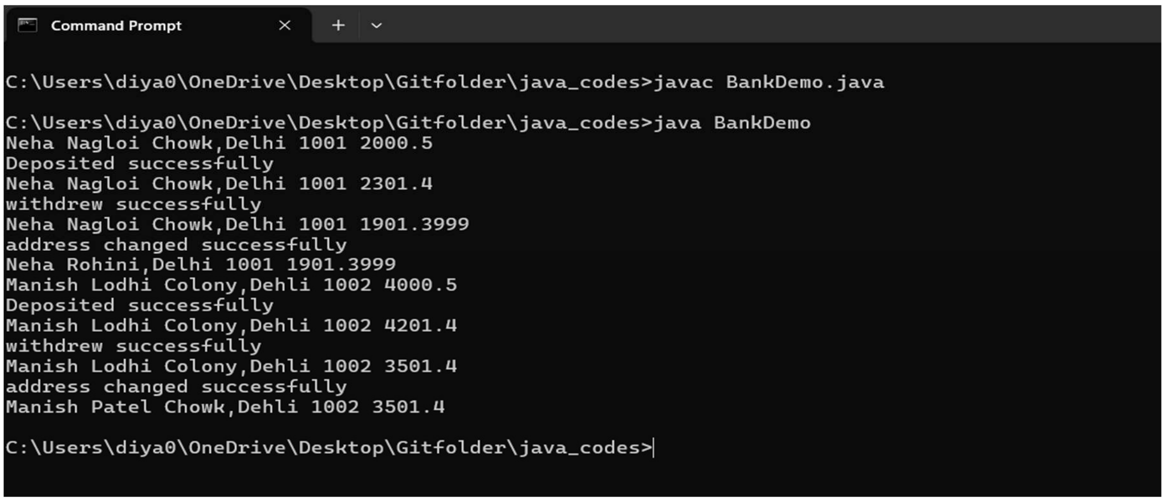
```java
/*Q25.Class definition, creating objects and constructors:

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.

2. For first depositor, account number will be 1001, for second depositor it will be 1002 and so
on

3. Display information and balance of depositor

4. Deposit more amount in balance of any depositor

5. Withdraw some amount from balance deposited.

6. Change address of depositor

2.After creating the class, do the following operations.

1. Enter the information (name, address, account number, balance) of the depositors. Number
of depositors is to be entered by the user.

2. Print the information of any depositor.

3. Add some amount to the account of any depositor and then display final information of that
depositor.

4. Remove some amount from the account of any depositor and then display final information
of that depositor.

5. Change the address of any depositor and then display the final information of that depositor.

6. Randomly repeat these processes for some other bank accounts.

*/


class Bank{

    private String name,add;

    private static int acc;

    private float bal;

    static{

        acc=1001;
```

```
}
Bank(String s ,String ad,float b){
   name=s;add=ad;bal=b;
}
public static void unique(){
   acc++;
}
public void display(){
   System.out.println(name+" "+add+" "+acc+" "+bal);
}
public void deposit(float d){
   bal=bal+d;
   System.out.println("Deposited successfully");
   display();
}
public void withdraw(float w){
   if(bal==0){
      System.out.println("zero amount");
   }
   else if(w>bal){
      System.out.println("not enough amount");
   }
   else{
      bal=bal-w;
      System.out.println("withdrew successfully ");
   }
   display();
}
public void changeAddress(String s){
   add=s;
```

```
        System.out.println("address changed successfully");
        display();
    }
}
public class BankDemo
{
        public static void main(String[] args) {
                Bank b=new Bank("Neha","Nagloi Chowk,Delhi",2000.50f);
                b.display();
                b.deposit(300.90f);
                b.withdraw(400.00f);
                b.changeAddress("Rohini,Delhi");
                Bank.unique();
                Bank b1=new Bank("Manish","Lodhi Colony,Dehli",4000.50f);
                b1.display();
                b1.deposit(200.90f);
                b1.withdraw(700.00f);
                b1.changeAddress("Patel Chowk,Dehli");
        }
}
```

Output: Neha Nagloi Chowk, Delhi 1001 2000.5

Manish Lodhi Colony, Delhi 1002 4000.5



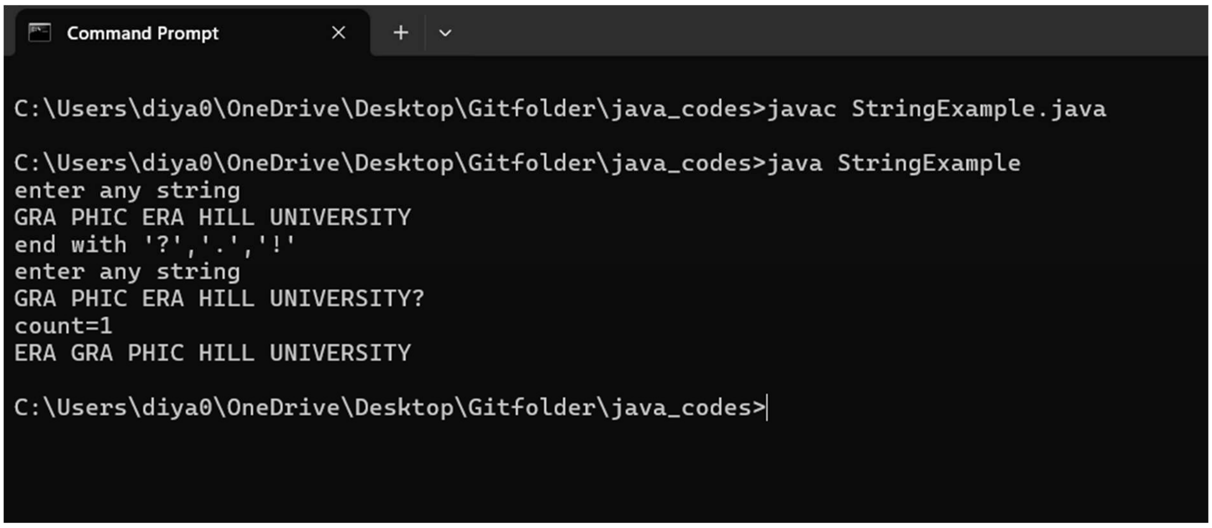
```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac BankDemo.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java BankDemo
Neha Nagloi Chowk,Delhi 1001 2000.5
Deposited successfully
Neha Nagloi Chowk,Delhi 1001 2301.4
withdrew successfully
Neha Nagloi Chowk,Delhi 1001 1901.3999
address changed successfully
Neha Rohini,Delhi 1001 1901.3999
Manish Lodhi Colony,Dehli 1002 4000.5
Deposited successfully
Manish Lodhi Colony,Dehli 1002 4201.4
withdrew successfully
Manish Lodhi Colony,Dehli 1002 3501.4
address changed successfully
Manish Patel Chowk,Dehli 1002 3501.4

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q26.Define a class Word Example having the followingdescription:
Data members/instance variables:
private String strdata: to store a sentence.
Parameterized Constructor WordExample(String) : Accept a
sentence which may be terminated by either'.', '? 'or'!' only. The wordsmay be separated by
more than one blank space and are in UPPER CASE.
Member Methods:
void countWord(): Find the number of wordsbeginning and
ending with a vowel.
void placeWord(): Place the words which begin andend with a vowel at the
beginning, followed by the remaining words as they occur in the sentence
*/

import java.util.*;
public class StringExample{
  public static void main(String[] args) {
    Scanner scn=new Scanner(System.in);
    String s;
    while(true){
    System.out.println("enter any string");
    s=scn.nextLine();
    String upper=s.toUpperCase();
    if(!s.equals(upper)){
      System.out.println("enter uppercase");
      continue;
    }
    if(s.charAt(s.length()-1)=='.'||s.charAt(s.length()-1)=='?'||s.charAt(s.length()-1)=='!'){
      break;
    }
    else
```

```
        {
           System.out.println("end with '?','.','!' ");
        }
        }
        WordExample w=new WordExample(s);
        w.countWord();
        w.placeWord();
    }
}
class WordExample{
    private String strdata;
    WordExample(String s){
        strdata=s;
    }
    void countWord(){
        int count=0;
        String []newstr=SplitString(strdata);
        for(int i=0;i<newstr.length;i++){
           if(isVowel(newstr[i].charAt(0))&&isVowel(newstr[i].charAt(newstr[i].length()-1))){
               count++;
           }
        }
        System.out.println("count="+count);

    }
    void placeWord(){
        String []newstr=SplitString(strdata);
        String res="";
        String vow="",nonvow="";
        for(int i=0;i<newstr.length;i++){
            if(isVowel(newstr[i].charAt(0))&&isVowel(newstr[i].charAt(newstr[i].length()-1))){
                vow=vow+" "+newstr[i];
            }
            else{
            nonvow=nonvow+" "+newstr[i];
            }
        }
        res=vow+nonvow;
        System.out.println(res.trim());
    }
    boolean isVowel(char ch){
        return "AEIOU".indexOf(ch)!=-1;
    }
    String[] SplitString(String strdata){
        String newstr[]=strdata.substring(0,strdata.length()-1).split("\\s+");
        return newstr;
    }
}
```

Output: 1

ERA GRA PHIC HILL UNIVERSITY



```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac StringExample.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java StringExample
enter any string
GRA PHIC ERA HILL UNIVERSITY
end with '?','.','!'
enter any string
GRA PHIC ERA HILL UNIVERSITY?
count=1
ERA GRA PHIC HILL UNIVERSITY

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q27.Method overloading (Compile time Polymorphism):

Write a Java program to create a class called ArrayDemo and overload arrayFunc() function.

void arrayFunc(int [], int) To find all pairs of elements in an Array whose sum is equal to a
given number :

Array numbers= [4, 6, 5, -10, 8, 5, 20], target=10

Output :

Pairs of elements whose sum is 10 are :4 + 6 = 10

5 + 5 = 10

-10 + 20 = 10

void arrayFunc(int A[], int p, int B[], int q) Given two sorted arrays A and B of size p and q,
Overload method arrayFunc() to merge elements of A with B by maintaining the sorted order
i.e. fill A with first p smallest elements and fill B with remaining elements.

Example:

Input :

2

int[] A = { 1, 5, 6, 7, 8, 10 }

int[] B = { 2, 4, 9 }

Output:

Sorted Arrays:

A: [1, 2, 4, 5, 6, 7]

B: [8, 9, 10]

(Use Compile time Polymorphism MethodOverloading)

*/


import java.util.*;

class arraydemo{

   public void arrayfun(int []ar,int key){

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

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

              if(ar[i]+ar[j]==key){

                  System.out.println(ar[i]+","+ar[j]);
              }
          }}}
   public void arrayfun(int ara[],int p,int arb[],int q){
        int n=ara.length;
        int m=arb.length;
        int r=n+m;
        int newar[]=new int[r];
        int i=0,j=0,k=0;
        while(i<n&&j<m){
        if(ara[i]<=arb[j]){
          newar[k++]=ara[i++];
        }
        else{
          newar[k++]=arb[j++];
        }
        }
        while(i<n){
          newar[k++]=ara[i++];
        }
        while(j<m){
          newar[k++]=arb[j++];
        }
        for( i=0;i<p;i++){
          ara[i]=newar[i];
          System.out.print(ara[i]+" ");
        }
        System.out.println();
        for(j=0;j<q;j++){
          arb[j]=newar[i++];
          System.out.print(arb[j]+" ");
```

```
        }}}
public class Overload{
    public static void main(String []arg){
        Scanner sc=new Scanner(System.in);
        int arr[]=new int[7];
        int arr1[]=new int[6];
        int arr2[]=new int[3];
        System.out.println("INPUT1");
        for(int i=0;i<7;i++){
            arr[i]=sc.nextInt();
        }
        int key=sc.nextInt();
        System.out.println("INPUT2");
         for(int i=0;i<6;i++){
            arr1[i]=sc.nextInt();
        }
        for(int i=0;i<3;i++){
            arr2[i]=sc.nextInt();
        }
        int p=arr1.length;
        int q=arr2.length;
        arraydemo a=new arraydemo();
        System.out.println("DISPLAY1");
        a.arrayfun(arr,key);
        System.out.println("DISPLAY2");
        a.arrayfun(arr1,p,arr2,q);    }
}
```

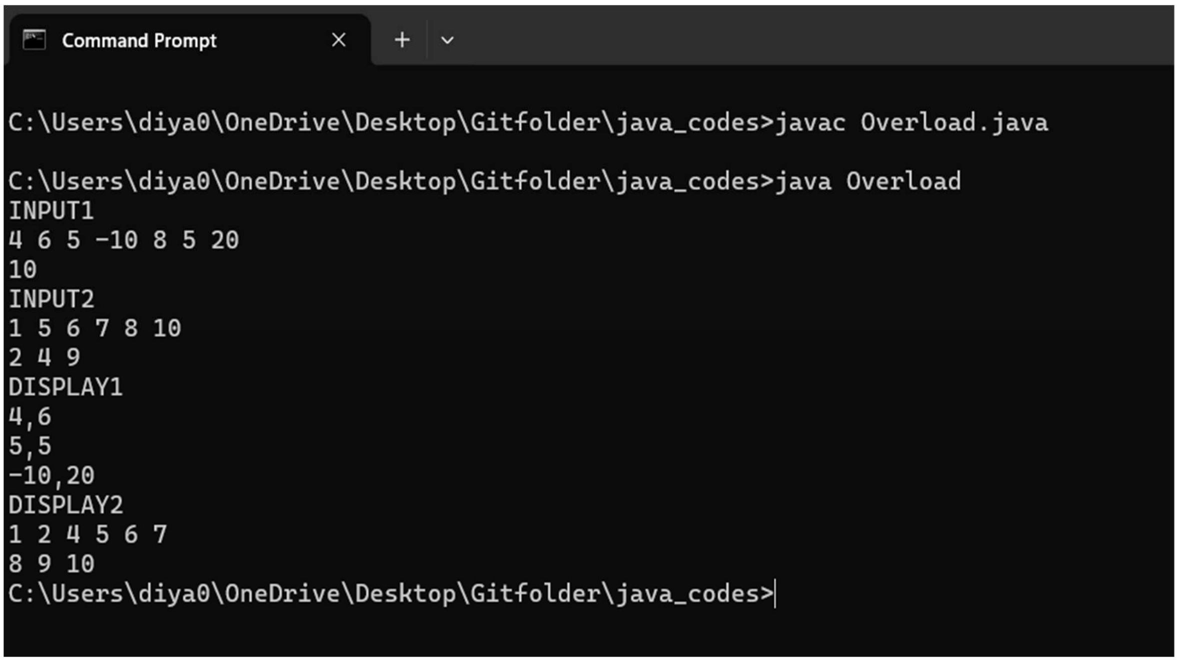
Output: 4,6

5,5

-10,20

1 2 3 4 5 6 7

8 9 10

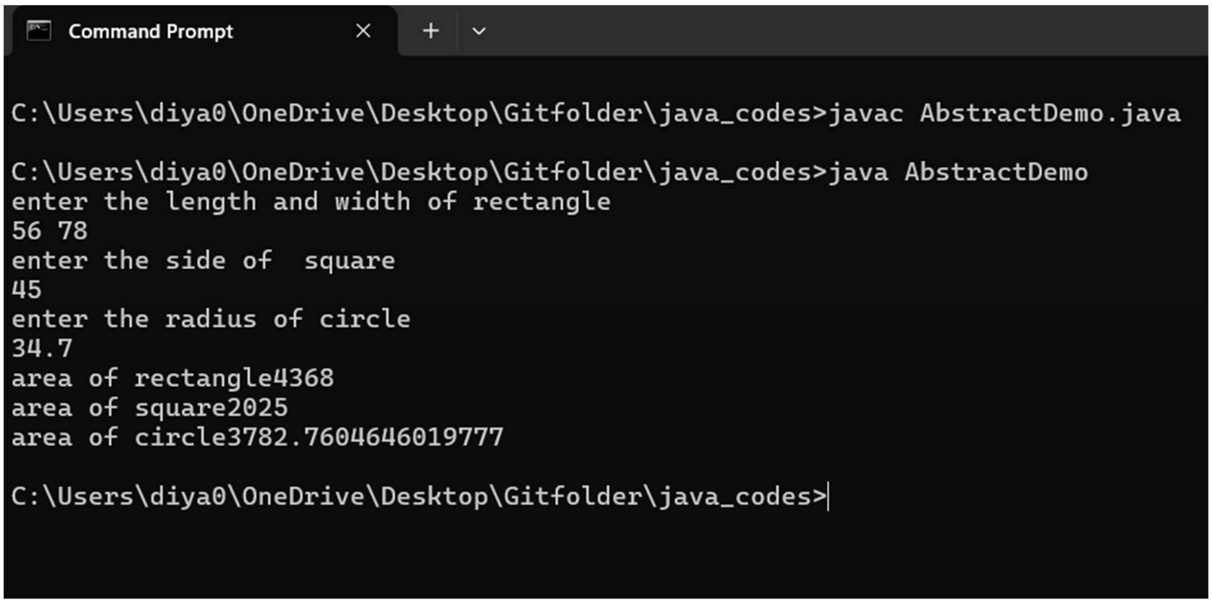


```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac Overload.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java Overload
INPUT1
4 6 5 -10 8 5 20
10
INPUT2
1 5 6 7 8 10
2 4 9
DISPLAY1
4,6
5,5
-10,20
DISPLAY2
1 2 4 5 6 7
8 9 10
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q28.Method overriding (Runtime Polymorphism), Abstract class and Abstract method,
Inheritance, interfaces:

Write a java program to calculate the area of a rectangle, a square and a circle.Create an abstract
class 'Shape' with three abstract methods namely rectangleArea() taking two parameters,
squareArea() and circleArea() takingone parameter each.Now create another class ‘Area’
containing all the three methods rectangleArea(), squareArea() and circleArea() for printing
the area of rectangle, square and circle respectively. Create an object of class Area and call all
the three methods.

(Use Runtime Polymorphism)

*/


import java.util.*;

abstract class Shape

{

   abstract public void rectanglearea(int l,int b);

   abstract public void squarearea(int a);

   abstract public void circlearea(float r);

}

class Area extends Shape{

   public void rectanglearea(int l,int b){

      System.out.println("area of rectangle"+l*b);

   }

   public void squarearea(int a){

       System.out.println("area of square"+a*a);

   }

   public void circlearea(float r){

       System.out.println("area of circle"+Math.PI*r*r);

   }

}

public class AbstractDemo{

   public static void main(String []arg){

      Scanner sc =new Scanner(System.in);
```

```
        System.out.println("enter the length and width of rectangle");
        int length=sc.nextInt();
        int breadth=sc.nextInt();
        System.out.println("enter the side of  square ");
        int s=sc.nextInt();
         System.out.println("enter the radius of circle");
        float r=sc.nextFloat();


        Area a=new Area();
        a.rectanglearea(length,breadth);
        a.squarearea(s);
        a.circlearea(r);
    }
}
```

Output:4368,2025,3782.76



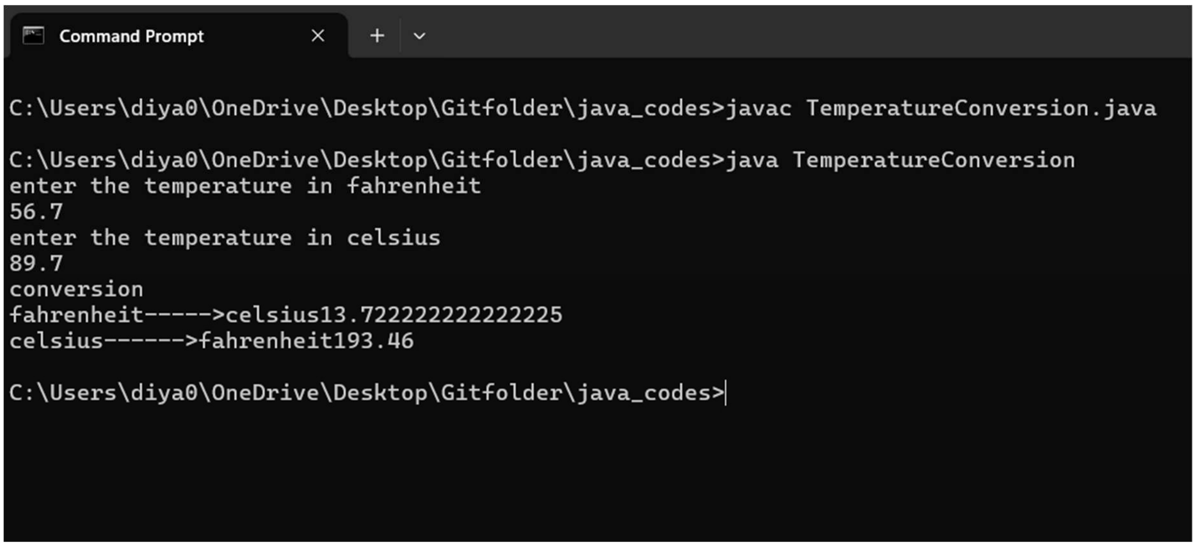
```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac AbstractDemo.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java AbstractDemo
enter the length and width of rectangle
56 78
enter the side of  square
45
enter the radius of circle
34.7
area of rectangle4368
area of square2025
area of circle3782.7604646019777

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q29.Write a java program to implement abstract class and abstract method with
following details:
Create a abstract Base Class TemperatureData members:
double temp;
Method members:
void setTempData(double) abstact void changeTemp()
Sub Class Fahrenheit (subclass of Temperature) Data members:
double ctemp;
method member:
Override abstract method changeTemp() to convert Fahrenheit temperature into degree Celsius
by using formula C=5/9*(F-32) and display converted temperature
Sub Class Celsius (subclass of Temperature)
Data member:
double ftemp;
Method member:
Override abstract method changeTemp() to convert degree Celsius into Fahrenheit temperature
by using formula F=9/5*c+32 and display converted temperature
*/


import java.util.*;
abstract class TemperatureData{
   double temp;
   public void setTempData(double t){
      temp=t;
   }
   abstract public void changeTemp();
}
class Fahrenheit extends TemperatureData{
   double ctemp;
   public void changeTemp(){
      ctemp=(5/9.0)*(temp-32);
```

```
    }
    public void display(){
        System.out.println("fahrenheit----->celsius"+ctemp);
    }
}
class Celsius extends TemperatureData{
    double ftemp;
    public void changeTemp(){
        ftemp=9/5.0*temp+32;
    }
    public void display(){
        System.out.println("celsius------>fahrenheit"+ftemp);
    }
}
public class TemperatureConversion{
    public static void main(String []arg){
        Scanner sc=new Scanner(System.in);
        Fahrenheit f=new Fahrenheit();
        System.out.println("enter the temperature in fahrenheit");
        double fah=sc.nextDouble();
        f.setTempData(fah);
        Celsius c=new Celsius();
        System.out.println("enter the temperature in celsius");
        double cel=sc.nextDouble();
        c.setTempData(cel);
        System.out.println("conversion");
        f.changeTemp();
        c.changeTemp();
        f.display();
        c.display();
    }
}
```

Output:13.72

193.46

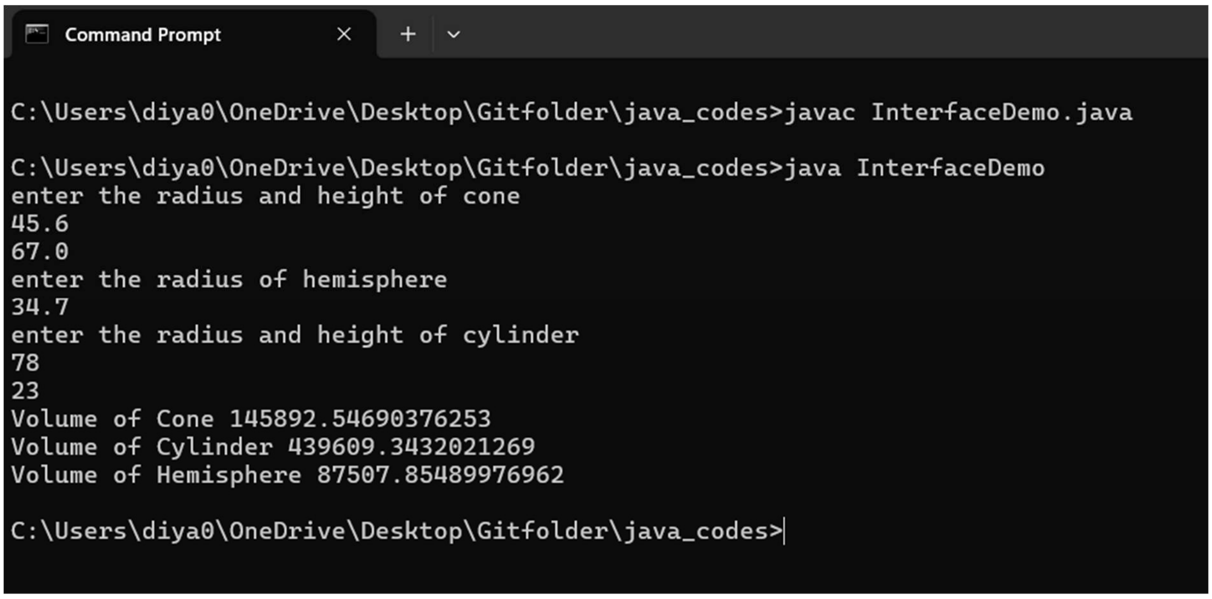


```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac TemperatureConversion.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java TemperatureConversion
enter the temperature in fahrenheit
56.7
enter the temperature in celsius
89.7
conversion
fahrenheit----->celsius13.722222222222225
celsius------>fahrenheit193.46

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q30.Write a java program to create an interface that consists of a method to display volume()
as an abstract method and redefine this method in the derived 2 classes to suit their
requirements. Create classes called Cone, Hemisphere and Cylinder that implements the

interface. Using these three classes, design a program that will accept dimensions of a cone,
cylinder and hemisphere interactively and display the volumes.

Volume of cone = (1/3)πr2h Volume of hemisphere = (2/3)πr3Volume of cylinder = πr2h

*/


import java.util.*;

interface MyVolume{

   public void volume();

}

class Cone implements MyVolume{

   double r,h;

   public void setDim(double r,double h){

      this.r=r;

      this.h=h;

   }

   public void volume(){

      System.out.println("Volume of Cone"+" "+ 1/3.0*Math.PI*r*r*h);

   }

}

class Hemisphere implements MyVolume{

     double r;

   public void setDim(double r){

      this.r=r;

   }

   public void volume(){

      System.out.println("Volume of Hemisphere"+" "+ 2/3.0*Math.PI*r*r*r);

   }

}
```

```
class Cylinder implements MyVolume{
      double r,h;
    public void setDim(double r,double h){
        this.r=r;
        this.h=h;
    }
    public void volume(){
        System.out.println("Volume of Cylinder"+" "+ Math.PI*r*r*h);
    }
}
public class InterfaceDemo{
    public static void main(String []arg){
        Scanner sc=new Scanner(System.in);
        Cone c=new Cone();
        Cylinder cl=new Cylinder();
        Hemisphere h=new Hemisphere();
        System.out.println("enter the radius and height of cone");
        double rcone=sc.nextDouble();
        double hcone=sc.nextDouble();
        c.setDim(rcone,hcone);
        System.out.println("enter the radius of hemisphere");
        double rhemis=sc.nextDouble();
        h.setDim(rhemis);
        System.out.println("enter the radius and height of cylinder");
        double rcy=sc.nextDouble();
        double hcy=sc.nextDouble();
        cl.setDim(rcy,hcy);
        c.volume();
        cl.volume();
        h.volume();
    }
}
```

Output:145892.54,439609.34,87507.85

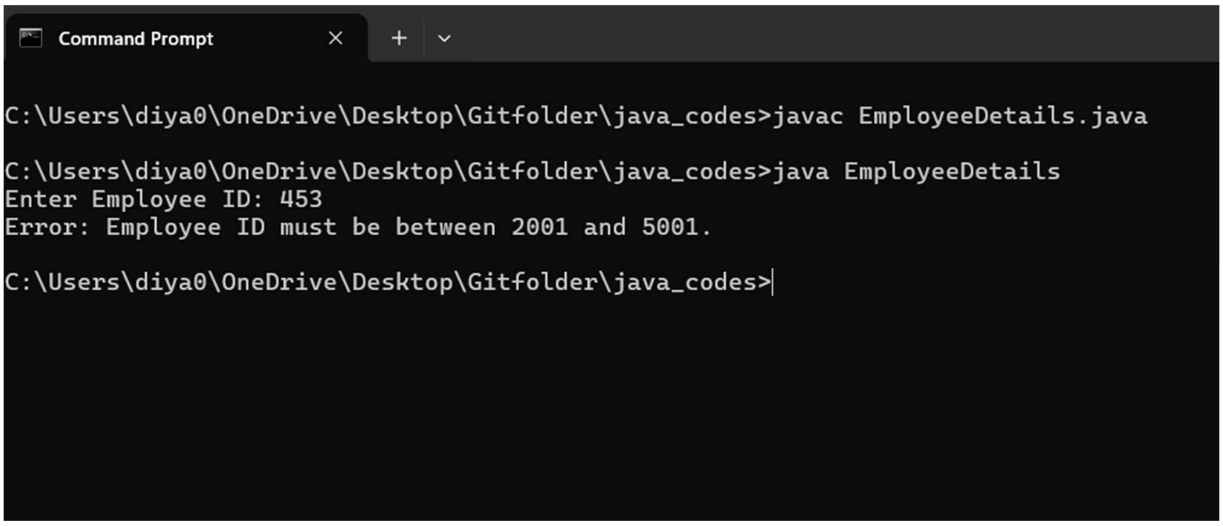


Command Prompt

```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac InterfaceDemo.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java InterfaceDemo
enter the radius and height of cone
45.6
67.0
enter the radius of hemisphere
34.7
enter the radius and height of cylinder
78
23
Volume of Cone 145892.54690376253
Volume of Cylinder 439609.3432021269
Volume of Hemisphere 87507.85489976962

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q31.Write a java program to accept and print the employee details during runtime.The details
will include employee id, name, dept_ Id. The program should raise an exception if user inputs
incomplete or incorrect data. The entered value should meet the following conditions:

a. First Letter of employee name should be in capital letter.

b. Employee id should be between 2001 and 5001

c. Department id should be an integer between 1 and 5.

If the above conditions are not met, then the application should raise specific exception else
should complete normal execution.*/


import java.util.Scanner;


class InvalidEmployeeNameException extends Exception {
   public InvalidEmployeeNameException(String message) {
      super(message);
   }
}


class InvalidEmployeeIdException extends Exception {
   public InvalidEmployeeIdException(String message) {
      super(message);
   }
}


class InvalidDepartmentIdException extends Exception {
   public InvalidDepartmentIdException(String message) {
      super(message);
   }
}


public class EmployeeDetails {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            // Accept Employee ID
            System.out.print("Enter Employee ID: ");
            int employeeId = scanner.nextInt();
            if (employeeId < 2001 || employeeId > 5001) {
                throw new InvalidEmployeeIdException("Employee ID must be between 2001 and
5001.");
            }

            // Accept Employee Name
            System.out.print("Enter Employee Name: ");
            String employeeName = scanner.next();
            if (Character.isLowerCase(employeeName.charAt(0))) {
                throw new InvalidEmployeeNameException("First letter of Employee Name must
be a capital letter.");
            }

            // Accept Department ID
            System.out.print("Enter Department ID: ");
            int deptId = scanner.nextInt();
            if (deptId < 1 || deptId > 5) {
                throw new InvalidDepartmentIdException("Department ID must be between 1 and
5.");
            }

            // Print Employee Details
            System.out.println("Employee Details:");
            System.out.println("Employee ID: " + employeeId);
            System.out.println("Employee Name: " + employeeName);

            System.out.println("Department ID: " + deptId);


        } catch (InvalidEmployeeIdException | InvalidEmployeeNameException |
InvalidDepartmentIdException e) {

            System.out.println("Error: " + e.getMessage());

        } finally {

            scanner.close();

        }

    }

}
```

Output: Employee ID must be between 2001 and 5001



```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac EmployeeDetails.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java EmployeeDetails
Enter Employee ID: 453
Error: Employee ID must be between 2001 and 5001.

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q32.Create a class MyCalculator which consists of a single method power (int, int). This
method takes two integers, n and p, as parameters and finds np. If either n or p is negative,
then the method must throw an exception which says, "n and p should be non- negative".

Input Format

Each line of the input contains two integers, n and p. Output Format

Each line of the output contains the result, if neither of n and p is negative.

Otherwise, the output contains "n and p should be non- negative".

Sample Input

-1 0

  4 5

  0 0

 -2 -5

Sample Output

n and p should be non-negative

1024

n and p should not be zero

n and p should be non-negative


java.lang.Exception: n and p should not be zero. java.lang.Exception: n or p

should not be negative. java. lang. Exception: n or p should not be negative.

Explanation

In the first two cases, both n and p are positive. So, the power function returns

the answer correctly.

In the third case, both n and p are zero. So, the exception, "n and p should not

be zero." is printed.

In the last two cases, at least one out of n and p is negative. So, the exception,

"n or p should not be negative." is printed for these two cases.

*/


import java.util.Scanner;
```

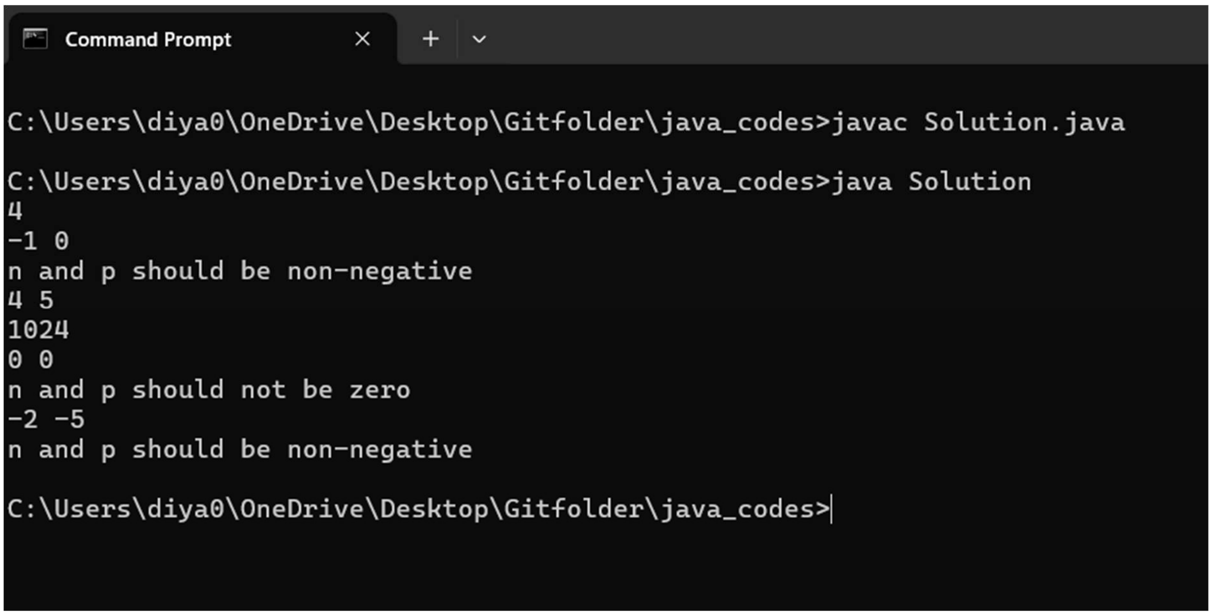
```
class MyCalculator {
    // Method to calculate n raised to the power of p
    public long power(int n, int p) throws Exception {
        if (n < 0 || p < 0) {
            throw new Exception("n and p should be non-negative");
        } else if (n == 0 && p == 0) {
            throw new Exception("n and p should not be zero");
        } else {
            return (long) Math.pow(n, p);
        }
    }
}

public class Solution {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MyCalculator myCalculator = new MyCalculator();
        int numberOfPairs = sc.nextInt(); // Assuming the first input is the number of pairs
        for (int i = 0; i < numberOfPairs; i++) {
            int n = sc.nextInt(); // Read the next integer n
            int p = sc.nextInt(); // Read the next integer p
            try {
                System.out.println(myCalculator.power(n, p));
            } catch (Exception e) {
                System.out.println(e.getMessage());
            }
        }
        sc.close();
    }
}
```

Output: n and p should be non-negative

1024

n and p should not be zero

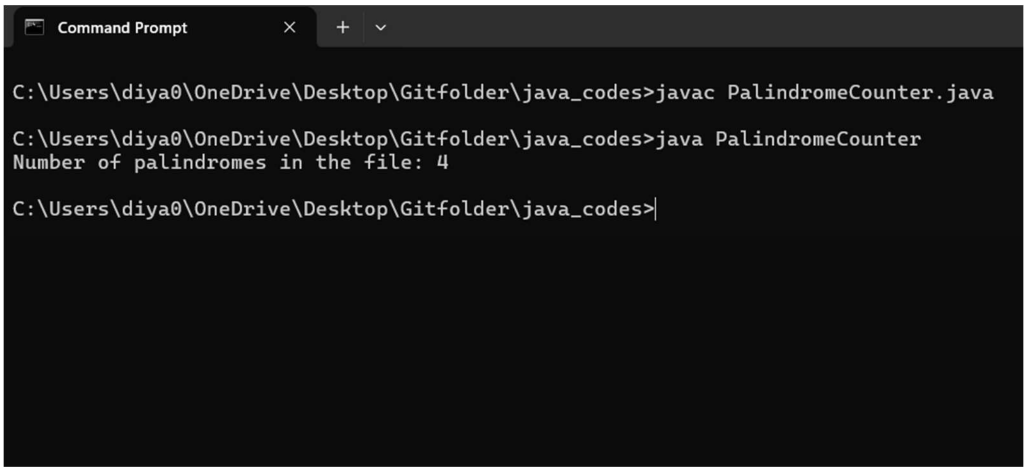
n and p should be non-negative



```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac Solution.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java Solution
4
-1 0
n and p should be non-negative
4 5
1024
0 0
n and p should not be zero
-2 -5
n and p should be non-negative

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q33.File Handling in Java:
Write a java file handling program to count and display the number of palindromes present in
a text file "myfile.txt".
Example: If the file "myfile.txt" contains the following lines,
My name is NITIN
Hello aaa and bbb wordHow are You
ARORA is my friend
Output will be => 4*/


import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;


public class PalindromeCounter {
    public static void main(String[] args) {
        String filename = "myfile.txt";

        try (BufferedReader br = new BufferedReader(new FileReader(myfile.txt))) {
            String line;
            int palindromeCount = 0;

            while ((line = br.readLine()) != null) {
                String[] words = line.split("\\s+");
                for (String word : words) {
                    if (isPalindrome(word)) {
                        palindromeCount++;
                    }
                }
            }

            System.out.println("Number of palindromes in the file: " + palindromeCount);
        } catch (IOException e) {
            System.err.println("Error reading file: " + e.getMessage());
        }
    }

    // Method to check if a string is a palindrome
    private static boolean isPalindrome(String str) {
        str = str.toLowerCase();
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

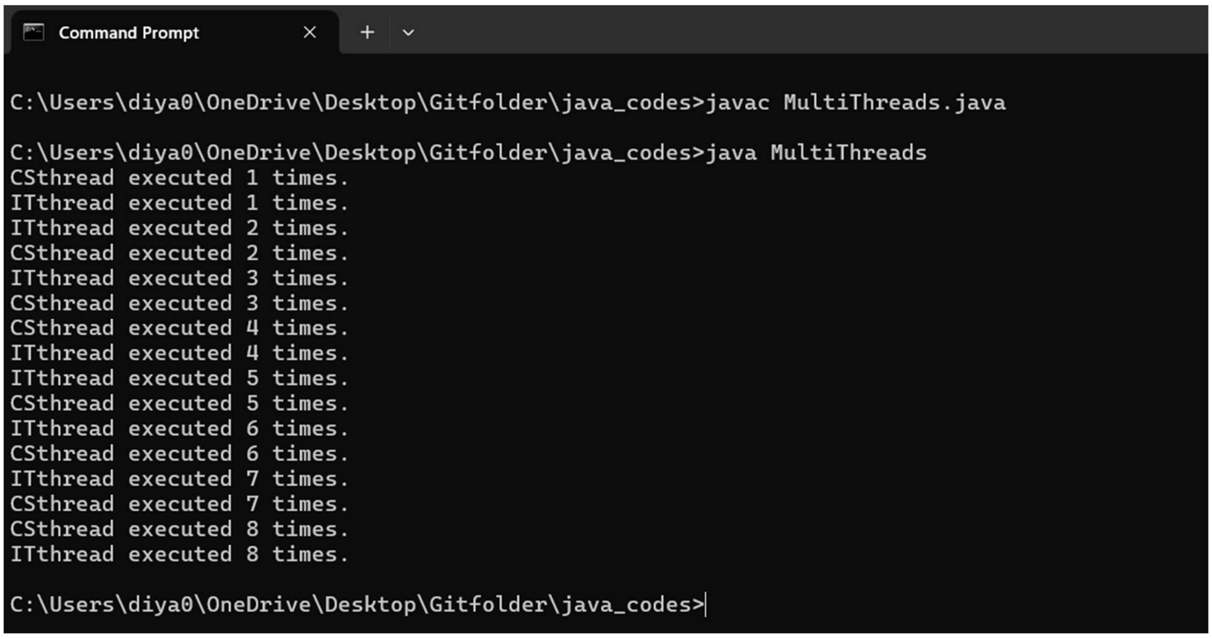
Output: 4



```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac PalindromeCounter.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java PalindromeCounter
Number of palindromes in the file: 4

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q34.Multithreaded programming:

Write a program MultiThreads that creates two threads-one thread with the name CSthread and
the other thread named ITthread.

Each thread should display its respective name and execute after a gap of 500 milliseconds.
Each thread should also display a number indicating the number of times it got a chance to
execute.*/


public class MultiThreads {
    public static void main(String[] args) {
        // Create two threads with specified names
        Thread csThread = new Thread(new CustomRunnable("CSthread"), "CSthread");
        Thread itThread = new Thread(new CustomRunnable("ITthread"), "ITthread");

        // Start both threads
        csThread.start();
        itThread.start();
    }
}

class CustomRunnable implements Runnable {
    private final String threadName;
    private int executionCount = 0;

    public CustomRunnable(String threadName) {
        this.threadName = threadName;
    }

    public void run() {
        try {
            // Loop to repeatedly execute the thread
            while (executionCount < 8) {
                // Increment the execution count
                executionCount++;
                // Display the thread's name and execution count
                System.out.println(threadName + " executed " + executionCount + " times.");
                // Sleep for 500 milliseconds
                Thread.sleep(500);
            }
        } catch (InterruptedException e) {
            System.err.println(threadName + " interrupted.");
        }
    }
}
```

Output:

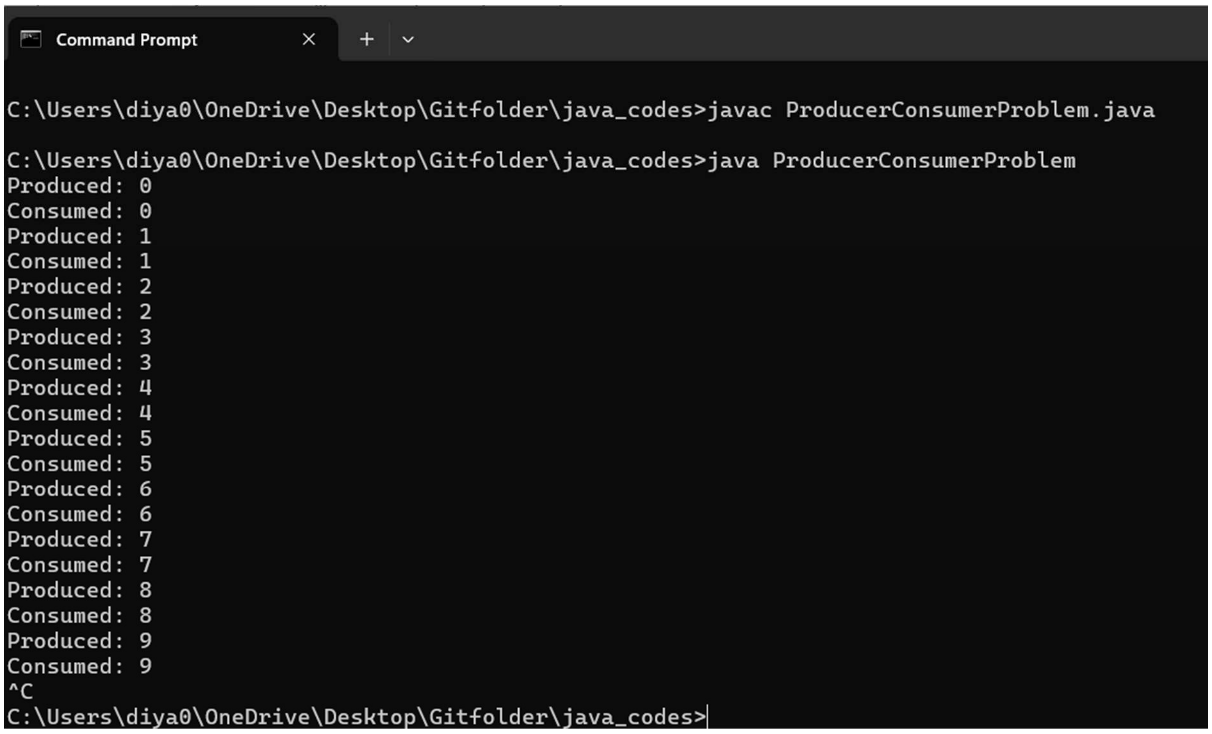


Command Prompt

```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac MultiThreads.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java MultiThreads
CSthread executed 1 times.
ITthread executed 1 times.
ITthread executed 2 times.
CSthread executed 2 times.
ITthread executed 3 times.
CSthread executed 3 times.
CSthread executed 4 times.
ITthread executed 4 times.
ITthread executed 5 times.
CSthread executed 5 times.
ITthread executed 6 times.
CSthread executed 6 times.
ITthread executed 7 times.
CSthread executed 7 times.
CSthread executed 8 times.
ITthread executed 8 times.

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q35.Write a java program for to solve producer consumer problem in which a
producer produces a value and consumer consume the value before producer
generate the next value.*/

class SharedBuffer {
    private int value;
    private boolean hasValue = false;

    public synchronized void produce(int newValue) throws InterruptedException {
        while (hasValue) {
            wait();
        }
        value = newValue;
        hasValue = true;
        System.out.println("Produced: " + value);
        notify();
    }

    public synchronized void consume() throws InterruptedException {
        while (!hasValue) {
            wait();
        }
        System.out.println("Consumed: " + value);
        hasValue = false;
        notify();
    }
}

class Producer implements Runnable {
    private SharedBuffer buffer;
    public Producer(SharedBuffer buffer) {
        this.buffer = buffer;
    }

    public void run() {
        int i = 0;
        while (true) {
            try {
                buffer.produce(i++);
                Thread.sleep(500); // Simulate time taken to produce
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class Consumer implements Runnable {
    private SharedBuffer buffer;

    public Consumer(SharedBuffer buffer) {
        this.buffer = buffer;
    }

    public void run() {
        while (true) {
            try {
                buffer.consume();
                Thread.sleep(500); // Simulate time taken to consume
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class ProducerConsumerProblem {
    public static void main(String[] args) {
        SharedBuffer buffer = new SharedBuffer();
        Thread producerThread = new Thread(new Producer(buffer));
        Thread consumerThread = new Thread(new Consumer(buffer));

        producerThread.start();
        consumerThread.start();
    }
}           } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class ProducerConsumerProblem {
    public static void main(String[] args) {
        SharedBuffer buffer = new SharedBuffer();
        Thread producerThread = new Thread(new Producer(buffer));
        Thread consumerThread = new Thread(new Consumer(buffer));

        producerThread.start();
        consumerThread.start();
    }
}
```

Output:

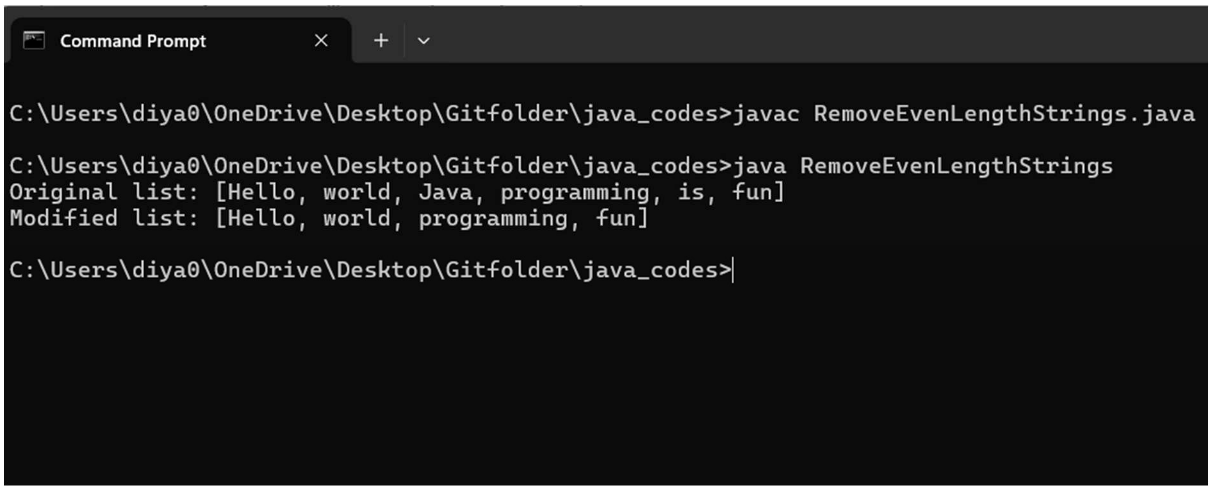


```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac ProducerConsumerProblem.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java ProducerConsumerProblem
Produced: 0
Consumed: 0
Produced: 1
Consumed: 1
Produced: 2
Consumed: 2
Produced: 3
Consumed: 3
Produced: 4
Consumed: 4
Produced: 5
Consumed: 5
Produced: 6
Consumed: 6
Produced: 7
Consumed: 7
Produced: 8
Consumed: 8
Produced: 9
Consumed: 9
^C
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q36.Collection and Generic Framework:
Write a method removeEvenLength that takes an ArrayList of Strings as a parameter and that
removes all the strings of even length from the list.
(Use ArrayList)*/


import java.util.ArrayList;
import java.util.Iterator;


public class RemoveEvenLengthStrings {
   public static void removeEvenLength(ArrayList<String> list) {
      // Use an iterator to safely remove elements while iterating
      Iterator<String> iterator = list.iterator();


      while (iterator.hasNext()) {
         String str = iterator.next();
         // Check if the length of the string is even
         if (str.length() % 2 == 0) {
            // Remove the string if its length is even
            iterator.remove();
         }
      }
   }


   public static void main(String[] args) {
      // Example usage
      ArrayList<String> strings = new ArrayList<>();
      strings.add("Hello");
      strings.add("world");
      strings.add("Java");
      strings.add("programming");
      strings.add("is");
```

```
        strings.add("fun");

        System.out.println("Original list: " + strings);

        // Remove strings of even length
        removeEvenLength(strings);

        System.out.println("Modified list: " + strings);
    }
}
```

Output: [Hello,world,programming,fun]

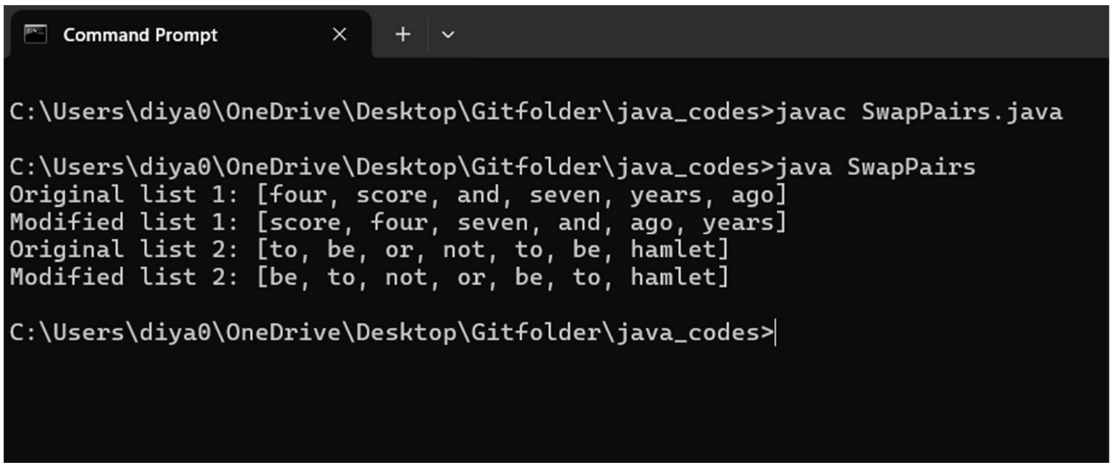


```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac RemoveEvenLengthStrings.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java RemoveEvenLengthStrings
Original list: [Hello, world, Java, programming, is, fun]
Modified list: [Hello, world, programming, fun]

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

```
/*Q37.Write a method swapPairs that switches the order of values in an ArrayList of Strings
in a pairwise fashion. Your method should switch the order of the first two values, then switch
the order of the next two, switch the order of the next two, and so on.

For example, if the list initially stores these values: {"four", "score", "and","seven", "years",

"ago"} your method should switch the first pair, "four", "score", the second pair, "and",
"seven", and the third pair, "years", "ago", to yield this list:{"score", "four", "seven", "and",
"ago", "years"}.If there are an odd number of values in the list, the final element is not moved.

For example, if the original list had been: {"to", "be", "or", "not", "to", "be",

"hamlet"} It would again switch pairs of values, but the final value, "hamlet"

would not be moved, yielding this list: {"be", "to", "not", "or", "be", "to",

"hamlet"}

*/


import java.util.ArrayList;


public class SwapPairs {

    public static void swapPairs(ArrayList<String> list) {

        for (int i = 0; i < list.size() - 1; i += 2) {

            // Swap the elements at index i and i+1

            String temp = list.get(i);

            list.set(i, list.get(i + 1));

            list.set(i + 1, temp);

        }

    }

    public static void main(String[] args) {

        // Example 1

        ArrayList<String> list1 = new ArrayList<>();

        list1.add("four");

        list1.add("score");

        list1.add("and");

        list1.add("seven");

        list1.add("years");
        list1.add("ago");
        System.out.println("Original list 1: " + list1);
        swapPairs(list1);
        System.out.println("Modified list 1: " + list1);
        ArrayList<String> list2 = new ArrayList<>();
        list2.add("to");
        list2.add("be");
        list2.add("or");
        list2.add("not");
        list2.add("to");
        list2.add("be");
        list2.add("hamlet");
        System.out.println("Original list 2: " + list2);
        swapPairs(list2);
        System.out.println("Modified list 2: " + list2);
    }
}
```

Output: [score,four,seven,and,ago,years]

[be,to,not,or,be,to,hamlet]



```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac SwapPairs.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java SwapPairs
Original list 1: [four, score, and, seven, years, ago]
Modified list 1: [score, four, seven, and, ago, years]
Original list 2: [to, be, or, not, to, be, hamlet]
Modified list 2: [be, to, not, or, be, to, hamlet]

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

/*Q38.Write a method called alternate that accepts two Lists of integers as its

parameters and returns a new List containing alternating elements from the twolists, in the

following order:

• First element from first list

• First element from second list

• Second element from first list

• Second element from second list

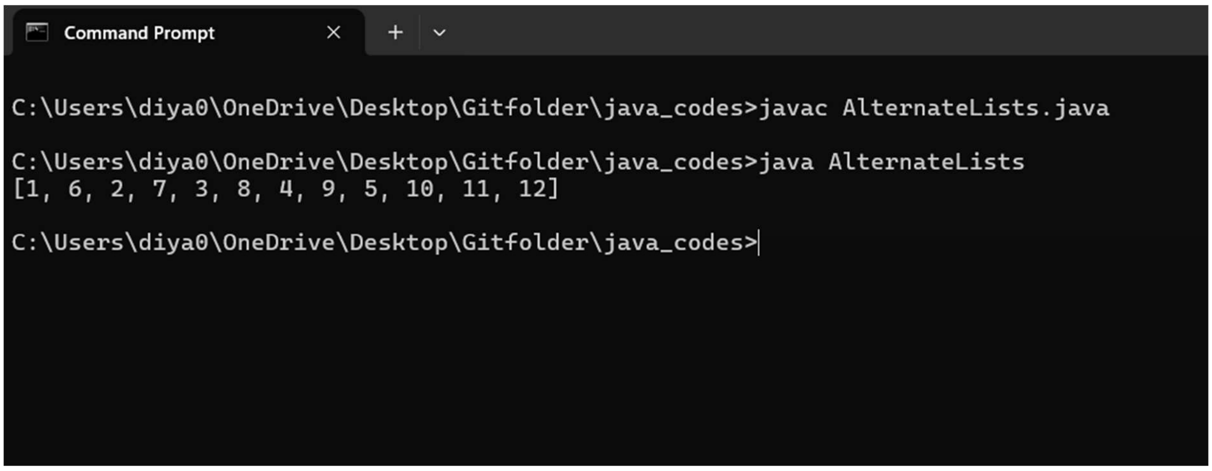
• Third element from first list

• Third element from second list

If the lists do not contain the same number of elements, the remaining elements from the longer list should be placed consecutively at the end. For example, for a first list of (1, 2, 3, 4, 5) and a second list of (6, 7, 8, 9, 10, 11, 12), a call of alternate (list1,list2) should return a list containing (1, 6, 2, 7, 3, 8, 4,9, 5, 10, 11, 12). Do not modify the parameter lists passed in.*/

```java
import java.util.ArrayList;
import java.util.List;

public class AlternateLists {
  public static List<Integer> alternate(List<Integer> list1, List<Integer> list2) {
    List<Integer> result = new ArrayList<>();

    int size1 = list1.size();
    int size2 = list2.size();
    int maxSize = Math.max(size1, size2);

    // Iterate through both lists simultaneously
    for (int i = 0; i < maxSize; i++) {
      if (i < size1) {
        result.add(list1.get(i)); // Add element from list1
      }
      if (i < size2) {
        result.add(list2.get(i)); // Add element from list2
      }
    }

    return result;
  }

  public static void main(String[] args) {
    List<Integer> list1 = List.of(1, 2, 3, 4, 5);
    List<Integer> list2 = List.of(6, 7, 8, 9, 10, 11, 12);
    List<Integer> result = alternate(list1, list2);
    System.out.println(result); // Output: [1, 6, 2, 7, 3, 8, 4, 9, 5, 10, 11, 12]
  }
}
```

Output: [1,6,2,7,3,8,4,9,5,10,11,12]



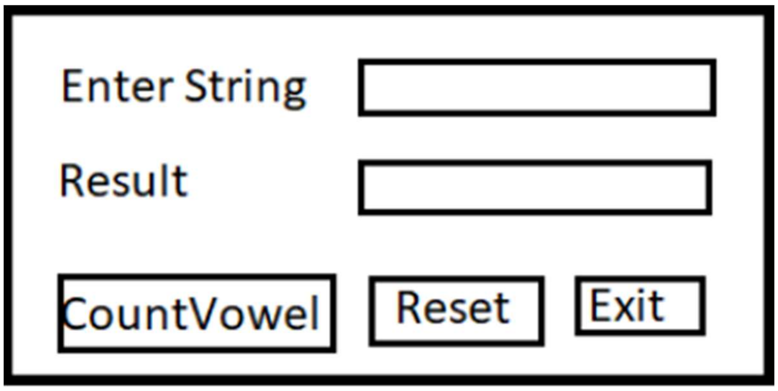
```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac AlternateLists.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java AlternateLists
[1, 6, 2, 7, 3, 8, 4, 9, 5, 10, 11, 12]

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```

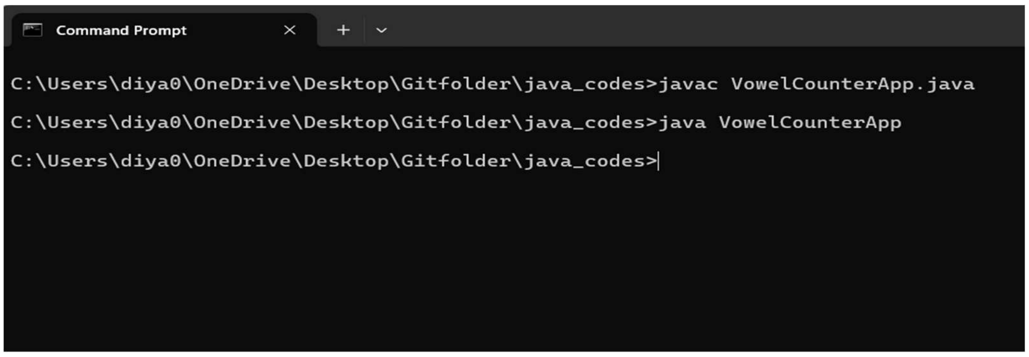
```
/*Q39.AWT & Swing, Event Handling:
Write a GUI program to develop an application that receives a string in one text field, and count
number of vowels in a string and returns it in another text field, when the button named
“CountVowel” is clicked.
When the button named “Reset” is clicked it will reset the value of
textfield one and Textfield two.
When the button named “Exit” is clicked it will closed the application*/


import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;


public class VowelCounterApp {
    public static void main(String[] args) {
        // Create the frame
        JFrame frame = new JFrame("Vowel Counter");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);
        frame.setLayout(new GridLayout(3, 1));


        // Create the panel for input
        JPanel inputPanel = new JPanel(new BorderLayout());
        JLabel inputLabel = new JLabel("Enter a string:");
        JTextField inputField = new JTextField();
        inputPanel.add(inputLabel, BorderLayout.WEST);
        inputPanel.add(inputField, BorderLayout.CENTER);


        // Create the panel for result
        JPanel resultPanel = new JPanel(new BorderLayout());
        JLabel resultLabel = new JLabel("Result:");
```

```
JTextField resultField = new JTextField();
resultField.setEditable(false);
resultPanel.add(resultLabel, BorderLayout.WEST);
resultPanel.add(resultField, BorderLayout.CENTER);

// Create the panel for buttons
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout());

// Create the buttons
JButton countVowelButton = new JButton("CountVowel");
JButton resetButton = new JButton("Reset");
JButton exitButton = new JButton("Exit");

// Add action listeners to the buttons
countVowelButton.addActionListener(new ActionListener() {

   public void actionPerformed(ActionEvent e) {
      String inputText = inputField.getText();
      int vowelCount = countVowels(inputText);
      resultField.setText(String.valueOf(vowelCount));
   }
});

resetButton.addActionListener(new ActionListener() {

   public void actionPerformed(ActionEvent e) {
      inputField.setText("");
      resultField.setText("");
   }
```

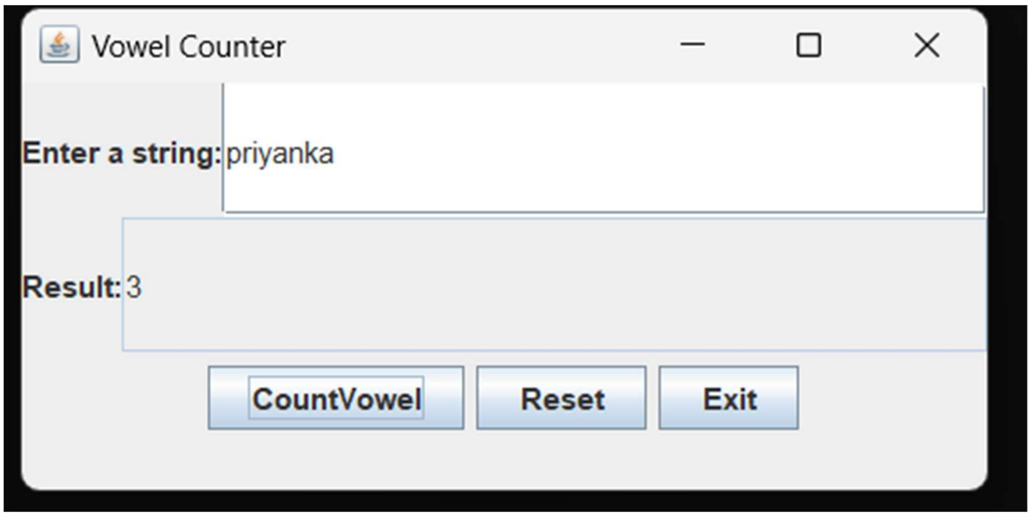
```
        });

        exitButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
                System.exit(0);
            }
        });

        // Add buttons to the panel
        buttonPanel.add(countVowelButton);
        buttonPanel.add(resetButton);
        buttonPanel.add(exitButton);

        // Add components to the frame
        frame.add(inputPanel);
        frame.add(resultPanel);
        frame.add(buttonPanel);

        // Make the frame visible
        frame.setVisible(true);
    }

    private static int countVowels(String input) {
        int count = 0;
        String vowels = "aeiouAEIOU";
        for (int i = 0; i < input.length(); i++) {
            if (vowels.indexOf(input.charAt(i)) != -1) {
                count++;
            }
        }
        return count;
    }
}
```

Output:





```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac VowelCounterApp.java
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java VowelCounterApp
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```



Enter a string: priyanka

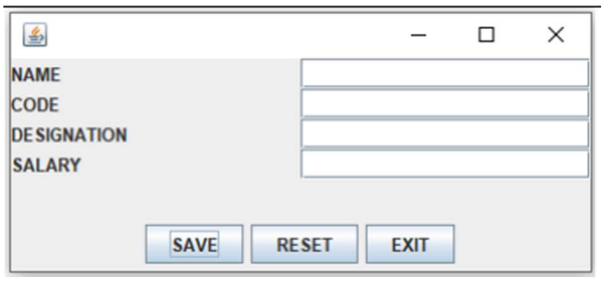
Result: 3

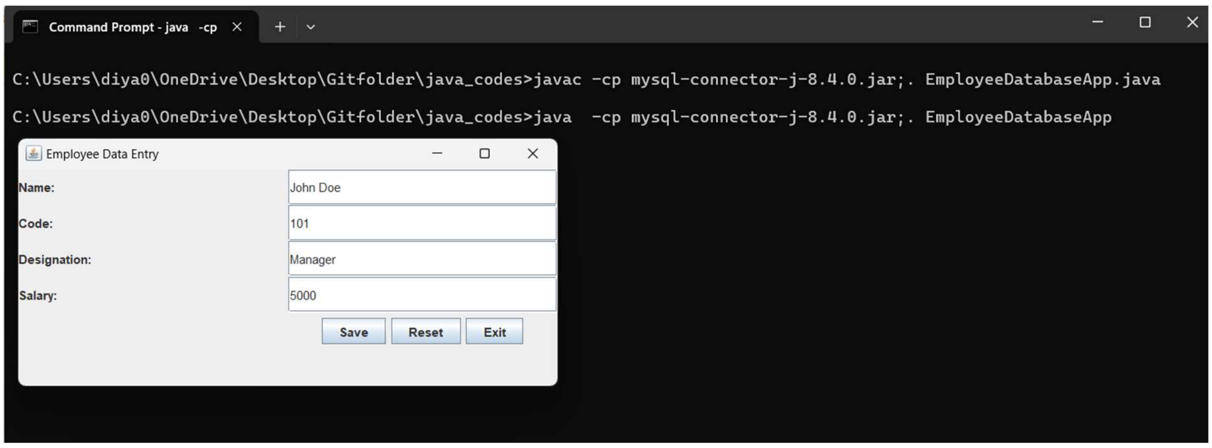
```
/*Q40.Java Database Connectivity (JDBC):

Create a database of employee with the following fields.

• Name

• Code

• Designation

• Salary

a) Write a java program to create GUI java application to take employee data from the
TextFields and store it in database using JDBC connectivity.

*/


import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.sql.*;


public class EmployeeDatabaseApp extends JFrame implements ActionListener {

    private JLabel nameLabel, codeLabel, designationLabel, salaryLabel;

    private JTextField nameField, codeField, designationField, salaryField;

    private JButton saveButton, resetButton, exitButton;


    // JDBC URL, username, and password of MySQL server

    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/employeedb";

    private static final String USERNAME = "root";

    private static final String PASSWORD = "your_sql_password";


    public EmployeeDatabaseApp() {

        setTitle("Employee Data Entry");

        setSize(400, 250);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


        // Initialize components
```

```
nameLabel = new JLabel("Name:");
codeLabel = new JLabel("Code:");
designationLabel = new JLabel("Designation:");
salaryLabel = new JLabel("Salary:");

nameField = new JTextField(20);
codeField = new JTextField(10);
designationField = new JTextField(20);
salaryField = new JTextField(10);

saveButton = new JButton("Save");
resetButton = new JButton("Reset");
exitButton = new JButton("Exit");

// Add action listeners to buttons
saveButton.addActionListener(this);
resetButton.addActionListener(this);
exitButton.addActionListener(this);

// Create panel for buttons
JPanel buttonPanel = new JPanel(new FlowLayout());
buttonPanel.add(saveButton);
buttonPanel.add(resetButton);
buttonPanel.add(exitButton);

// Set layout
setLayout(new GridLayout(6, 2));

// Add components to the frame
add(nameLabel);
```

```
        add(nameField);
        add(codeLabel);
        add(codeField);
        add(designationLabel);
        add(designationField);
        add(salaryLabel);
        add(salaryField);
        add(new JLabel()); // Placeholder for empty cell
        add(buttonPanel); // Add button panel

        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == saveButton) {
            saveEmployeeData();
        } else if (e.getSource() == resetButton) {
            resetFields();
        } else if (e.getSource() == exitButton) {
            System.exit(0);
        }
    }

    private void saveEmployeeData() {
        String name = nameField.getText();
        int code = Integer.parseInt(codeField.getText());
        String designation = designationField.getText();
        double salary = Double.parseDouble(salaryField.getText());

        try {
            // Register MySQL JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Open a connection
            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/employeedb",
"root","your_sql_password");

            // Create a prepared statement
            String query = "INSERT INTO employee (NAME, CODE, DESIGNATION,
SALARY) VALUES (?, ?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(query);
            statement.setString(1, name);
            statement.setInt(2, code);
            statement.setString(3, designation);
            statement.setDouble(4, salary);

            // Execute the statement
            int rowsInserted = statement.executeUpdate();
            if (rowsInserted > 0) {
                JOptionPane.showMessageDialog(this, "Employee data saved successfully!");
            } else {
                JOptionPane.showMessageDialog(this, "Failed to save employee data.");
            }

            // Close resources
            statement.close();
            connection.close();
        } catch (Exception ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());
        }
    }

    private void resetFields() {
        nameField.setText("");
        codeField.setText("");
        designationField.setText("");
        salaryField.setText("");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new EmployeeDatabaseApp());
    }
}
```

Output:





```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac -cp mysql-connector-j-8.4.0.jar;. EmployeeDatabaseApp.java
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java  -cp mysql-connector-j-8.4.0.jar;. EmployeeDatabaseApp
```

Employee Data Entry

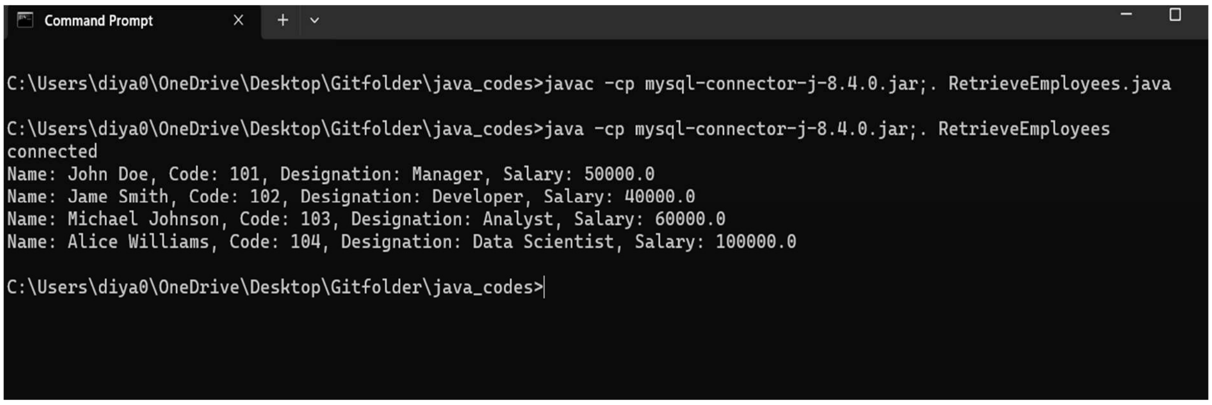
Name: John Doe

Code: 101

Designation: Manager

Salary: 5000

```java
/*Q40.b) Write a JDBC Program to retrieve all the records from the employee database.*/

import java.sql.*;

public class RetrieveEmployees {

   public static void main(String[] args) {

      try {

         Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/employeedb", "root",
"your_sql_password");

         System.out.println("connected");

         Statement statement = connection.createStatement();

         ResultSet resultSet = statement.executeQuery("SELECT * FROM employee");

         while (resultSet.next()) {

            String name = resultSet.getString("name");

            String code = resultSet.getString("code");

            String designation = resultSet.getString("designation");

            double salary = resultSet.getDouble("salary");

            System.out.println("Name: " + name + ", Code: " + code + ", Designation: " + designation + ",
Salary: " + salary);

         }

         connection.close();

      } catch (SQLException e) {

         e.printStackTrace();

      }

   }

}
```



```
C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>javac -cp mysql-connector-j-8.4.0.jar;. RetrieveEmployees.java

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>java -cp mysql-connector-j-8.4.0.jar;. RetrieveEmployees
connected
Name: John Doe, Code: 101, Designation: Manager, Salary: 50000.0
Name: Jame Smith, Code: 102, Designation: Developer, Salary: 40000.0
Name: Michael Johnson, Code: 103, Designation: Analyst, Salary: 60000.0
Name: Alice Williams, Code: 104, Designation: Data Scientist, Salary: 100000.0

C:\Users\diya0\OneDrive\Desktop\Gitfolder\java_codes>
```