RAJALAKSHMI ENGINEERING COLLEGE

[AUTONOMOUS]

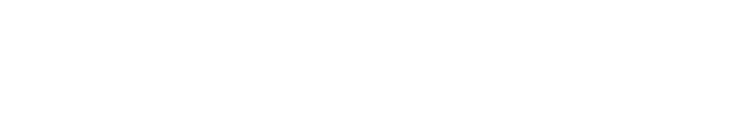
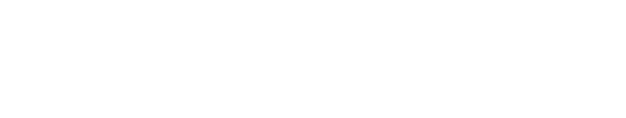
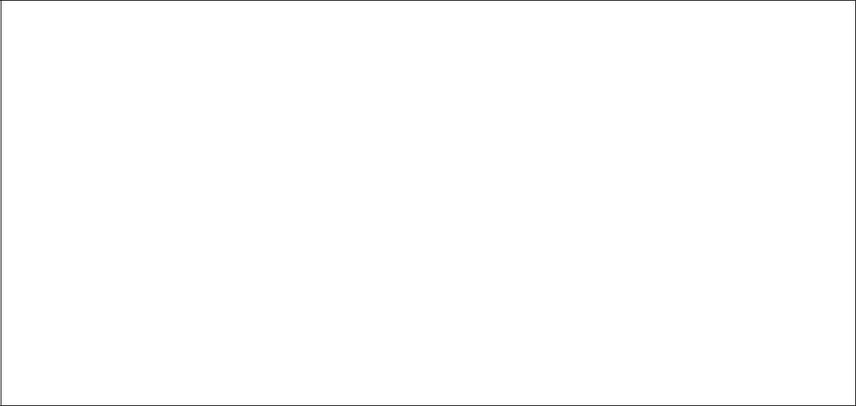
RAJALAKSHMI NAGAR, THANDALAM – 602 105





CS23333 OBJECT ORIENTED PROGRAMING USING JAVA

Laboratory Record Note Book



.

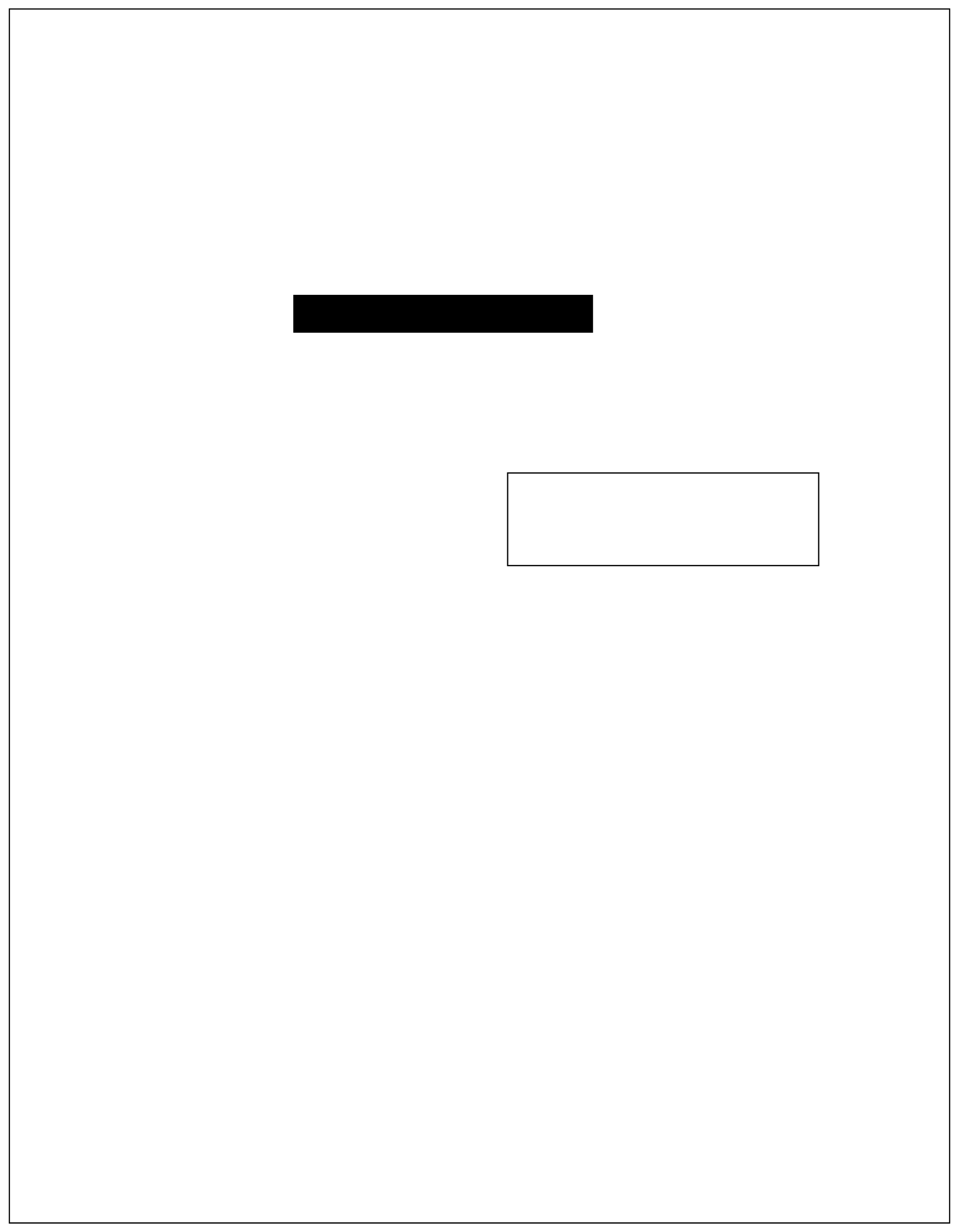
Name : HARISHA.D

Year / Branch / Section : II/IT/A

College Roll No : 2116231001058

Semester : III

Academic Year : 2024-2025



RAJALAKSHMI ENGINEERING COLLEGE

**[**

**AUTONOMOUS**

**]**

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name : HARISHA.D

Submitted for the Practical Examination held on . . . . . . . . . . . . .

Certified that this is the bonafide record of work done by the above

student in the CS23333 –Object Oriented Programming using JAVA

during the year 2024 - 2025.

2116231001058

Academic Year : 2024-2025

Semester: 3rd sem

Branch : IT - A

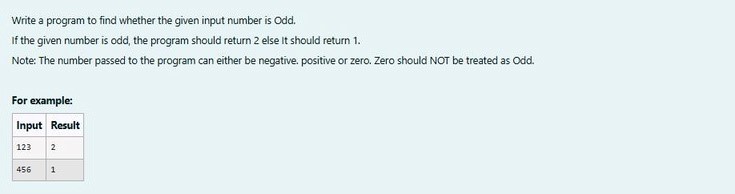
Register No :

Internal Examiner

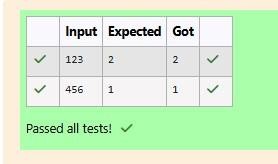
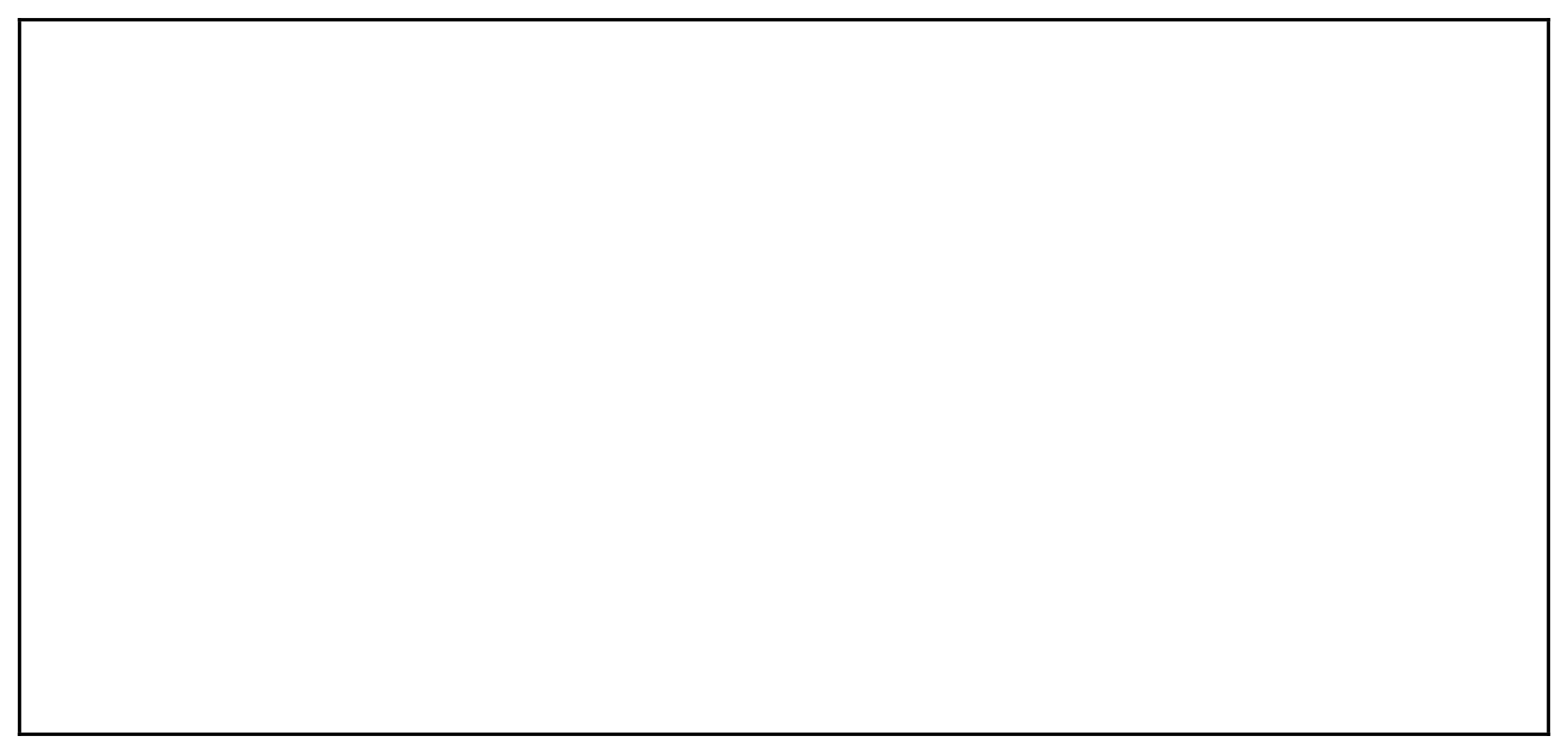
External Examiner

Signature of Faculty in-charge

1.



**SOLUTION :** import



java.util.Scanner;

public

class oddorEven{ public static void

main(String[]args){

Scanner

s=new

Scanner(System.in);

int

number

=

s.nextInt(); if(number %2==0){

System.out.println(1);

}

else

{

System.out.println(2);

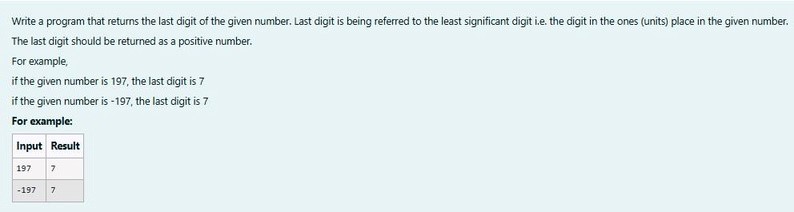
}

}

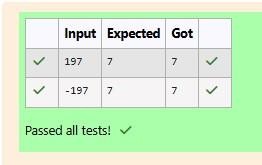
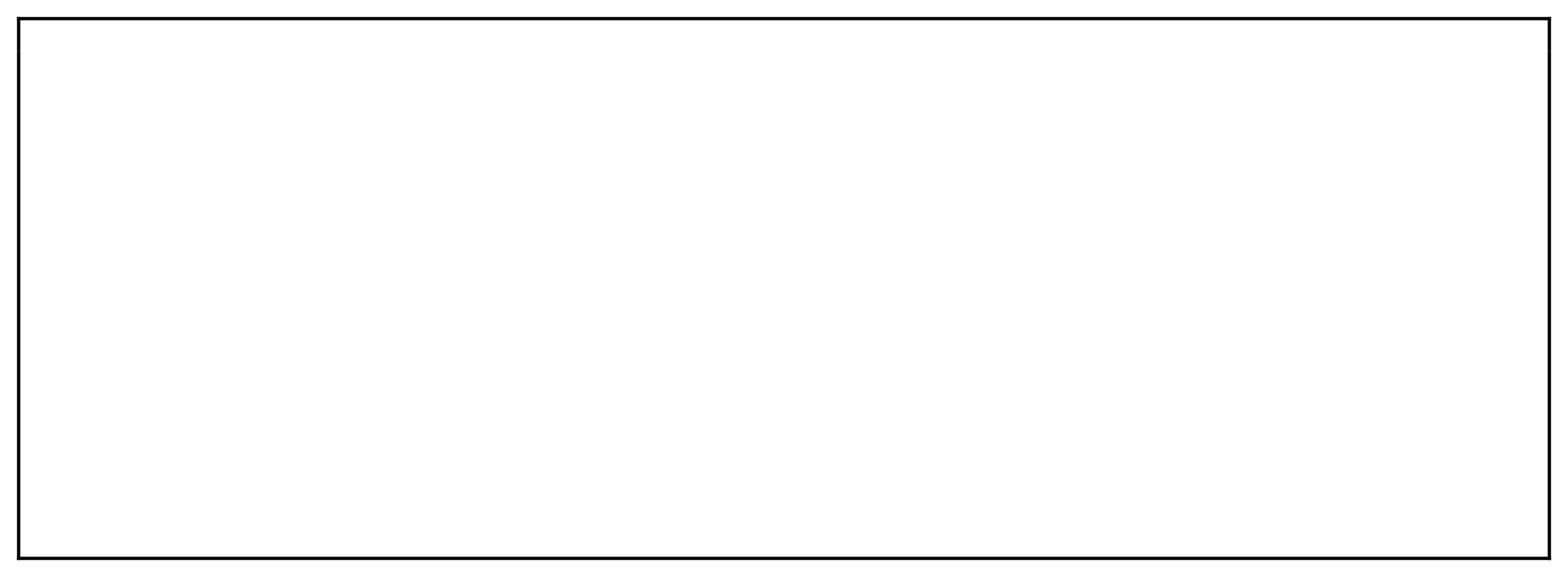
}

**OUTPUT :**

**2.**



**SOLUTION :**



import java.util.Scanner; import

java.lang.Math; public class LastDigit{

public static void main(String[]args){

Scanner s=new Scanner(System.in);

int a = s.nextInt(); int

lastDigit=Math.abs(a%10);

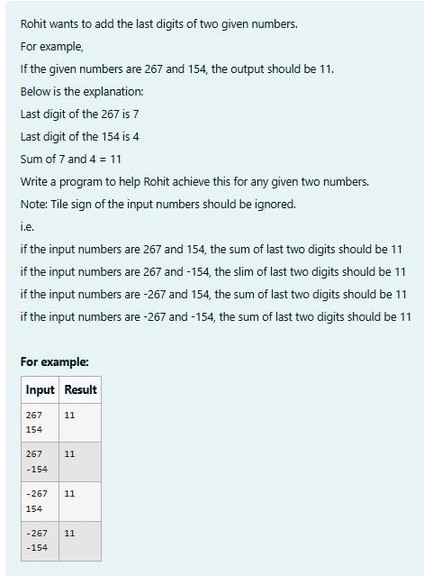
System.out.println(lastDigit);

}

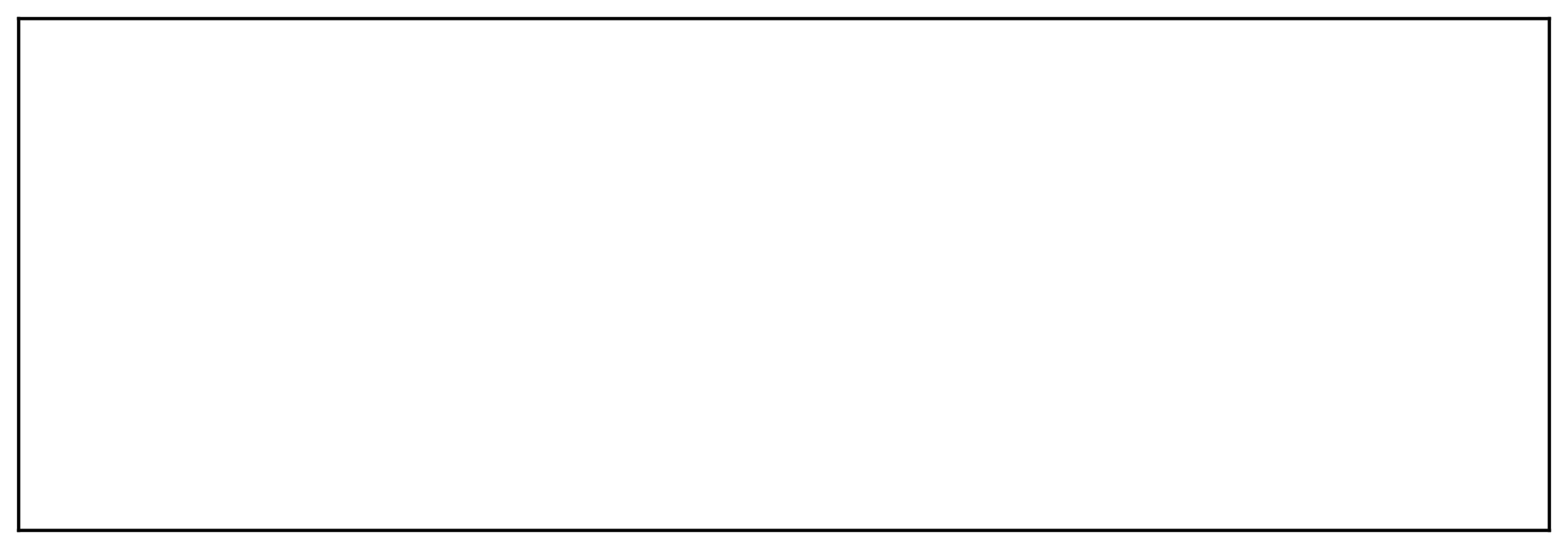
}

**OUTPUT :**

**3.**



**SOLUTION :**



import java.util.Scanner;

import java.lang.Math;

public class number{ public static void

main(String[]args){ Scanner s= new

Scanner(System.in);

int a = s.nextInt();

int b = s.nextInt();

System.out.println(Math.abs(a)%10+Math.abs(b)%10);

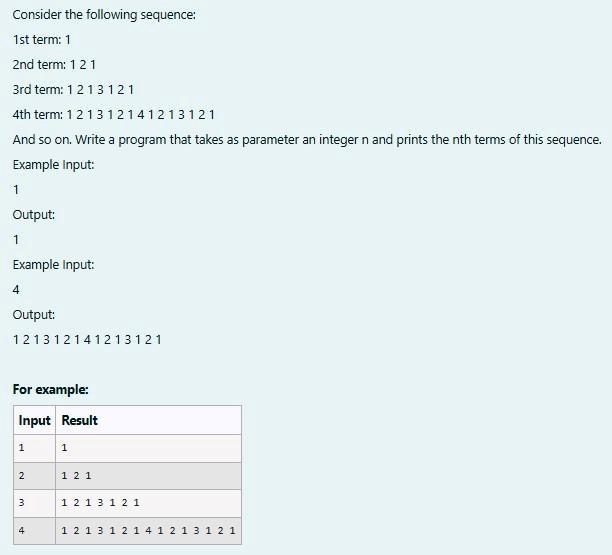
}

}

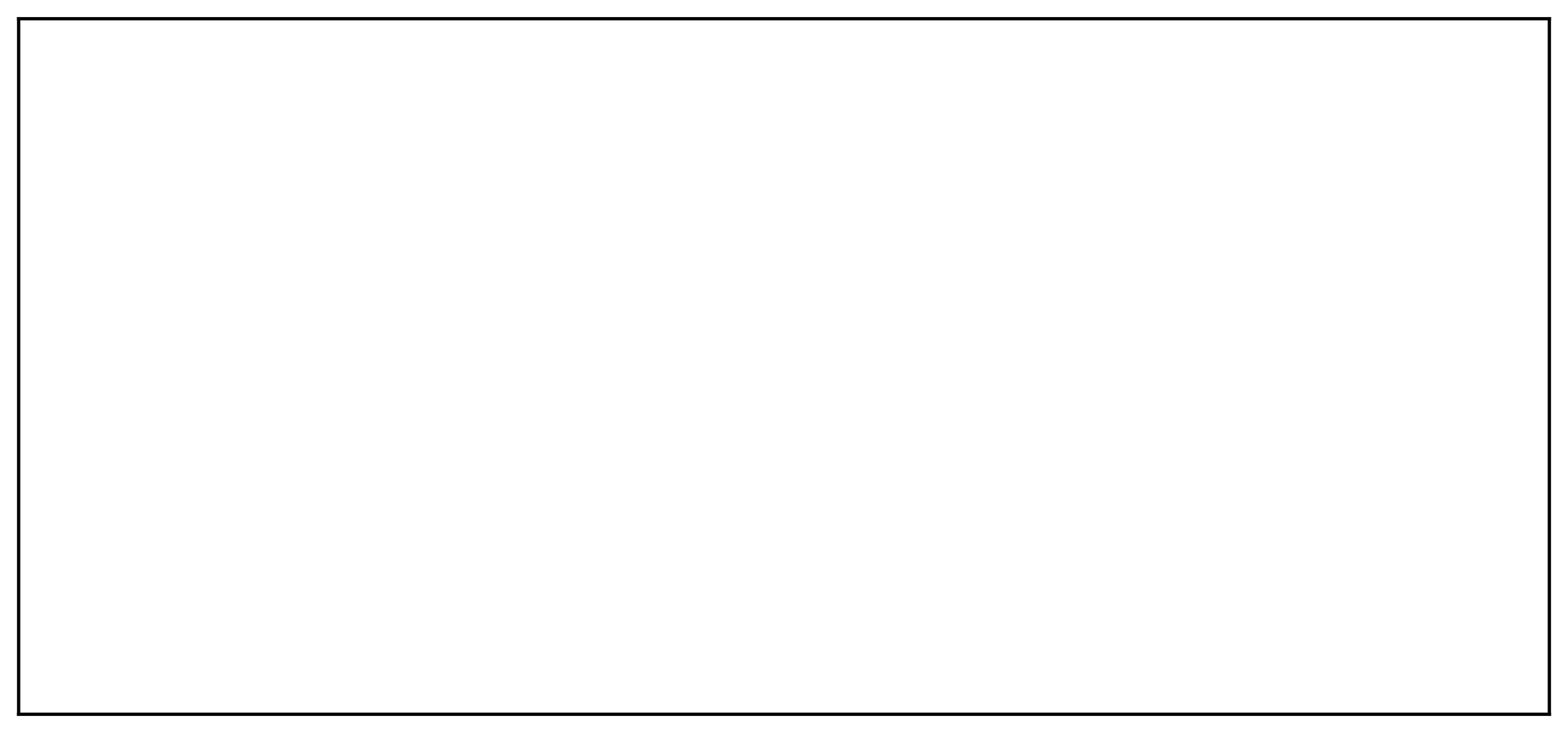
**OUTPUT:**

[**Lab-02-Flow Control Statements**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=50)

**1.**



**SOLUTION :**



import java.util.Scanner; public class

SequenceGenerator{ public static void

main(String[]args){ Scanner S = new

Scanner(System.in);

int n = S.nextInt();

String term = generateTerm(n);

System.out.print(term);

}

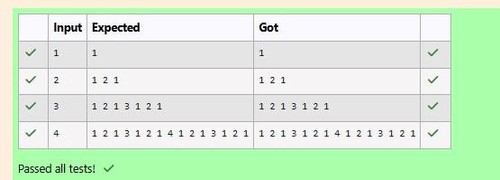
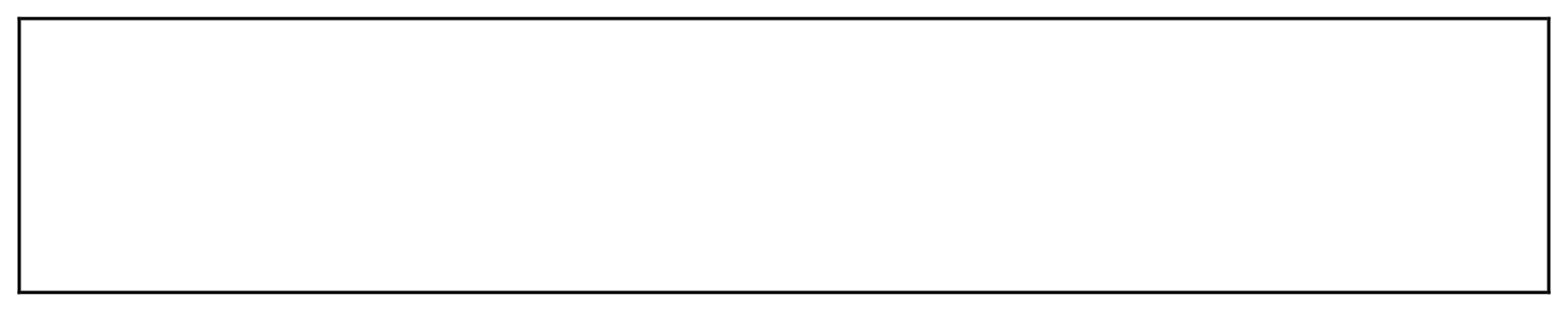
private static String generateTerm(int n){

if (n==1){ return "1";

}

String prevTerm = generateTerm (n-1);

StringBuilder currentTerm = new StringBuilder(prevTerm);



currentTerm.append(" " + n + " ");

currentTerm.append(prevTerm);

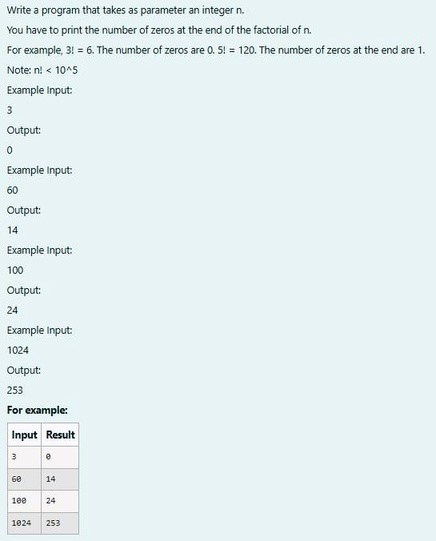
return currentTerm.toString();

}

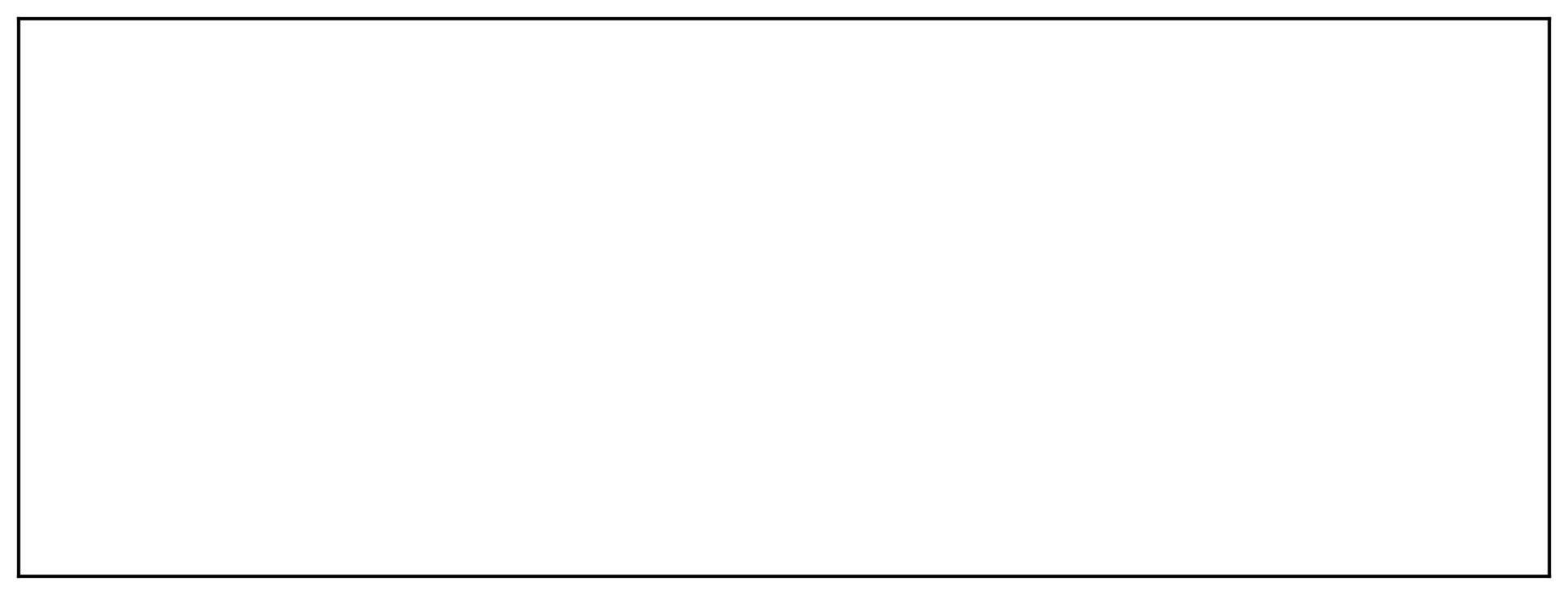
}

**OUTPUT :**

**2.**



**SOLUTION :**



// Java program to count trailing 0s in n!

import java.io.\*; import

java.util.Scanner;

class prog {

// Function to return trailing

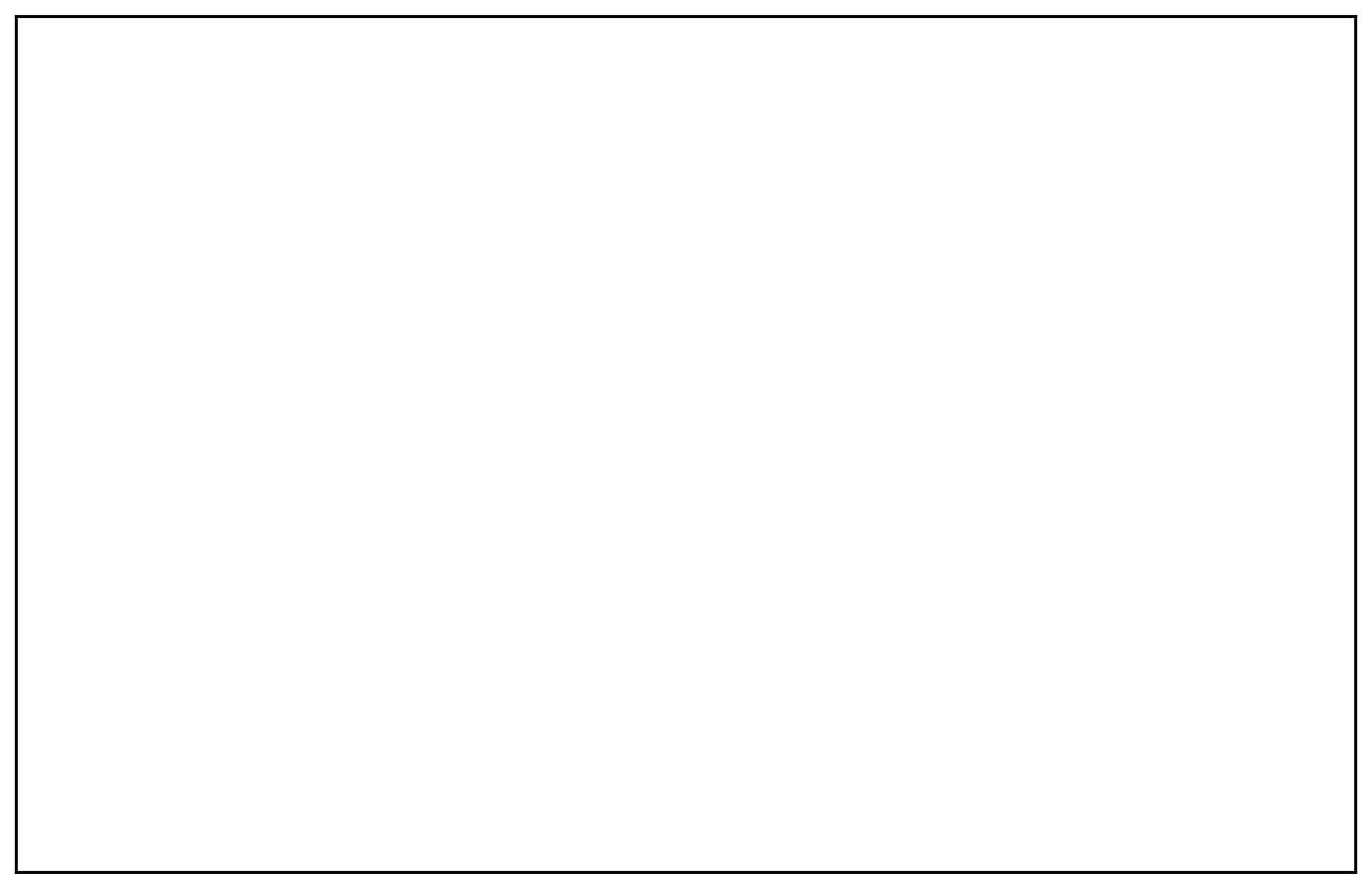
// 0s in factorial of n

static int findTrailingZeros(int n)

{

if (n < 0) // Negative Number Edge Case

return -1;



// Initialize result

int count=0;

// Keep dividing n by powers //

of 5 and update count for (int i =

5

; n / i >= 1; i\*

=5

+=

n / i;

){

count

}

return count;

}

// Driver Code

public static void main(String[] args)

{

Scanner sc= new Scanner(System.in);

int n=sc.nextInt();

int res=findTrailingZeros(n);

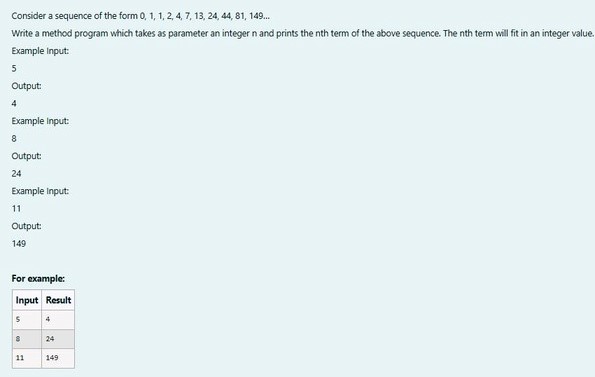
System.out.println(res);

}

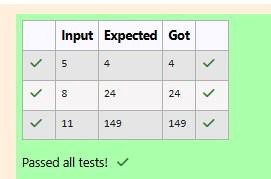
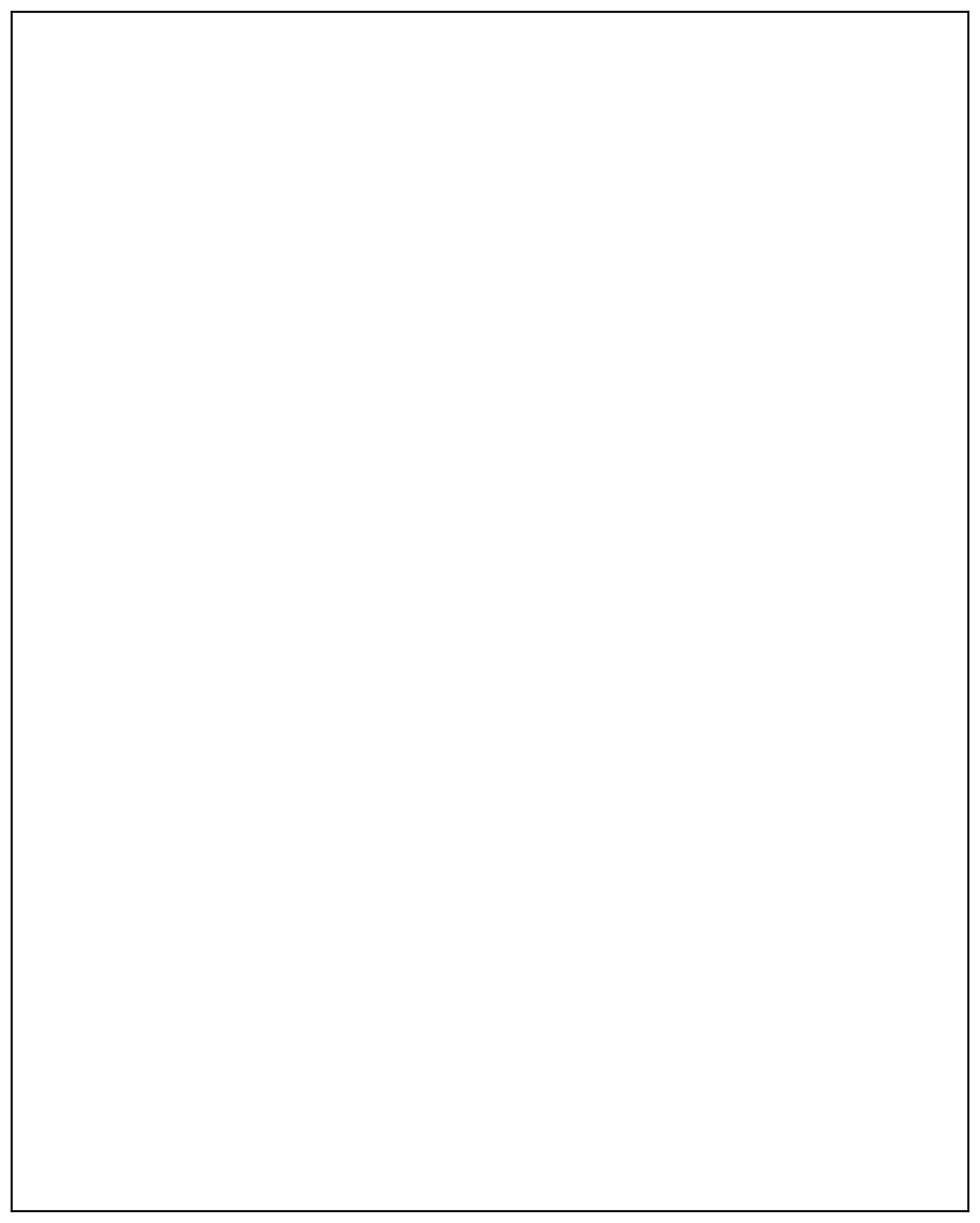
}

**OUTPUT :**

**3.**



**SOLUTION :**



import java.util.Scanner;

class fibo3{ int a; int b;

int c;

fibo3(int a,int b,int c){

this.a = a; this.b =

b; this.c = c;

}

int nth(int x){

if (x == 1){

return 0;

}

else if(x == 2 && x == 3)

return 1;

else{ int temp1,temp2,temp; int

count = 4; while(x >=

count){ temp =

this.a+this.b+this.c;

temp1 = this.c;

this.c = temp;

temp2 = this.b;

this.b = temp1;

this.a = temp2;

count++;

}

return this.c;

}

}

}

public class Main{ public static void

main(String[] args){ Scanner s = new

Scanner(System.in);

int t = s.nextInt(); fibo3 r

=

new fibo3(0,1,1);

System.out.print(r.nth(t));

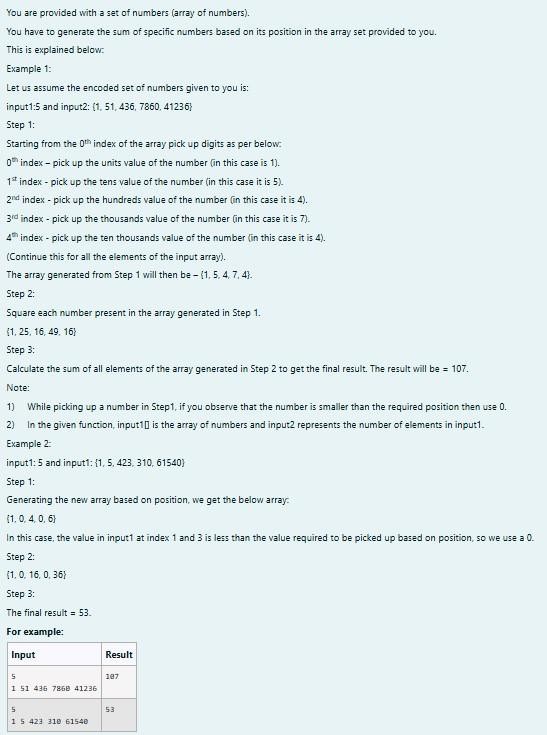
}

}

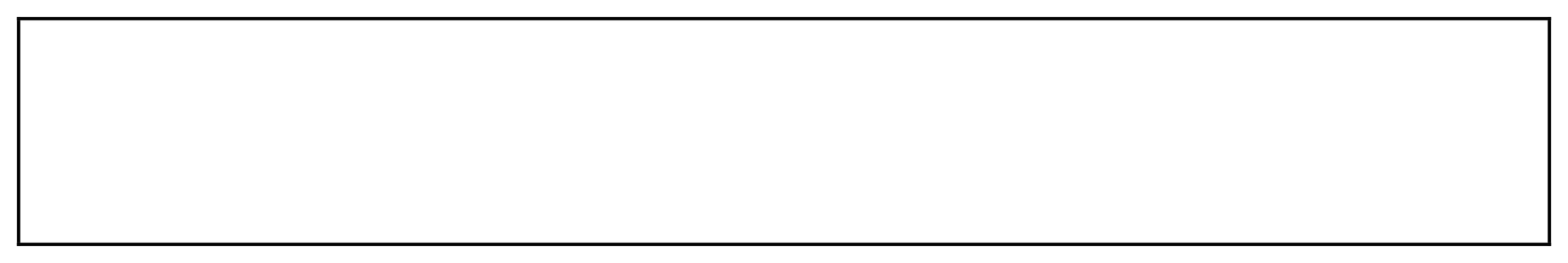
**OUTPUT :**

[**Lab-03-Arrays**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=51)

**1.**



**SOLUTION :**

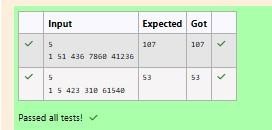
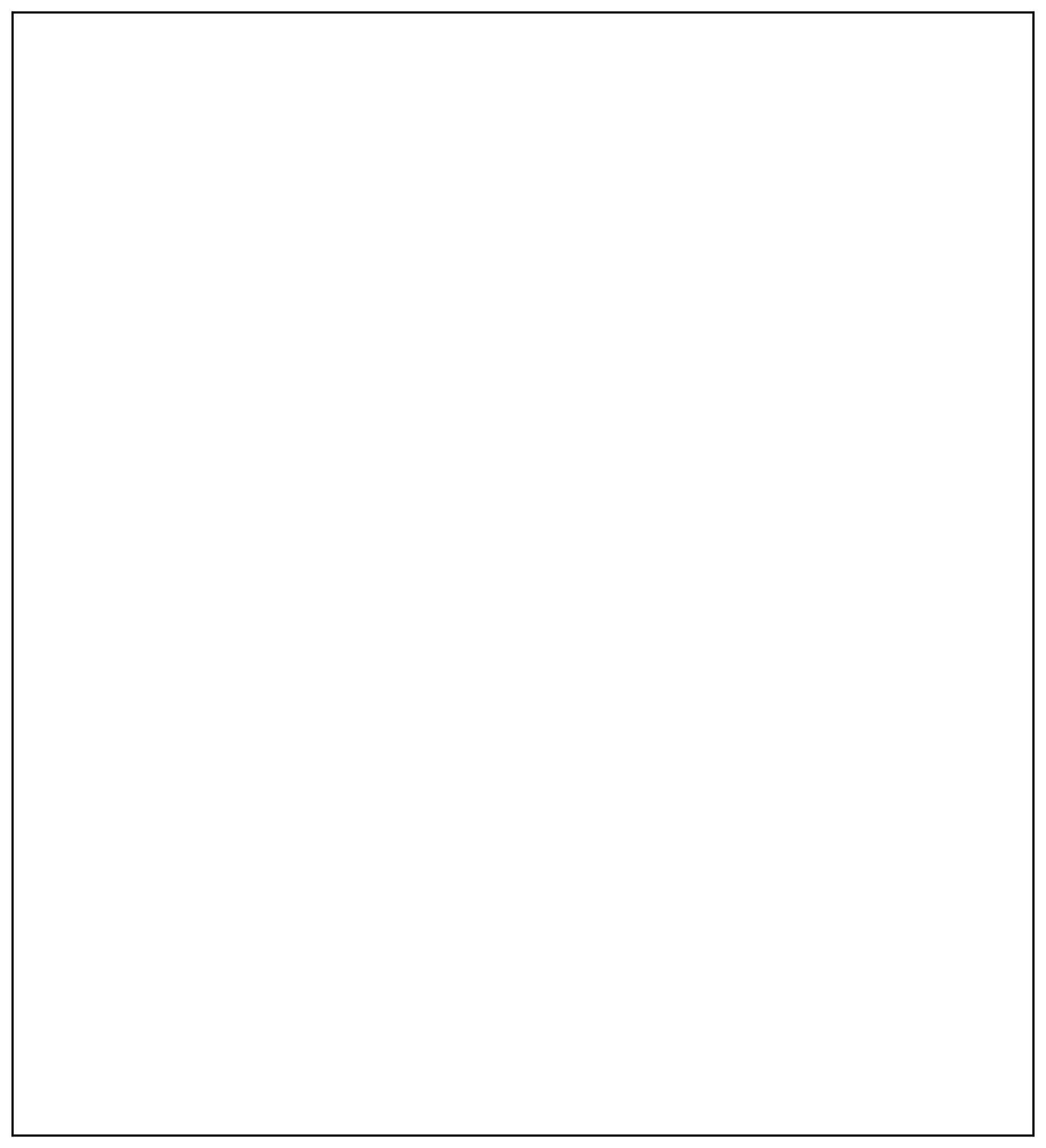


import java.util.Scanner; public class

digit{ public static void

main(String[]args){

Scanner scanner =new Scanner(System.in);



int size =scanner.nextInt();

int[]inpar=new int[size];

for(int i=0;i<size;i++){

inpar[i]=scanner.nextInt();

}

int[]dig=new int[size];

for(int i=0;i<size;i++){

int num=inpar[i];

if(i==0){

dig[i]=num%10;

}

else if (i==1){

dig[i]=(num/10)%10;

}

else if(i==2){

dig[i]=(num/100)%10;

}

else if(i==3){

dig[i]=(num/1000)%10;

}

else if(i==4){

dig[i]=(num/10000)%10;

}

else

{

dig[i]=0;

}

}

int fin=0;

for(int digi:dig){

fin+=digi\*digi;

}

System.out.print(fin);

}

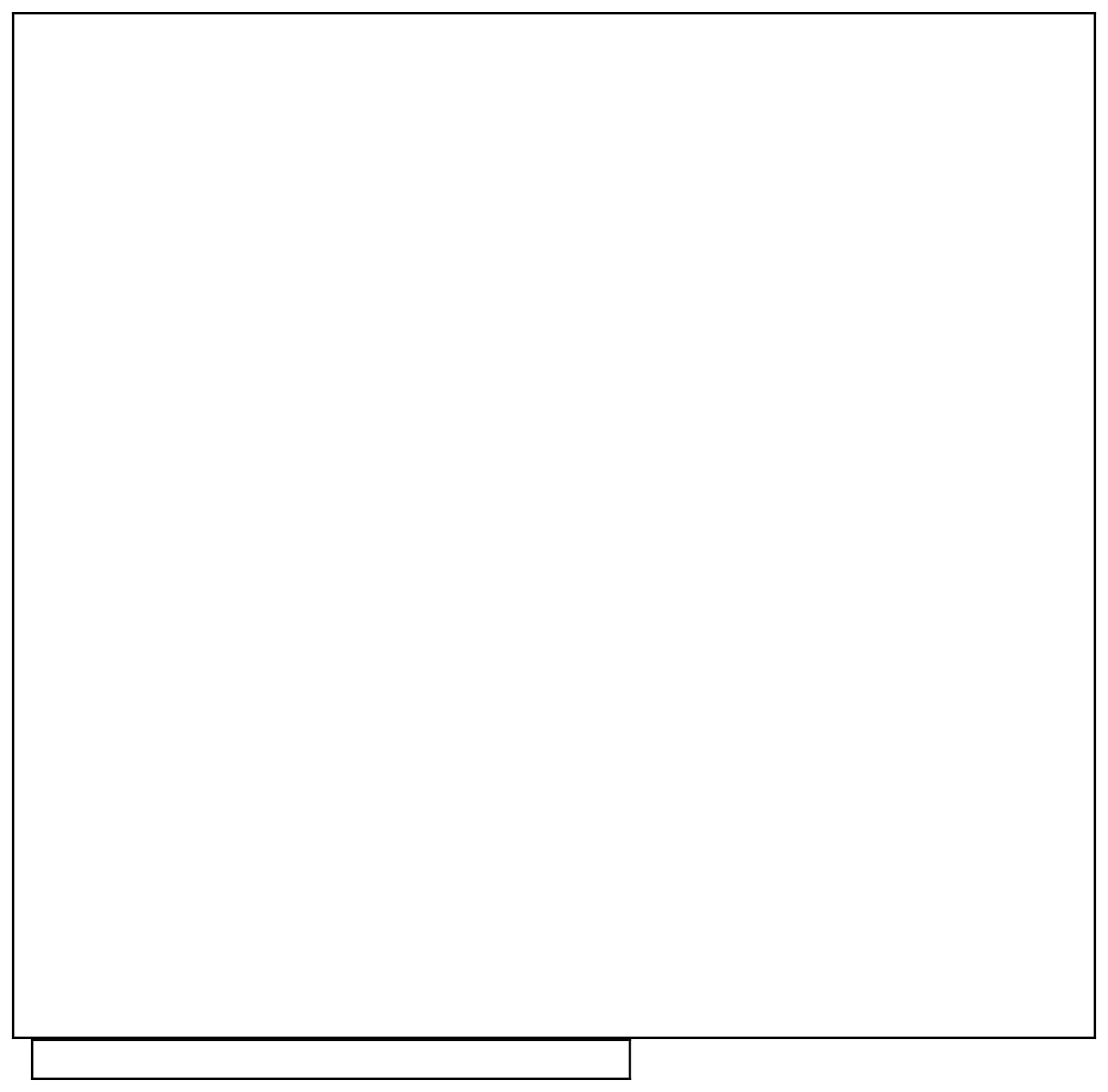
}

**OUTPUT :**

**2.**



**SOLUTION :**



}

import java.util.Scanner; public class

longdig{ public static void

main(String[]args){ Scanner sc=new

Scanner(System.in);

int n=sc.nextInt();

int c = 1,v,seqtemp = 0,seq = 0,countmax = 0;

int count = 0; while(c <= n){ v = sc.nextInt();

if(v >= 0){ countmax= countmax + v;

seqtemp++;

}

else{

seqtemp = 0;

countmax = 0;

}

if(seqtemp > seq ){

seq = seqtemp;

count = countmax;

}

else if (seq == seqtemp){

count = count + countmax;

}

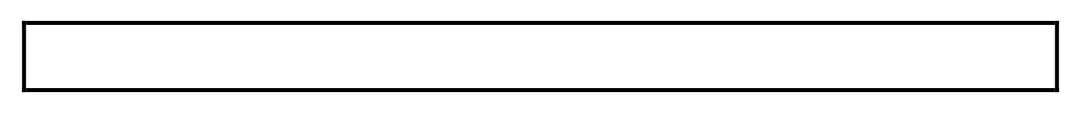
c++; }

if (count == 0)

System.out.print(-1);

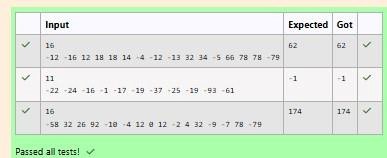
else

System.out.print(count);

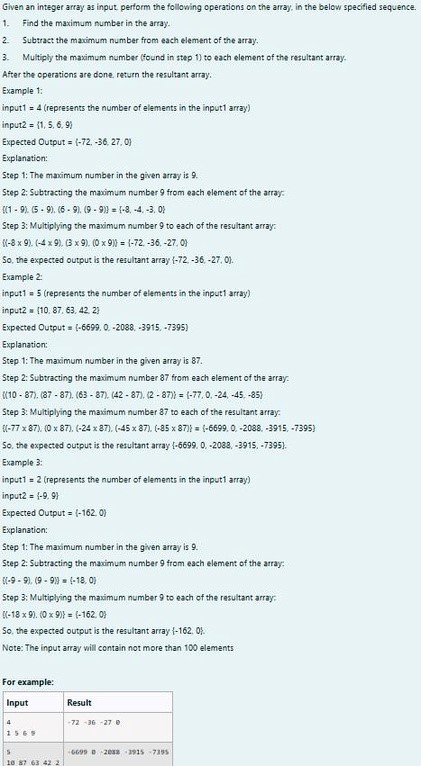


}

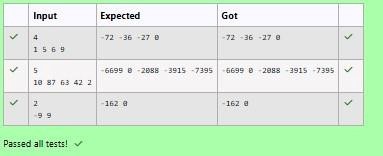
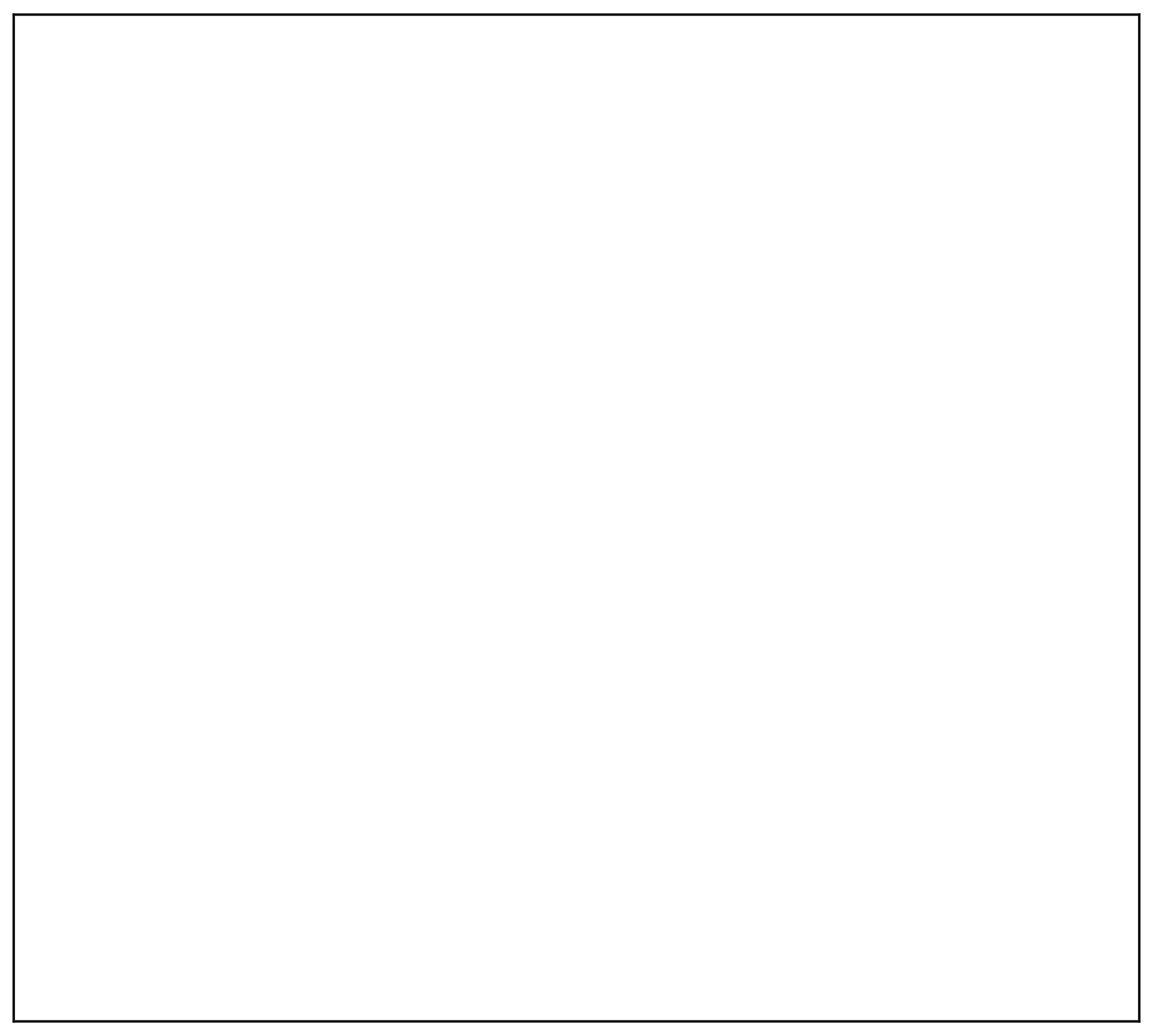
**OUTPUT :**



**3.**



**SOLUTION :**



import java.util.Scanner; public

class res{ public static

int[]pa(int[]arr){

int maxs=Integer.MIN\_VALUE;

for (int num:arr){

if(num>maxs){

maxs=num;

}

}

for(int i=0;i<arr.length;i++){ arr[i]=(arr[i]-

maxs)\*maxs;

}

return arr;

}

public static void main(String[]args){

Scanner scanner =new Scanner (System.in);

int n=scanner.nextInt();

int[]arr=new int[n]; for(int

i=0;i<n;i++){

arr[i]=scanner.nextInt();

}

int[]res=pa(arr);

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

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

}

scanner.close();

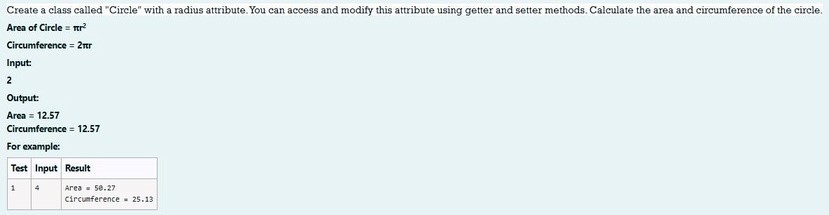
}

}

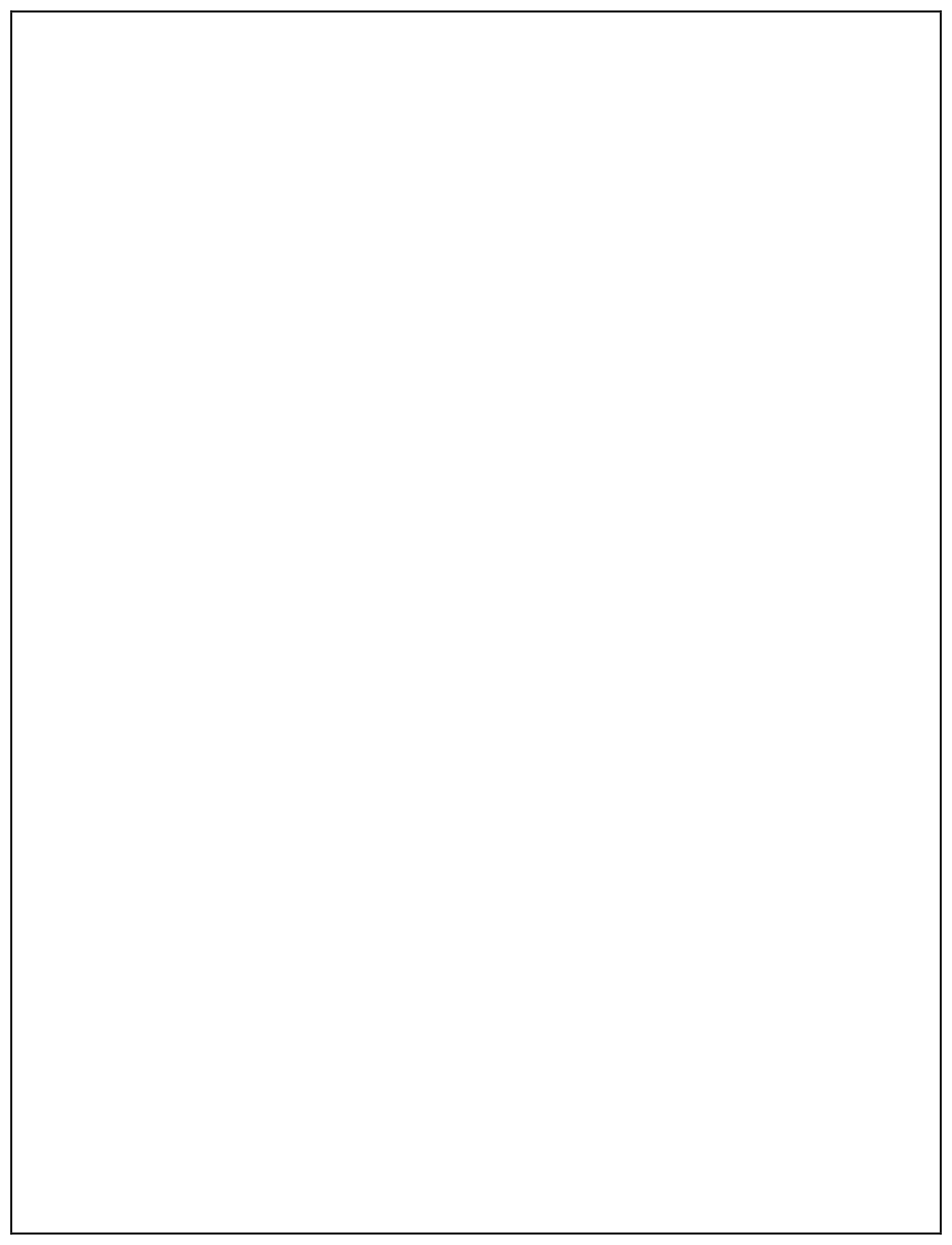
**OUTPUT :**

[**Lab-04-Classes and Objects**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=52)

**1.**



**SOLUTION :**



import java.io.\*; import

java.util.Scanner; class

Circle

{

private double radius; public

Circle(double radius){

// set the instance variable radius

this.radius =radius;

}

public void setRadius(double

radius){

// set the radius

this.radius=radius;

}

public double getRadius()

{

// return the radius

return radius;

}

public double calculateArea() { // complete the below statement

return Math.PI\*radius\*radius;

}

public double calculateCircumference()

{

// complete the statement return

2

\*Math.PI\*radius;

}

}

class prog{ public static void

main(String[] args) { int r;

Scanner sc= new Scanner(System.in);

r=sc.nextInt();

Circle c= new Circle(r);

System.out.println("Area = "+String.format("%.2f",

c.calculateArea()));

// invoke the calculatecircumference method

System.out.println("Circumference = "+String.format("%.2f" ,

c.calculateCircumference()));

sc.close();

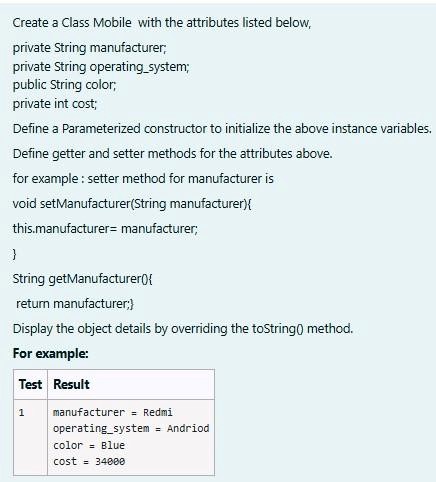
}

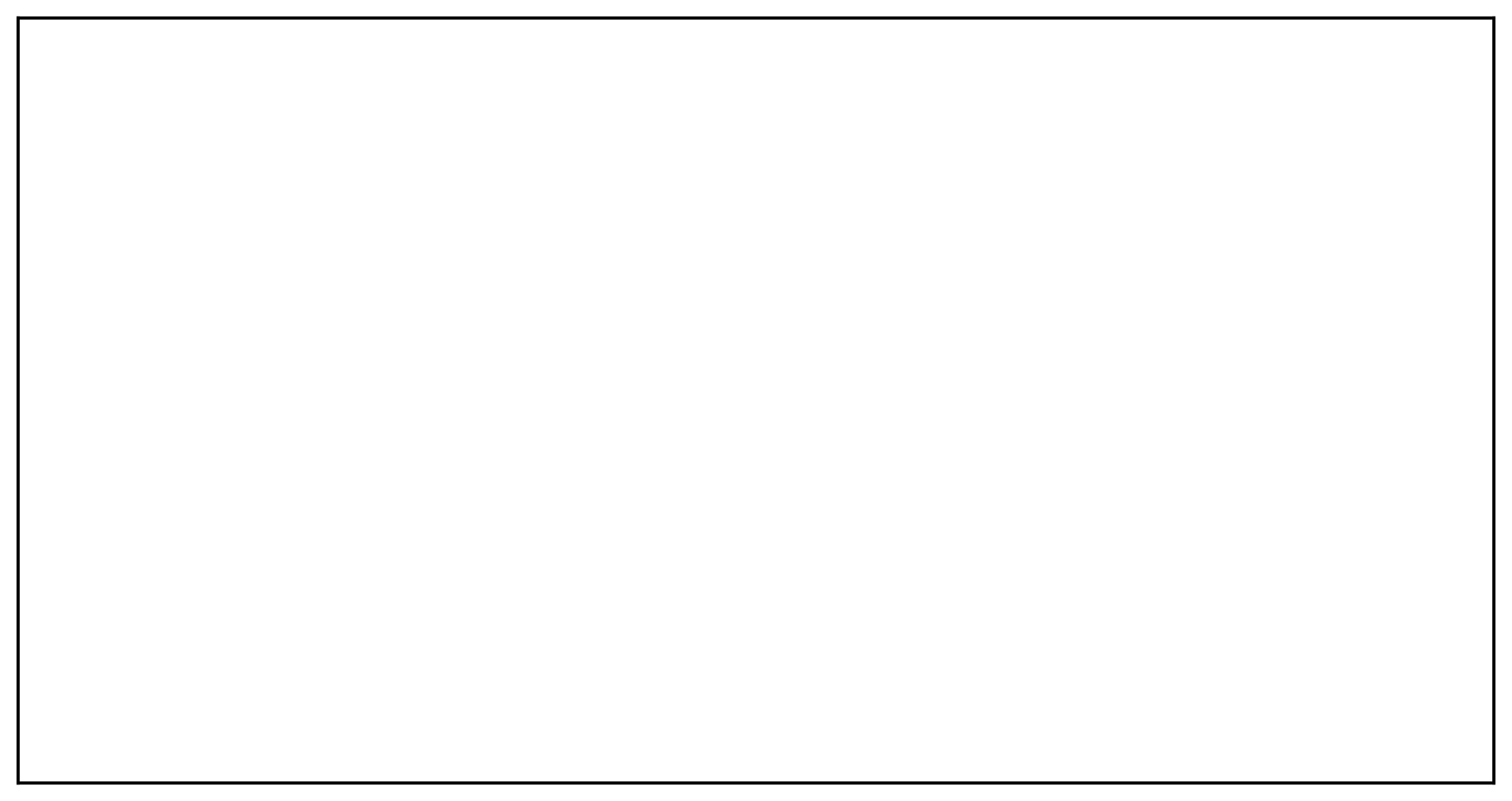
}

**OUTPUT :**



**2.**





**SOLUTION :**

public class mobile{

private String man;

private String os;

public String clr;

private int cost;

public mobile(String man,String os,String clr,int cost){

this.man=man; this.os=os; this.clr=clr;

this.cost=cost;

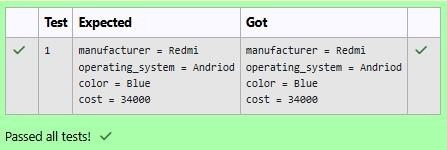
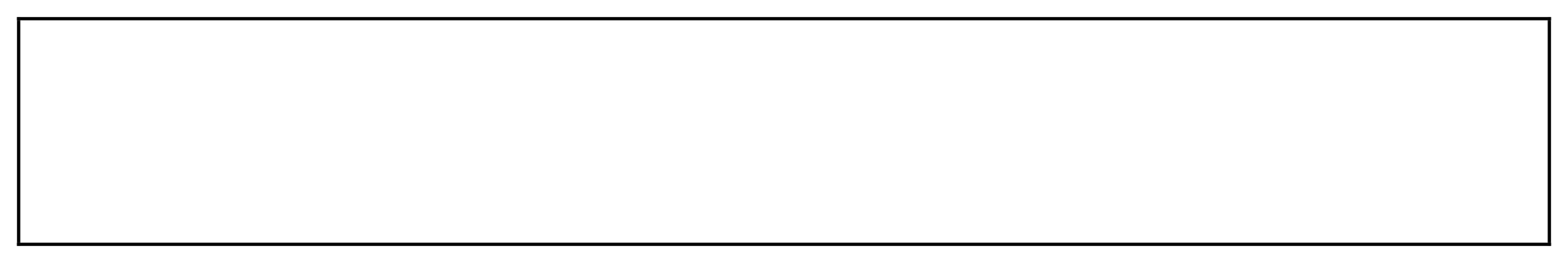
}

public String toString(){ return "manufacturer = "+man+"\n"+"operating\_system =

"+os+"\n"+"color = "+ clr+"\n"+"cost = "+cost;

}

public static void main(String[]args){



mobile mobile=new mobile("Redmi","Andriod","Blue",34000);

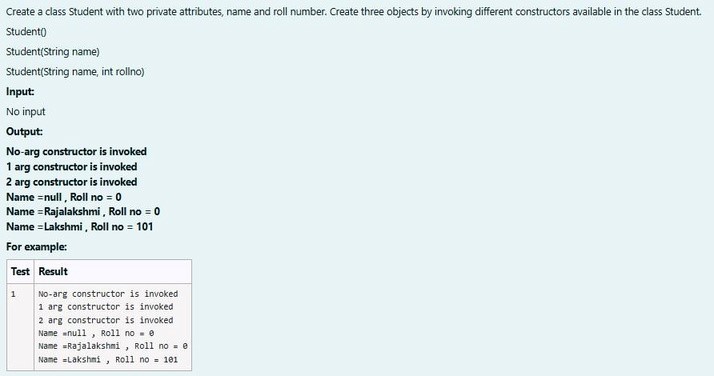
System.out.println(mobile);

}

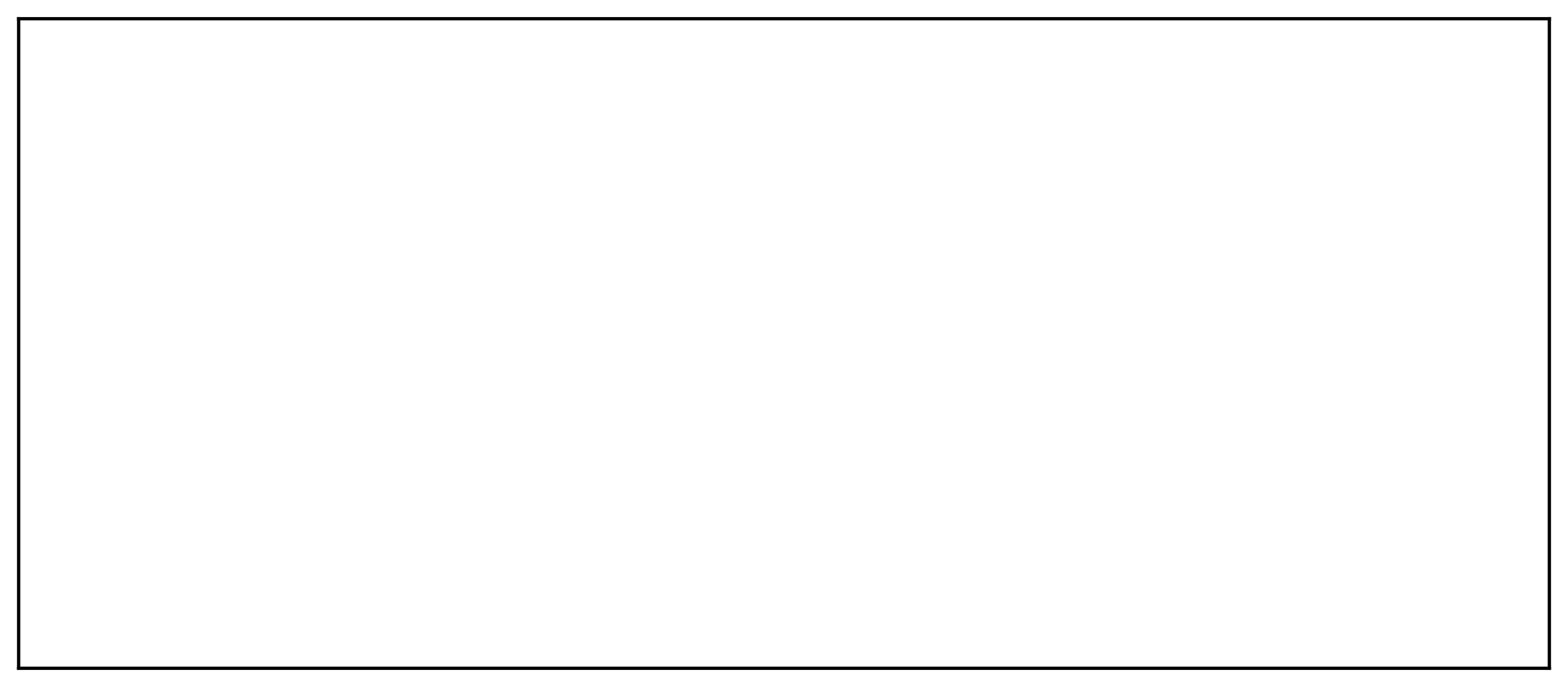
}

**OUTPUT :**

**3.**



**SOLUTION :**



public class stud{ private String name; private int roll;

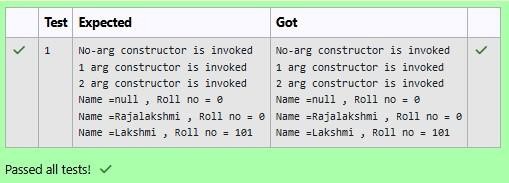
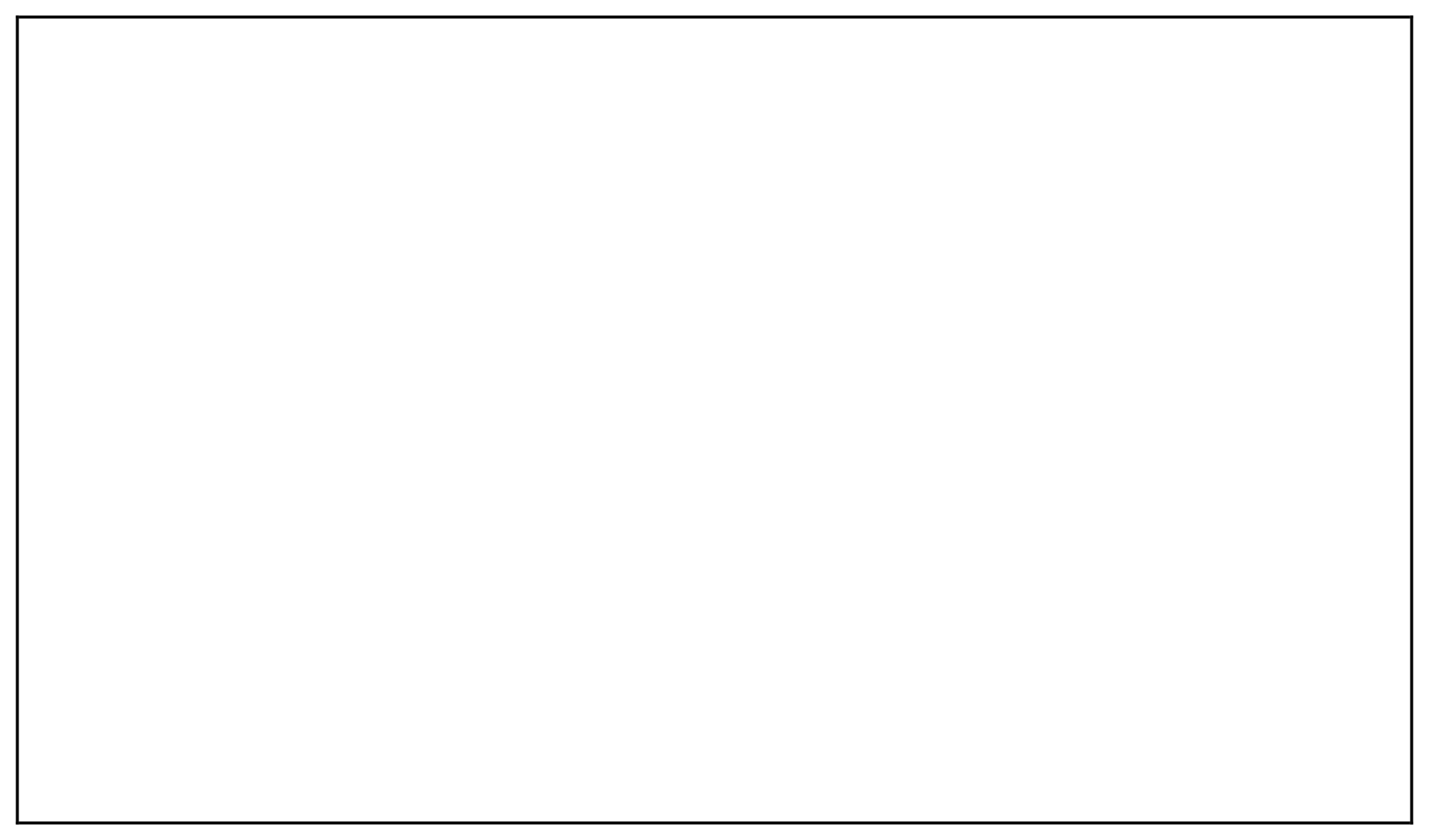
public stud(){

System.out.println("No-arg constructor is invoked"); name=null; roll=0;

}

public stud(String name){

System.out.println("1 arg constructor is invoked"); this.name=name; roll=0;



}

public stud(String name,int roll){

System.out.println("2 arg constructor is invoked"); this.name=name;

this.roll=roll;

}

public static void main (String[]args){

stud s1=new stud(); stud s2=new

stud("Rajalakshmi"); stud s3=new

stud("Lakshmi",101);

System.out.println("Name ="+s1.name+" , Roll no = "+s2.roll);

System.out.println("Name ="+s2.name+" , Roll no = "+s2.roll);

System.out.println("Name ="+s3.name+" , Roll no = "+s3.roll);

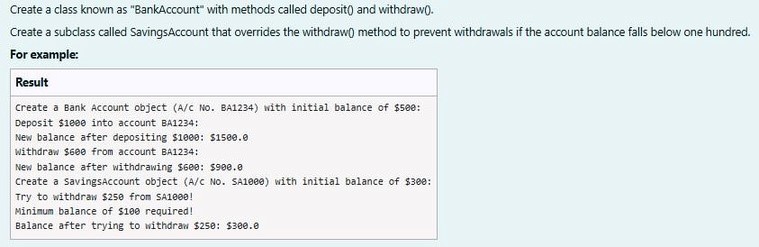
}

}

**OUTPUT :**

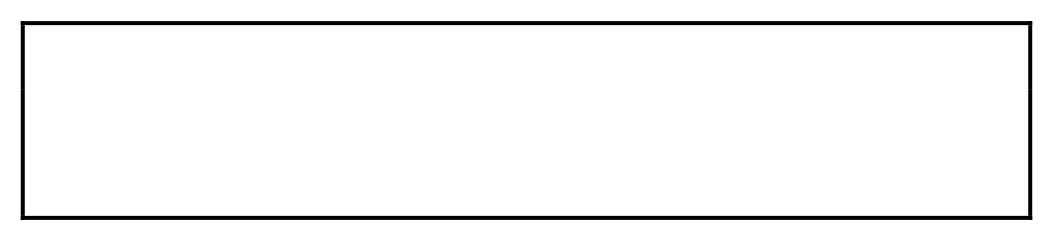
[**Lab-05-Inheritance**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=55)

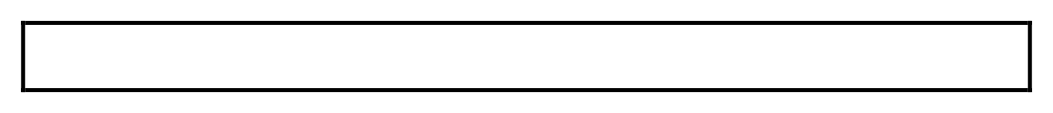
**1.**

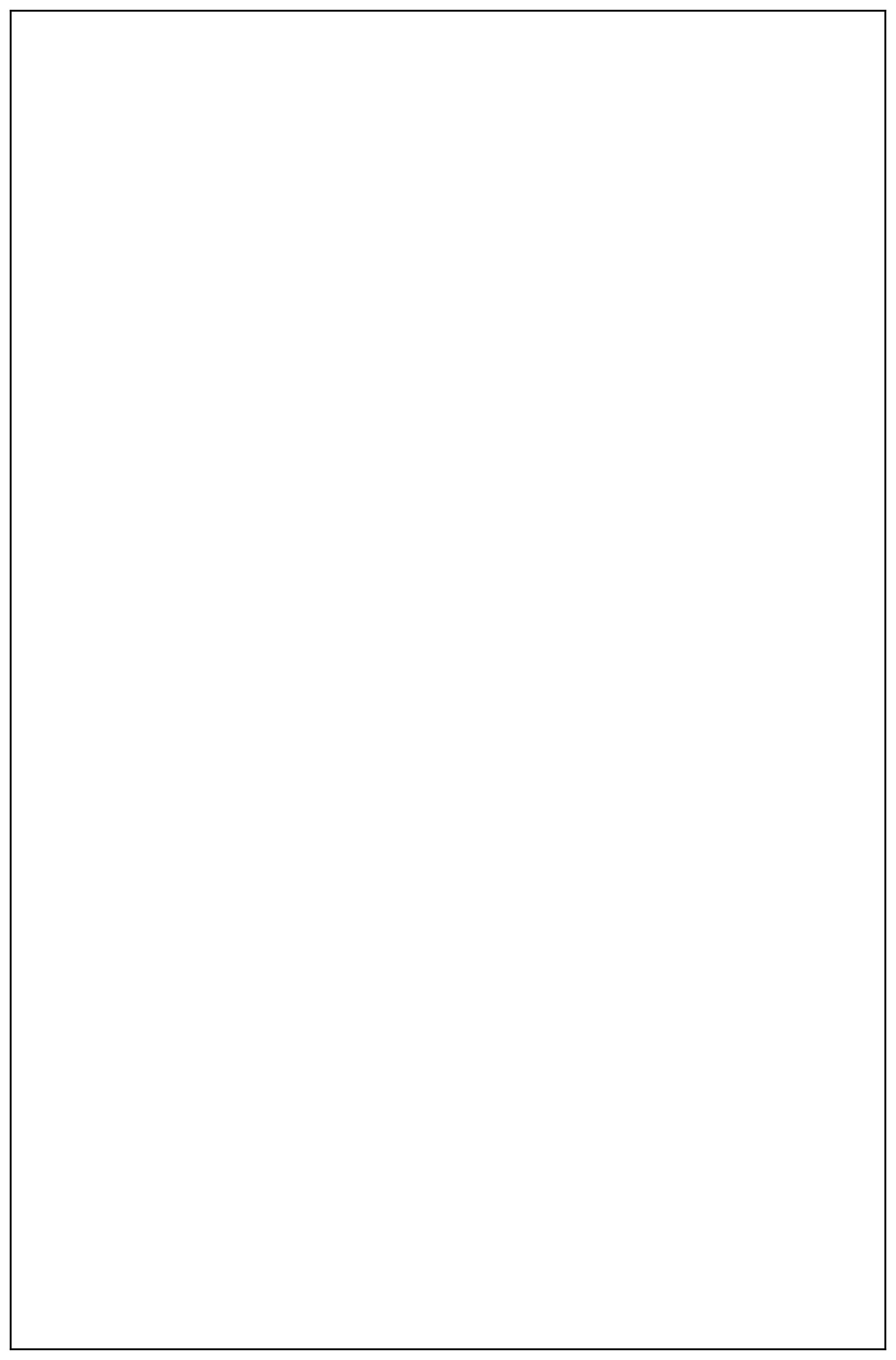


**SOLUTION :**

class BankAccount {

// Private field to store the account number private String accountNumber;

// Private field to store the balance



private double balance;

// Constructor to initialize account number and balance

public BankAccount(String accountNumber,double balance){

this.accountNumber=accountNumber;

this.balance=balance;

}

// Method to deposit an amount into the account

public void deposit(double amount) {

// Increase the balance by the deposit amount

balance+=amount;

}

// Method to withdraw an amount from the account

public void withdraw(double amount) {

// Check if the balance is sufficient for the withdrawal

if (balance >= amount) {

// Decrease the balance by the withdrawal amount

balance -= amount;

}

else

{

// Print a message if the balance is

insufficient System.out.println("Insufficient

balance"); }

}

// Method to get the current balance

public double getBalance() { //

Return the current balance

return balance;

}

public String getAccountNumber(){

return accountNumber;

}

}

class SavingsAccount extends BankAccount {

// Constructor to initialize account number and balance

public SavingsAccount(String accountNumber, double balance) {

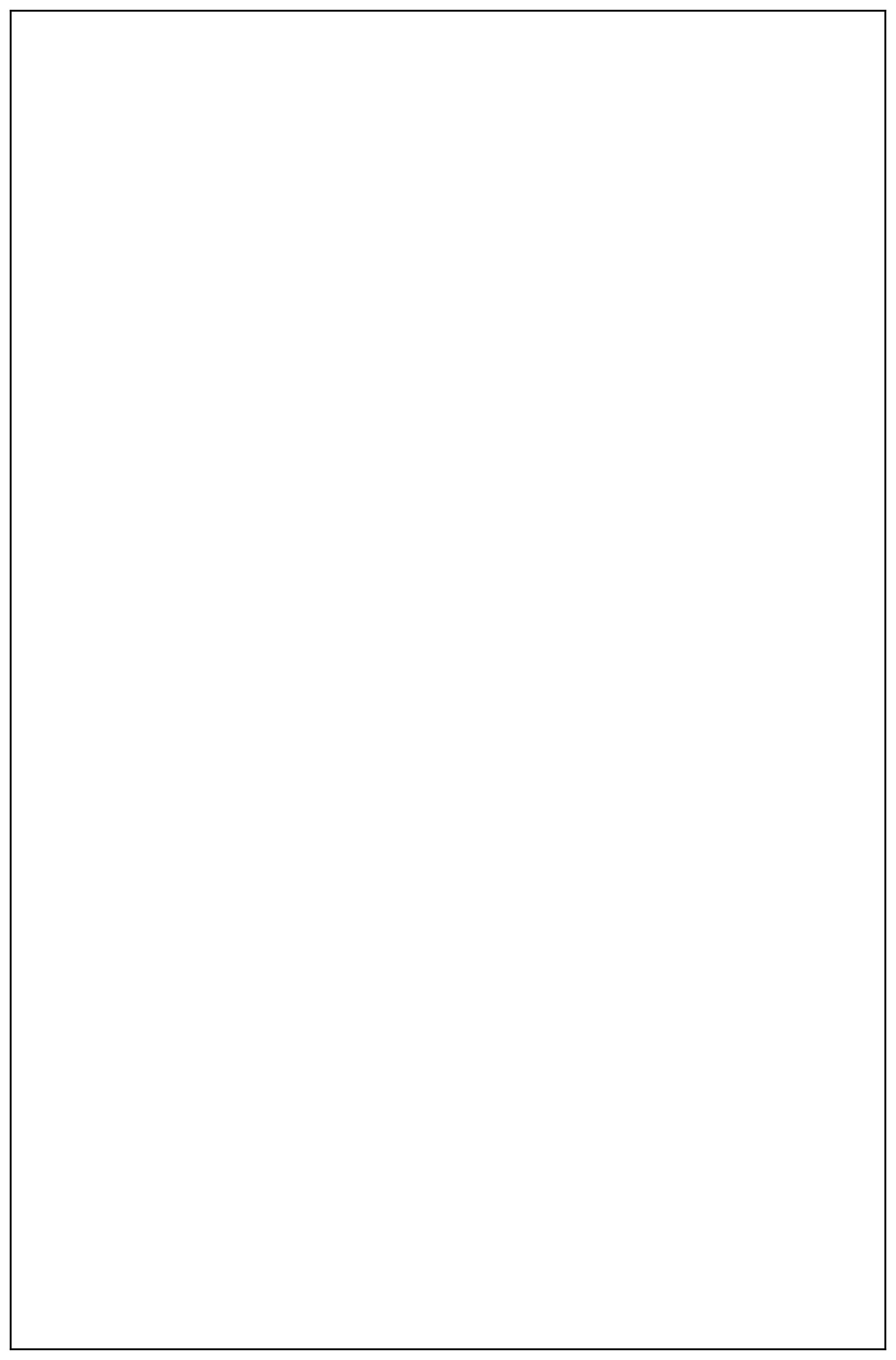
// Call the parent class constructor

super(accountNumber,balance);

}

// Override the withdraw method from the parent class

@Override



public void withdraw(double amount) {

// Check if the withdrawal would cause the balance to drop below $100

if (getBalance() - amount < 100) {

// Print a message if the minimum balance requirement is not met System.out.println("Minimum balance of $100 required!");

} else {

// Call the parent class withdraw method super.withdraw(amount);

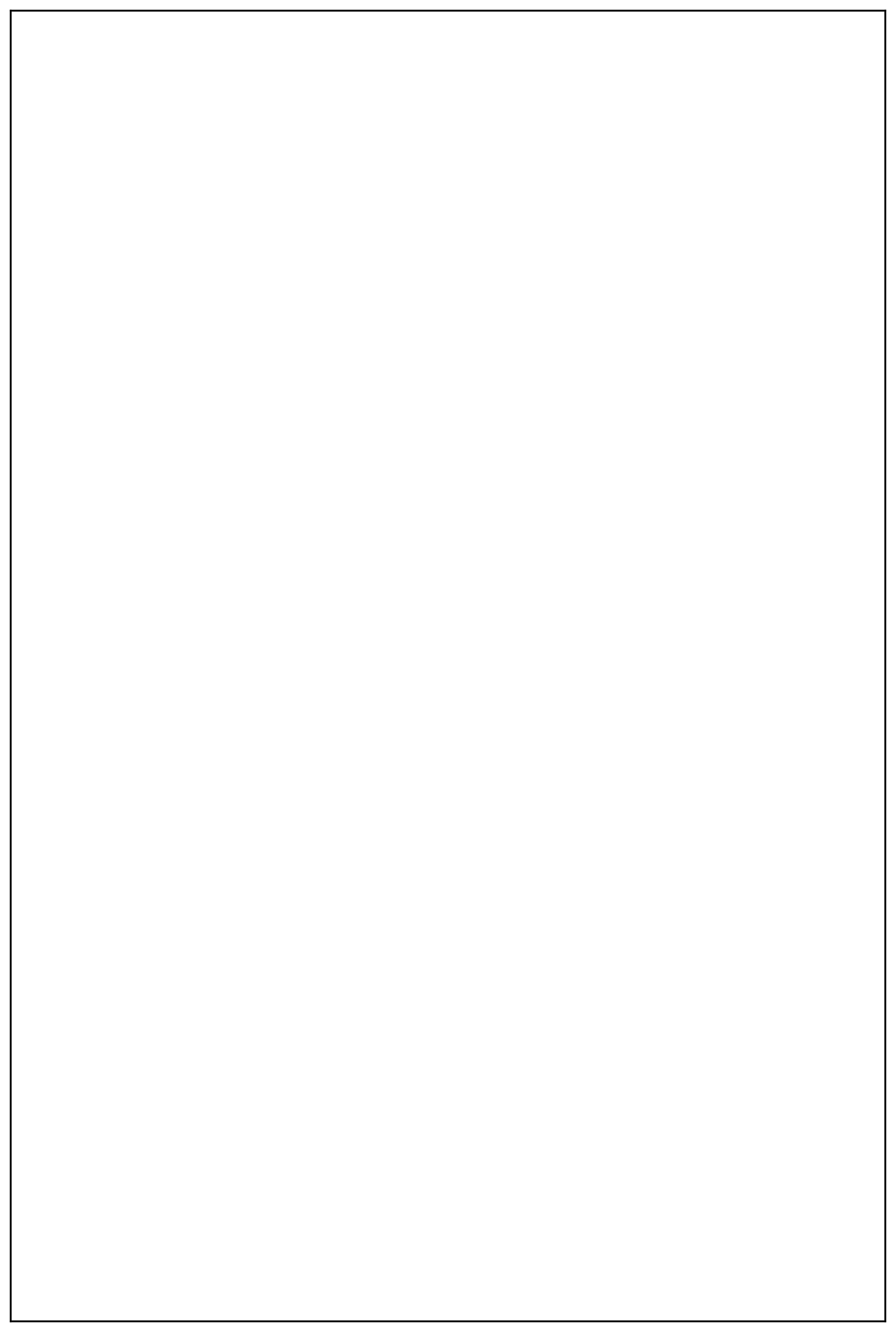
}

}

} public class Main {

public static void main(String[] args) {

// Print message to indicate creation of a BankAccount object

System.out.println("Create a Bank Account object (A/c No. BA1234) with initial balance of $500:");

// Create a BankAccount object (A/c No. "BA1234") with initial balance of $500

BankAccount BA1234 = new BankAccount("BA1234", 500);

// Print message to indicate deposit action

System.out.println("Deposit $1000 into account BA1234:");

// Deposit $1000 into account BA1234 BA1234.deposit(1000);

// Print the new balance after deposit

System.out.println("New balance after depositing $1000: $"+BA1234.getBalance());

// Print message to indicate withdrawal action

System.out.println("Withdraw $600 from account BA1234:");

// Withdraw $600 from account BA1234

BA1234.withdraw(600);

// Print the new balance after withdrawal

System.out.println("New balance after withdrawing $600: $" +

BA1234.getBalance());

// Print message to indicate creation of another SavingsAccount object

System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:");

// Create a SavingsAccount object (A/c No. "SA1000") with initial balance of $300

SavingsAccount SA1000 = new SavingsAccount("SA1000", 300);

// Print message to indicate withdrawal action

System.out.println("Try to withdraw $250 from SA1000!");

// Withdraw $250 from SA1000 (balance falls below $100)

SA1000.withdraw(250);

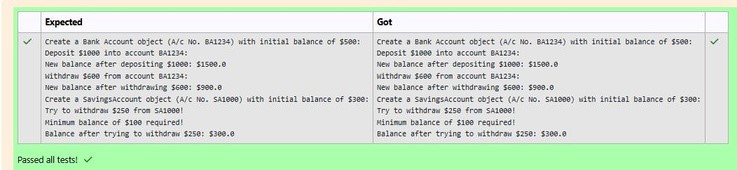
// Print the balance after attempting to withdraw $250

System.out.println("Balance after trying to withdraw $250: $" +

SA1000.getBalance()); }

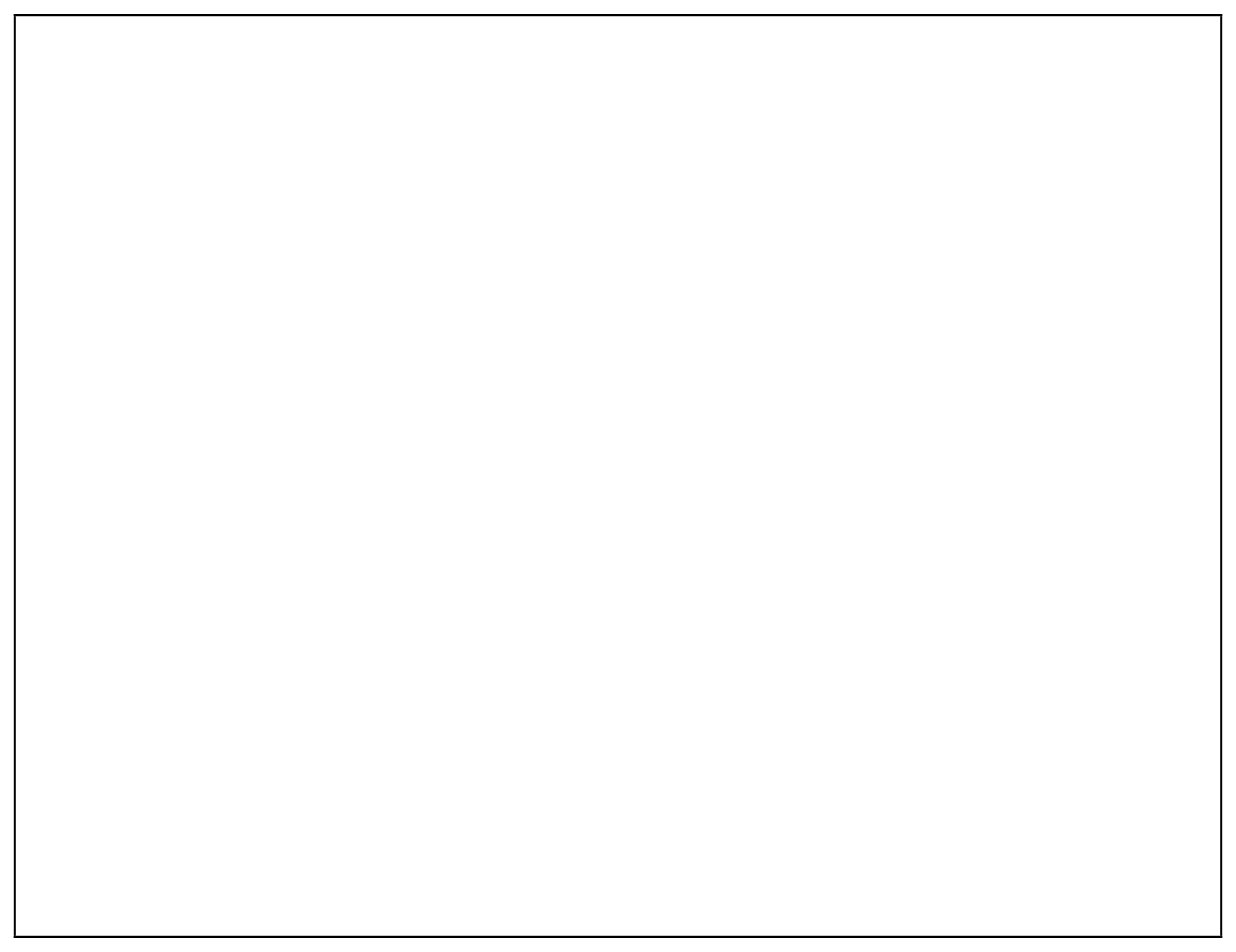
}

**OUTPUT :**



**2.**





**SOLUTION :**

class College

{

public String collegeName;

public College(String collegeName)

{

// initialize the instance variables

this.collegeName=collegeName; }

public void admitted() {

System.out.println("A student admitted in "+collegeName);

}

} class Student extends

College{

String studentName;

String department;

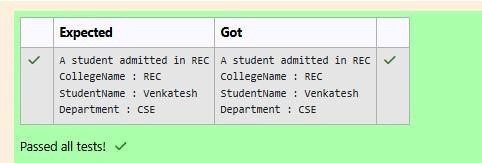
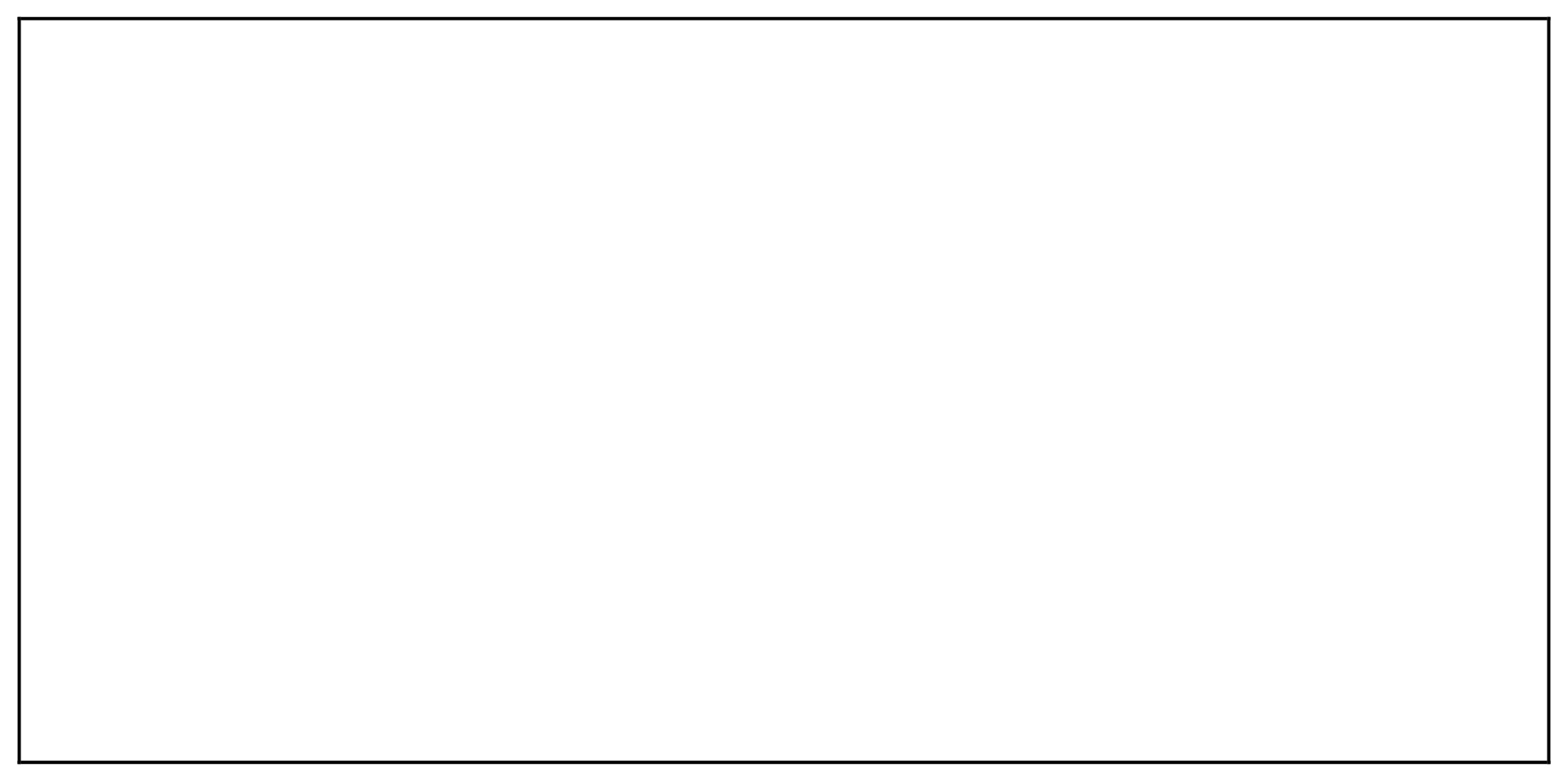
public Student(String collegeName, String studentName,String department) {

// initialize the instance variables

super(collegeName);

this.studentName=studentName;

this.department=department;



}

public String toString(){

// return the details of the student return "CollegeName :

"+collegeName+"\n"+"StudentName :

"+studentName+"\n"+"Department : "+department;

}

}

public class Main {

public static void main (String[] args) {

Student s1 = new Student("REC","Venkatesh","CSE");

s1.admitted();

// invoke the admitted() method

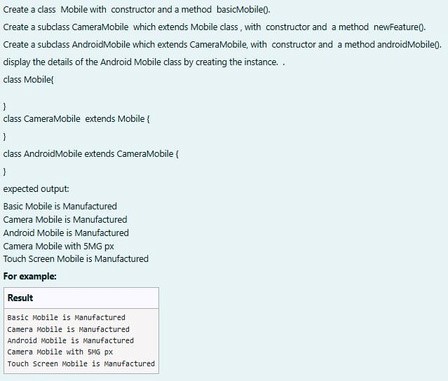
System.out.println(s1.toString());

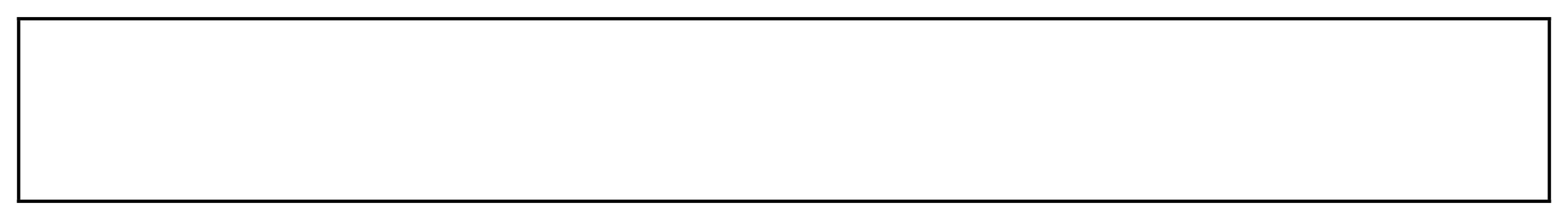
}

}

**OUTPUT :**

**3.**



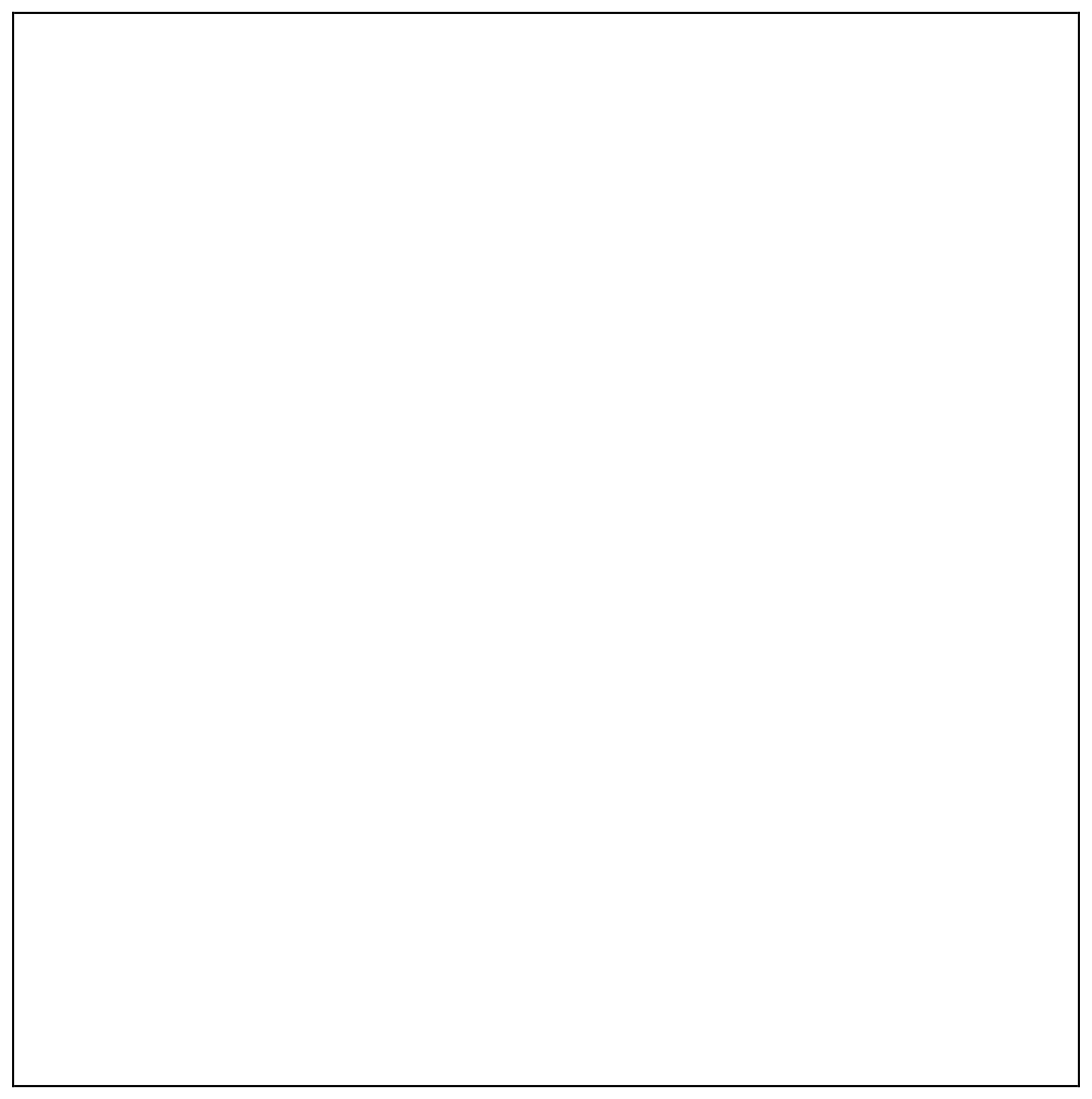


**SOLUTION :**

class mob{

mob(){

System.out.println("Basic Mobile is Manufactured");



**OUTPUT :**

}

void basmob(){

System.out.println("Basic Mobile is Manufactured");

}

}

class cam extends

mob{ cam(){

super();

System.out.println("Camera Mobile is Manufactured");

}

void newm(){

System.out.println("Camera Mobile with 5MG px");

}

}

class and extends

cam{ and(){

super();

System.out.println("Android Mobile is Manufactured");

}

void andmob(){

System.out.println("Touch Screen Mobile is Manufactured");

}

public class Main{ public static

}

void main(String[]args){ and

andmob=new and(); andmob.newm();

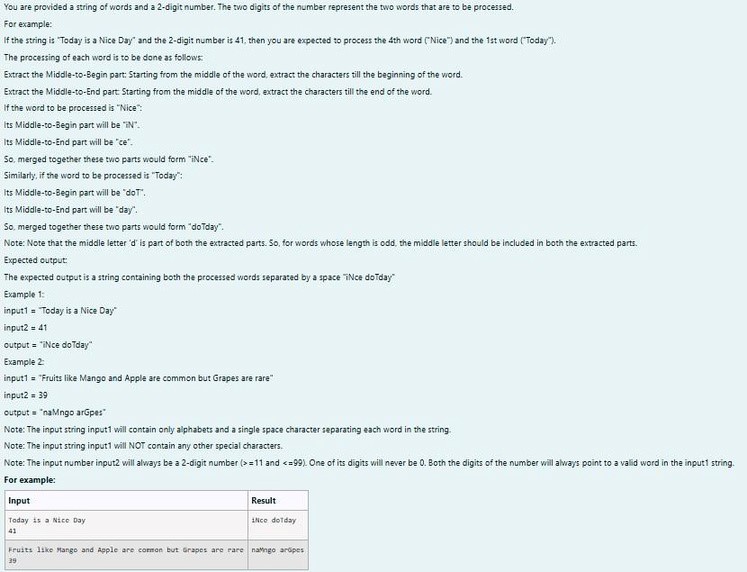
andmob.andmob();

}

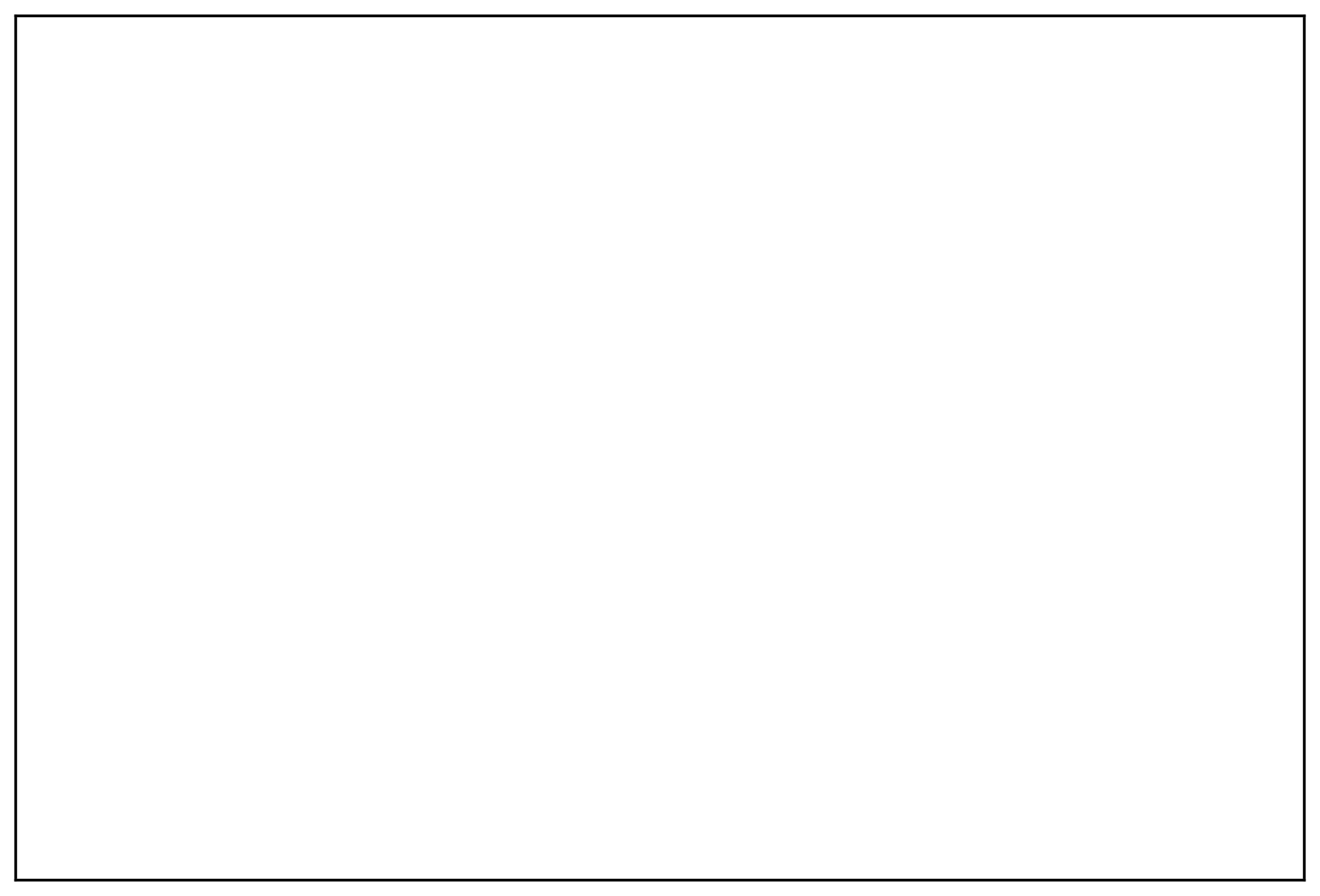
}

[**Lab-06-String, StringBuffer**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=54)

**1.**



**SOLUTION :**



import java.util.\*; public class mix{

public static void main(String[] args){

Scanner scan = new Scanner(System.in);

String g = scan.nextLine(); int n =

scan.nextInt(),ones,flag = 0; StringBuffer

temp = new StringBuffer(); StringBuffer

temp1 = new StringBuffer(); int space =

0

; while (n > 0){ ones = (n %10) - 1;

for(int i = 0; i < g.length();i++){

if (g.charAt(i) == ' '){ space

=

space + 1;

}

else if(space == ones && flag == 0){

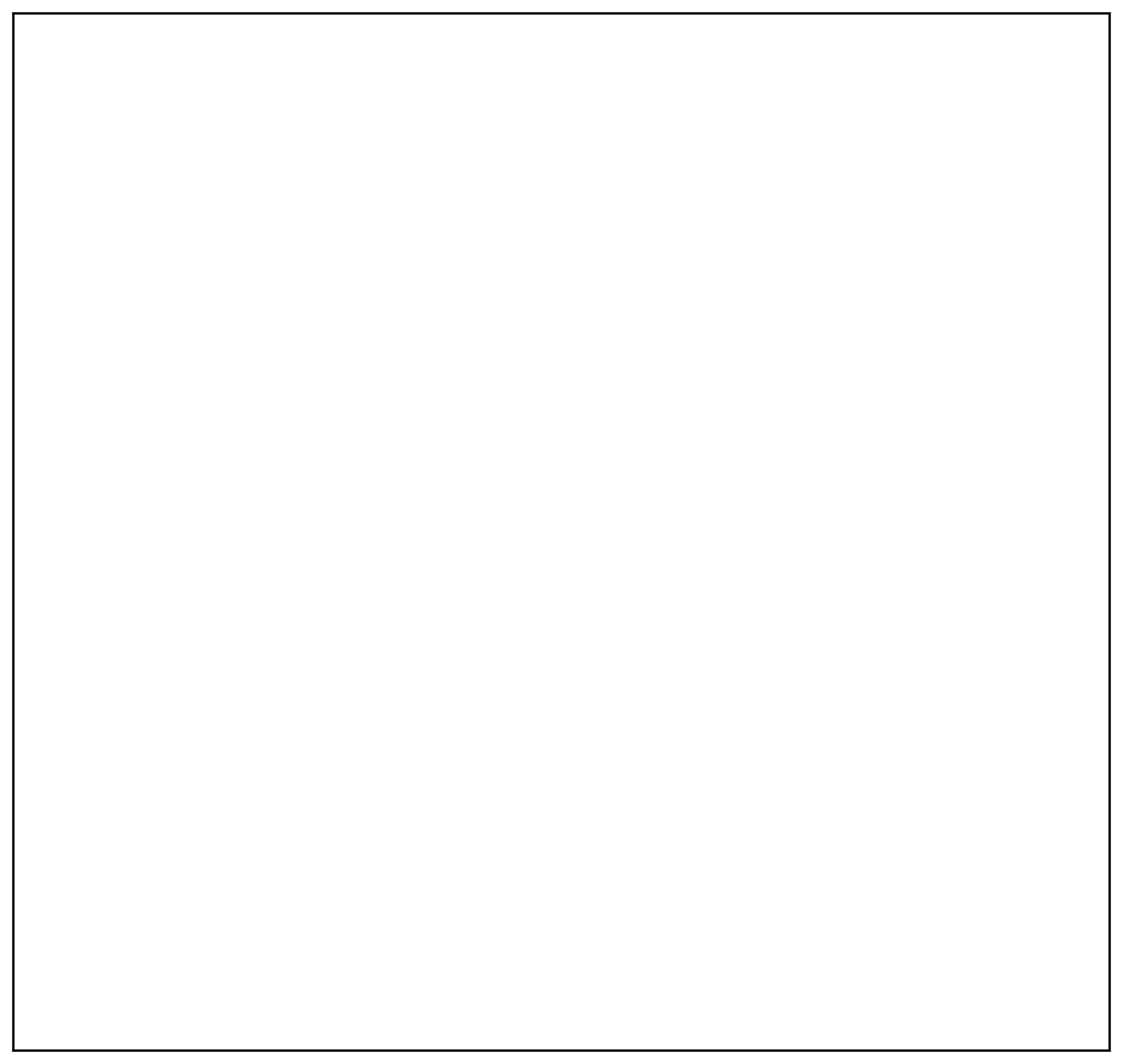
temp.append(Character.toString(g.charAt(i)));

}

else if(space == ones && flag == 1){

temp1.append(Character.toString(g.charAt(i)));

}



}

space

=

0

; flag

=

1

; n = n

/10;

}

rew m = new rew();

System.out.println(m.r(temp1.toString()) + " " + m.r(temp.toString()));

}

}

class rew{

String r(String a){ int le

=

a.length(),n,q;

StringBuffer temp3 = new StringBuffer();

if(le % 2 == 1){

n = ((int)(le/2));

q = ((int)(le/2));

}

else{ n

=

((

int)(le/2)) - 1;

q = ((int)(le/2));

}

for(int i = n;i >= 0;i--){ temp3.append(Character.toString(a.charAt(i)));

}

for(int i = q;i < le;i

++){

temp3.append(Character.toString(a.charAt(i)));

}

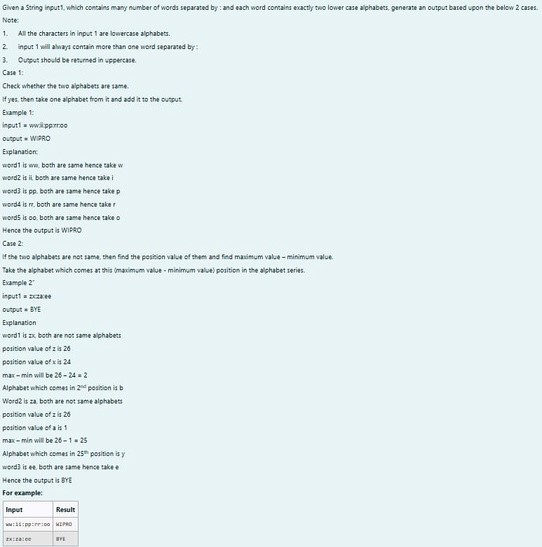
return temp3.toString();

}

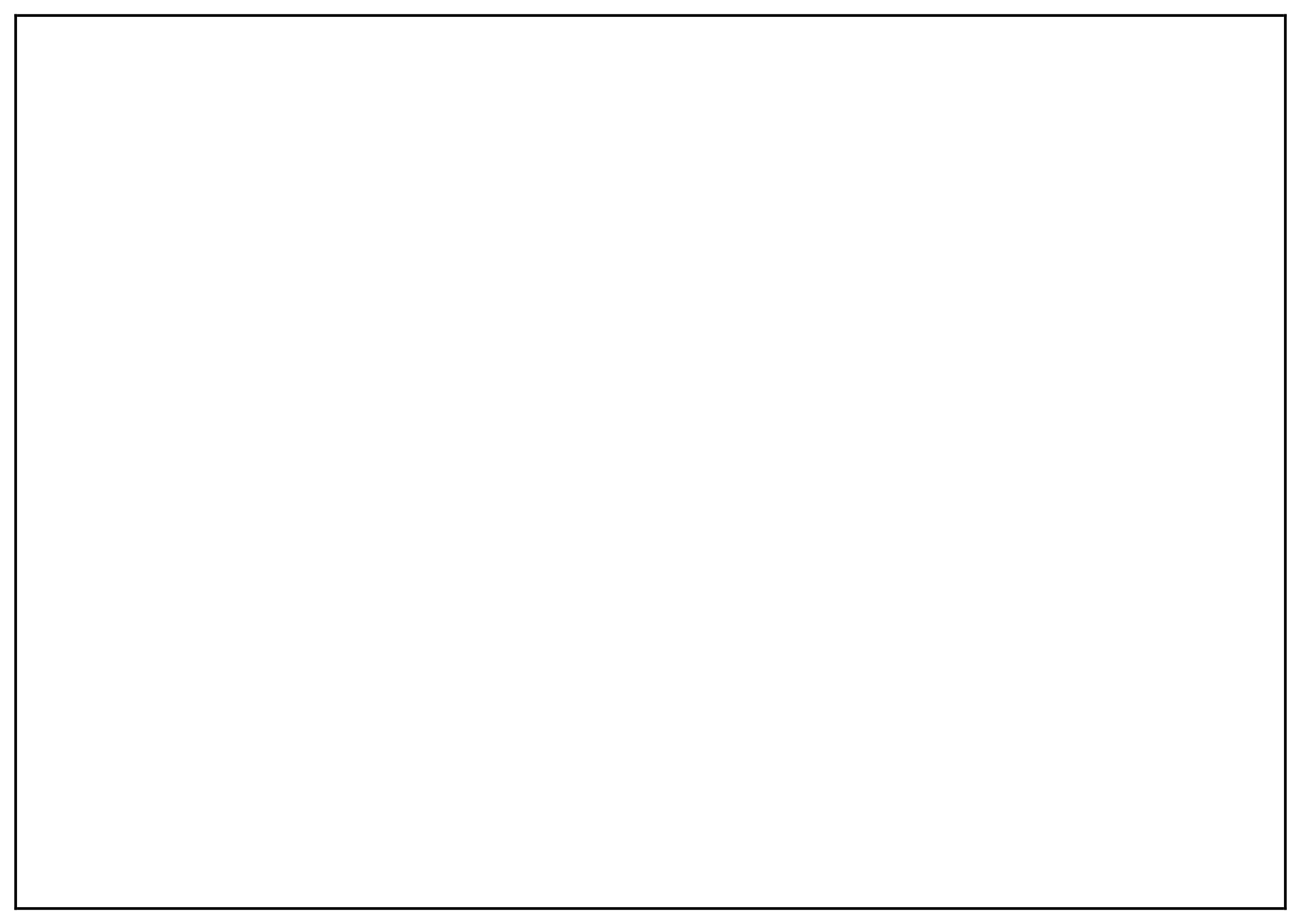
}

**OUTPUT :**

**2.**



**SOLUTION :**



import java.util.\*; class diff{ char different(char

a, char b){ if ((int)a != (int)b) return

(

char)((int)'a' + ((int)a-(int)b) - 1);

return a;

}

}

public class Main{ public static void

main(String[] args){ Scanner scan = new

Scanner(System.in);

diff z = new diff();

String q = scan.nextLine();

StringBuffer ans = new StringBuffer();

StringBuffer temp = new

StringBuffer(); for(int i = 0;i <

q.length();i++){ if(q.charAt(i) == ':'){

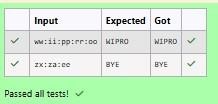
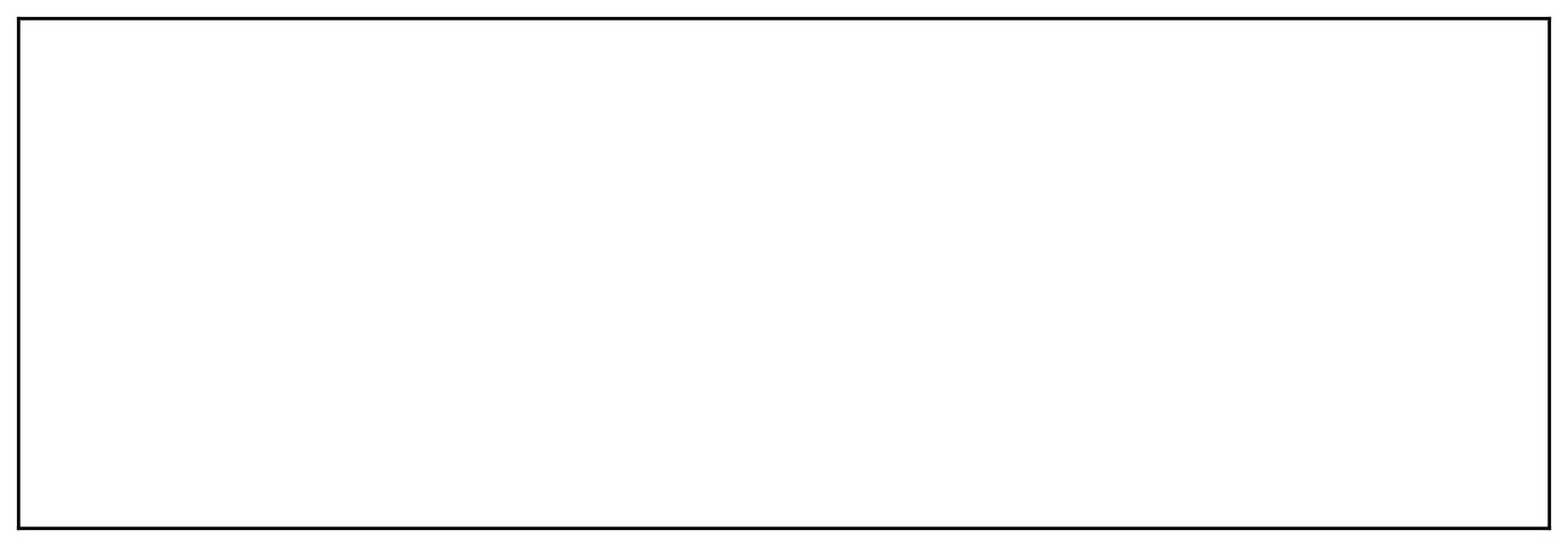
temp.append(" ");

}

else

{

temp.append(Character.toString(q.charAt(i))); }



}

String h = temp.toString(); for(int i

=

0;i < temp.length();i++){ if(i%

3

== 0){

ans.append(Character.toString(z.different(h.charAt(i),h.charAt(i+1))));

}

}

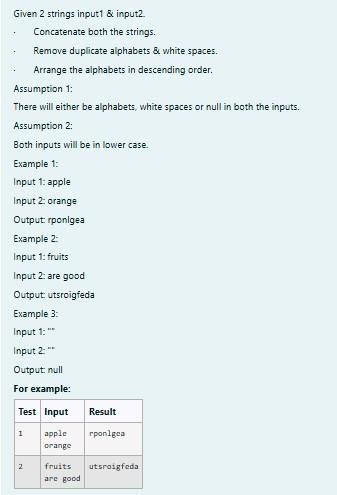
System.out.print(ans.toString().toUpperCase());

}

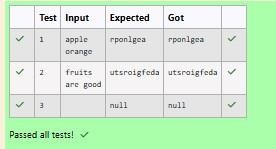
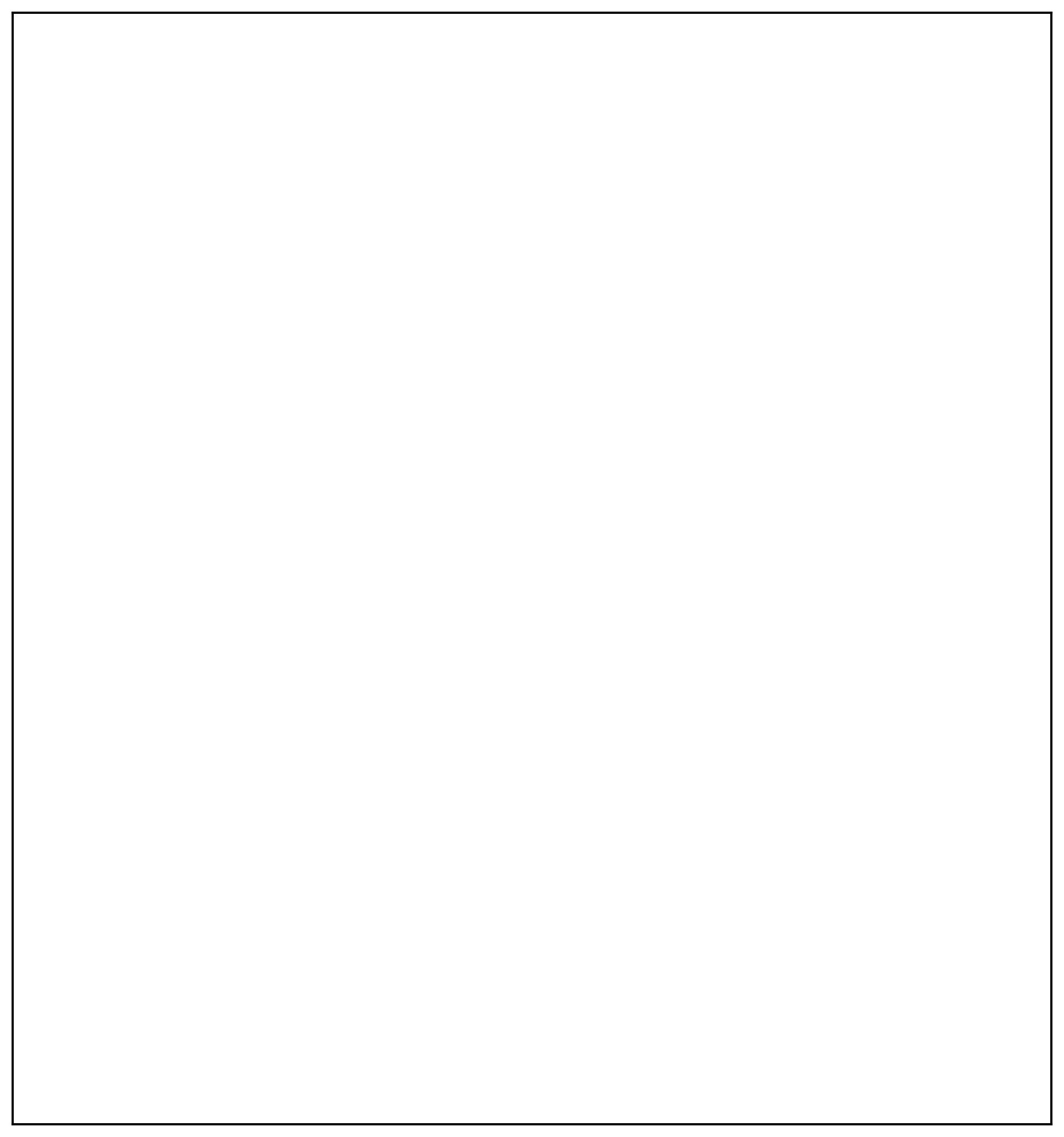
}

**OUTPUT :**

**3.**



**SOLUTION :**



import java.util.\*;

public class HelloWorld { public static

void main(String[] args) {

Scanner scan = new Scanner(System.in);

String a = scan.nextLine();

String b = scan.nextLine();

StringBuffer ab = new StringBuffer();

if(a.trim().isEmpty() && b.trim().isEmpty()){

System.out.print("null");

}

else{

for(int i = 0;i < a.length();i++){ if (a.charAt(i)

!=

' ')

{

ab.append(Character.toString(a.charAt(i))); }

}

for(int i = 0;i < b.length();i++){ if (b.charAt(i)

!=

' '

){

ab.append(Character.toString(b.charAt(i))); }

}

char[] d = ab.toString().toCharArray();

Arrays.sort(d);

for(int i = d.length - 1;i >= 1;i--){

if(d[i] != d[i-1])

System.out.print(d[i]);

}

System.out.print(d[0]);

}

}

}

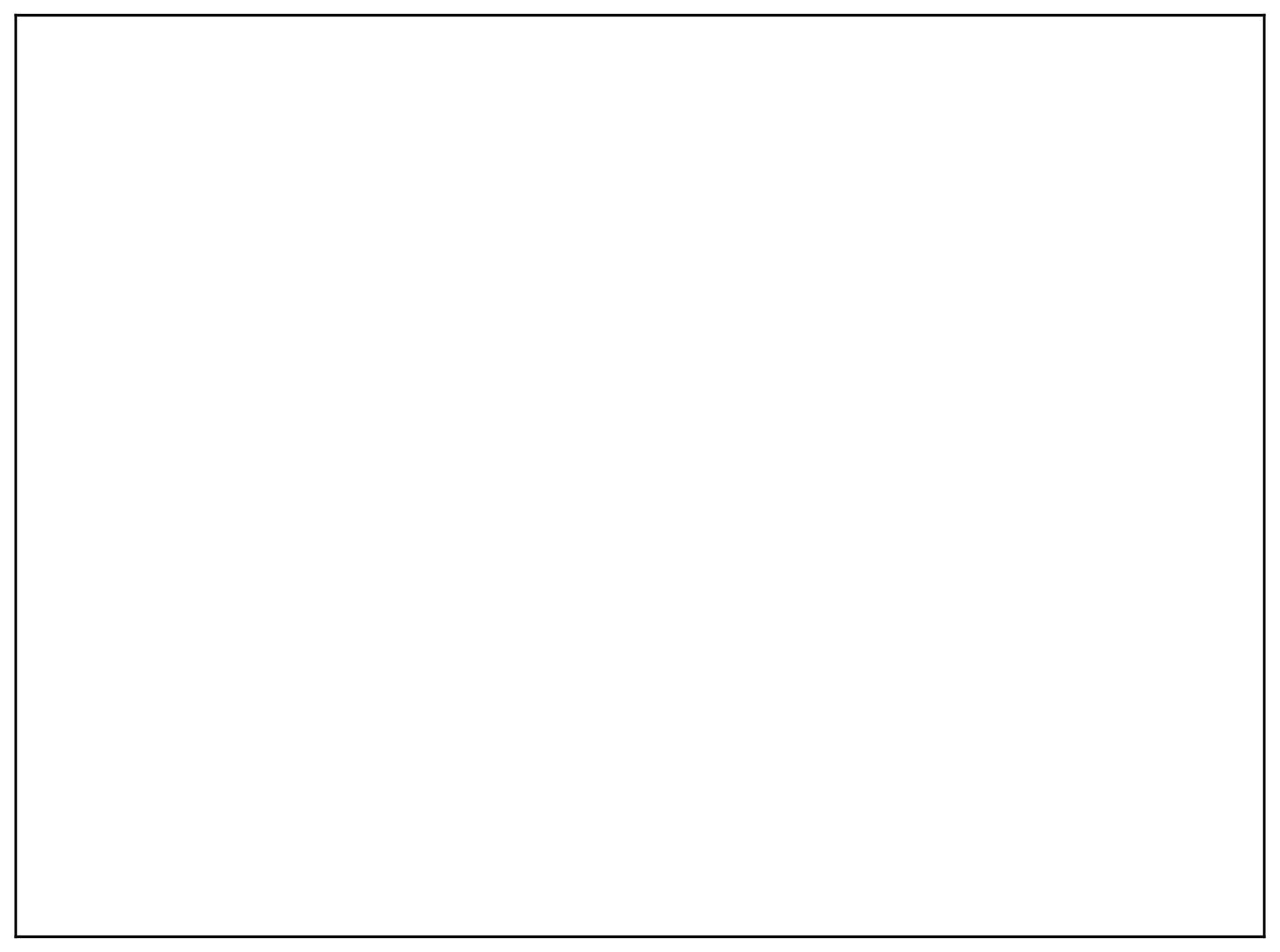
**OUTPUT :**

[**Lab-07-Interfaces**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=58)

**1.**



**SOLUTION :**



// Define the RBI interface

interface RBI {

// Variable declaration

String parentBank = "RBI";

// Abstract method

double rateOfInterest();

// Default method

default void policyNote() {

System.out.println("RBI has a new Policy issued in 2023");

}

// Static method

static void regulations() {

System.out.println("RBI has updated new regulations in 2024.");

}

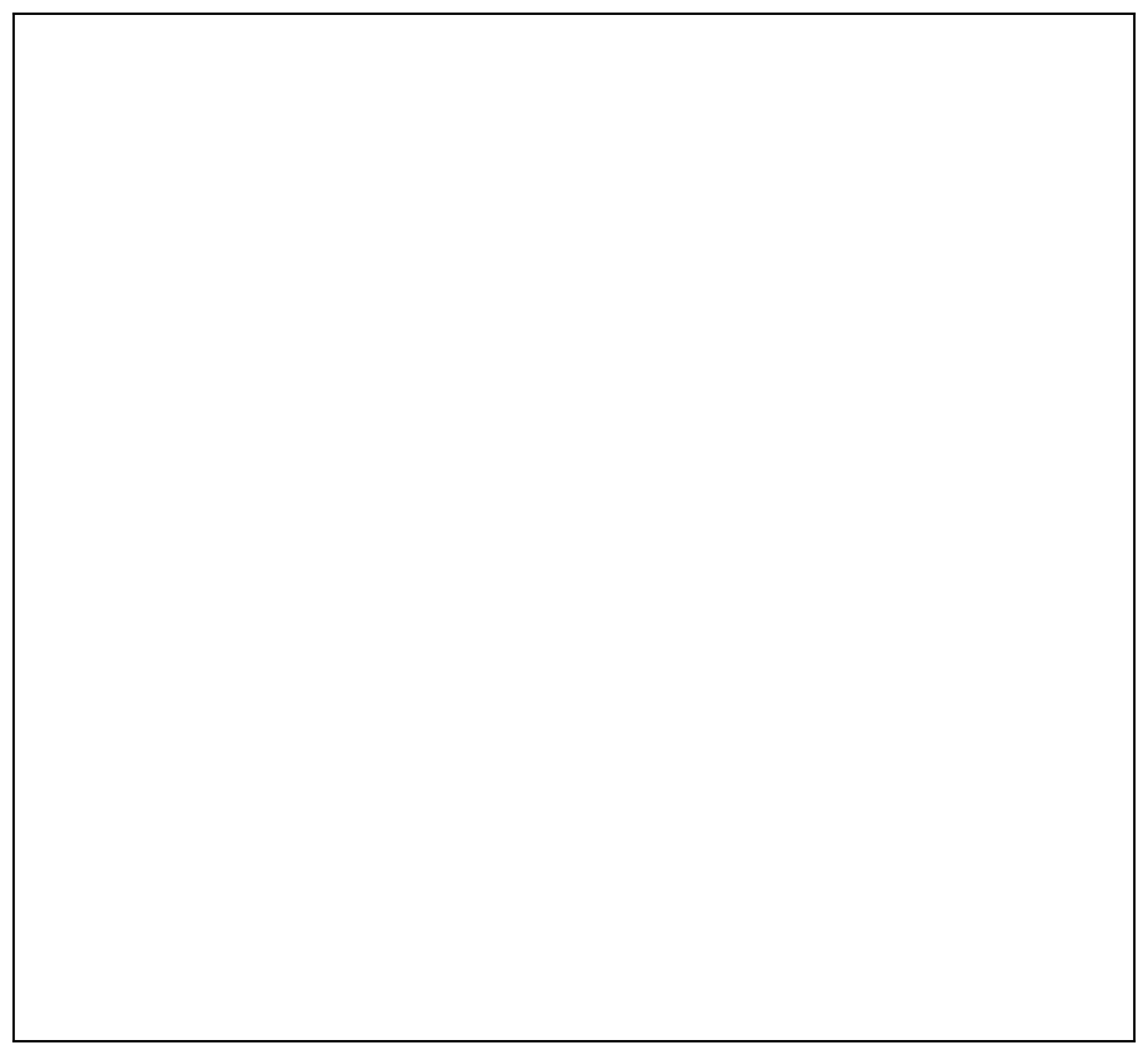
}

// SBI class implementing RBI interface

class SBI implements RBI {

// Implementing the abstract method

public double rateOfInterest() {



return 7.6;

}

}

// Karur class implementing RBI

interface class Karur implements RBI { //

Implementing the abstract method public

double rateOfInterest() { return 7.4;

}

}

// Main class to test the functionality

public class Main { public static void

main(String[] args) {

// RBI policies and regulations

RBI rbi = new SBI(); // Can be any class implementing RBI

rbi.policyNote(); // Default method RBI.regulations();

// Static method

// SBI bank details

SBI sbi = new SBI();

System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");

// Karur bank details

Karur karur = new Karur();

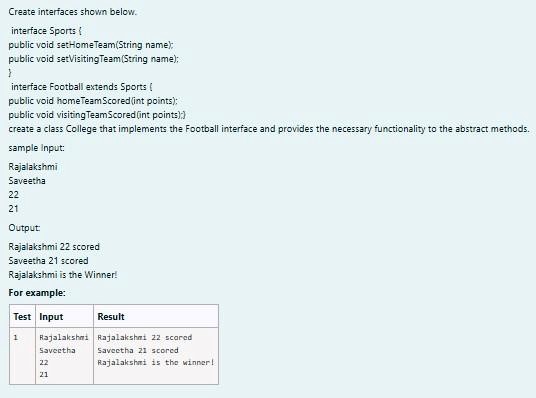
System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");

}

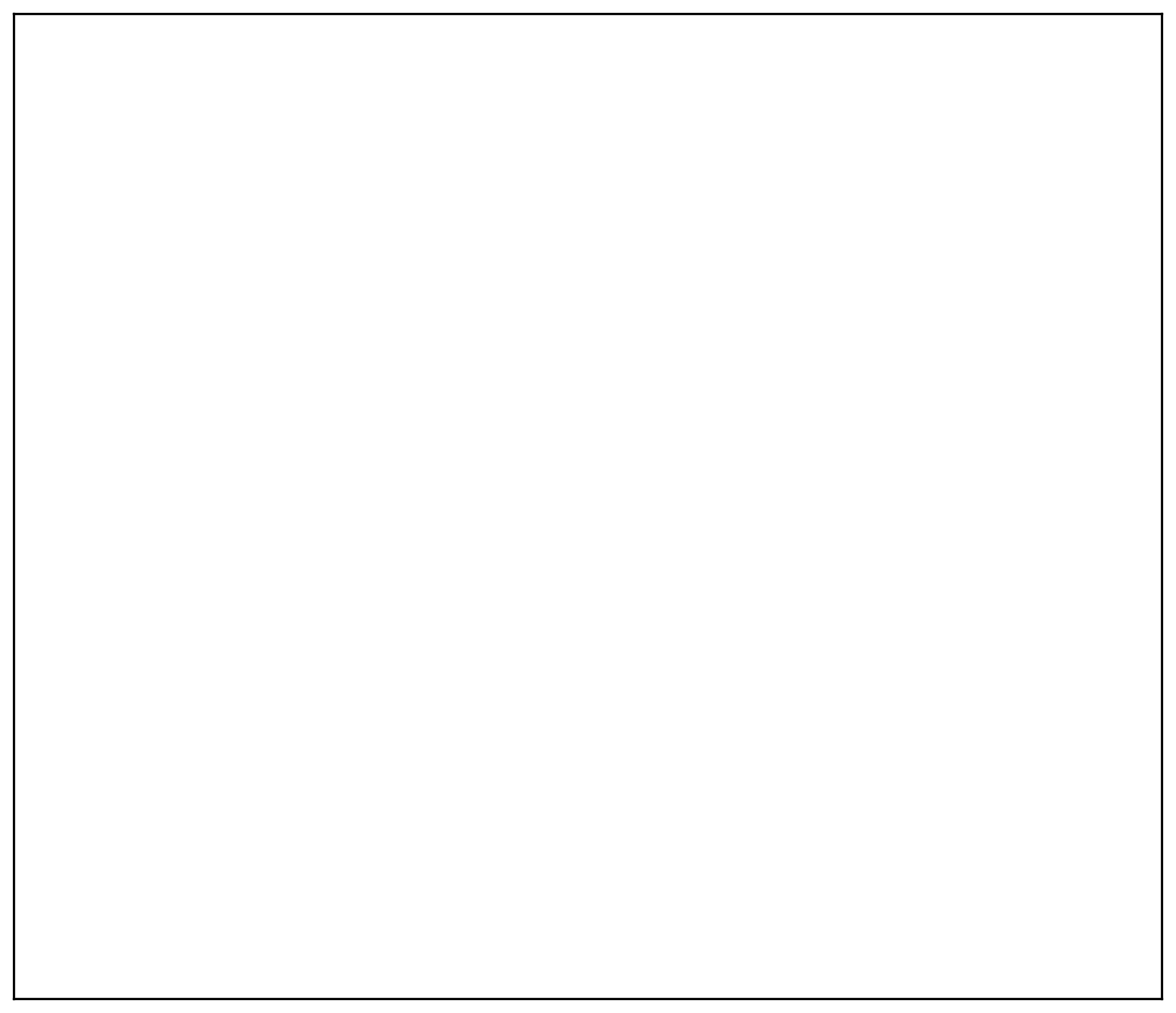
}

**OUTPUT :**

**2.**



**SOLUTION :**



import java.util.Scanner;

interface Sports { void

setHomeTeam(String name); void

setVisitingTeam(String name);

}

interface Football extends Sports {

void homeTeamScored(int points);

void visitingTeamScored(int points);

}

class College implements Football {

private String homeTeam; private

String visitingTeam; private int

homeTeamPoints = 0; private int

visitingTeamPoints = 0;

public void setHomeTeam(String name) {

this.homeTeam = name;

}

public void setVisitingTeam(String name) {

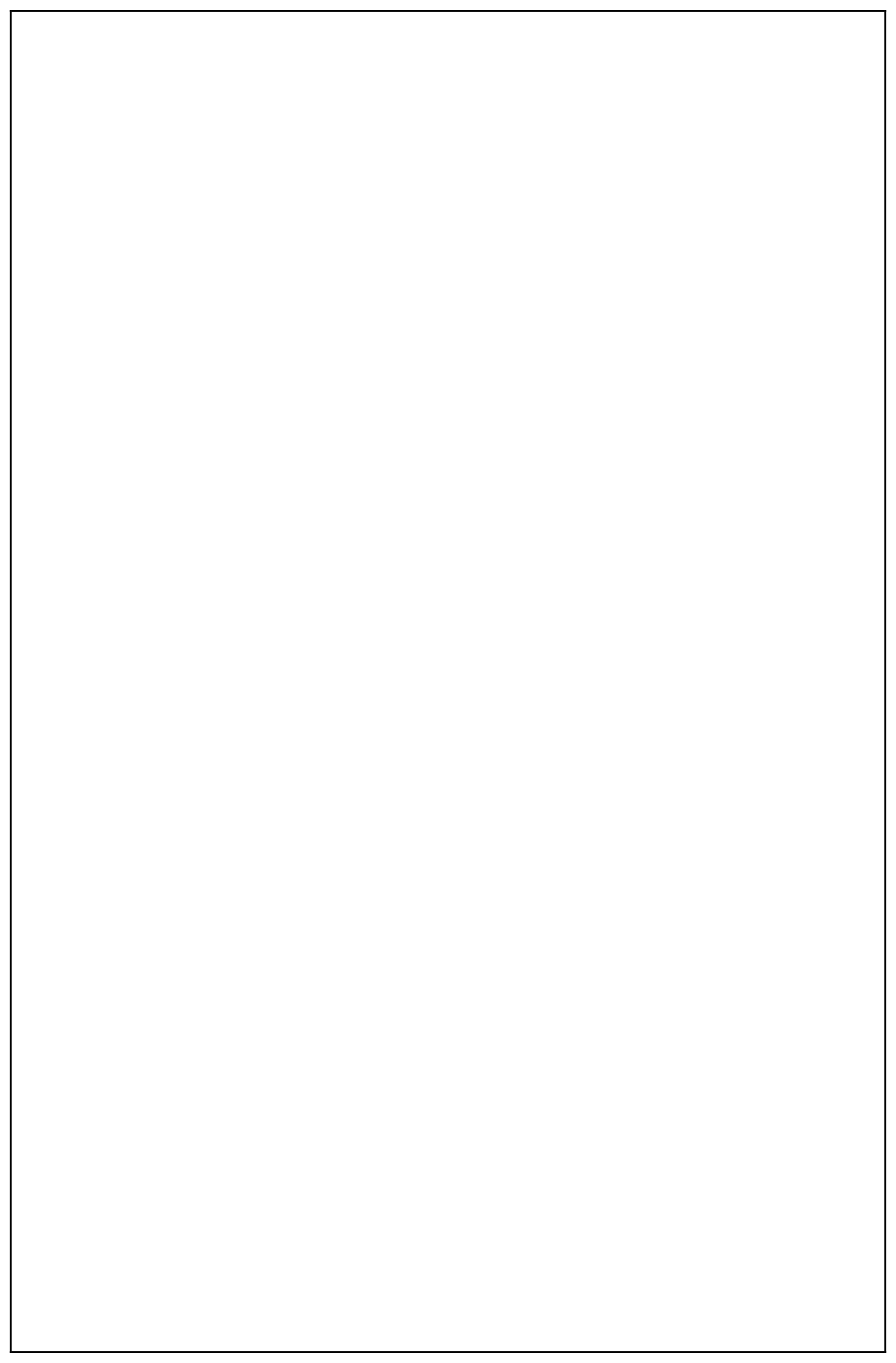
this.visitingTeam = name;

}

public void homeTeamScored(int points

)

{



homeTeamPoints += points;

System.out.println(homeTeam + " " + points + " scored");

}

public void visitingTeamScored(int points) {

visitingTeamPoints += points;

System.out.println(visitingTeam + " " + points + " scored");

}

public void winningTeam() { if

(

homeTeamPoints > visitingTeamPoints)

{

System.out.println(homeTeam + " is the winner!");

}

else if (homeTeamPoints < visitingTeamPoints)

{

System.out.println(visitingTeam + " is the winner!");

}

else

{

System.out.println("It's a tie match.");

}

}

}

public class Main { public static void

main(String[] args) {

Scanner sc = new Scanner(System.in);

// Get home team name

String hname = sc.nextLine();

// Get visiting team name

String vteam = sc.nextLine();

// Create College object College

match = new College();

match.setHomeTeam(hname);

match.setVisitingTeam(vteam);

// Get points scored by home team

int htpoints = sc.nextInt();

match.homeTeamScored(htpoints);

// Get points scored by visiting team

int vtpoints = sc.nextInt();

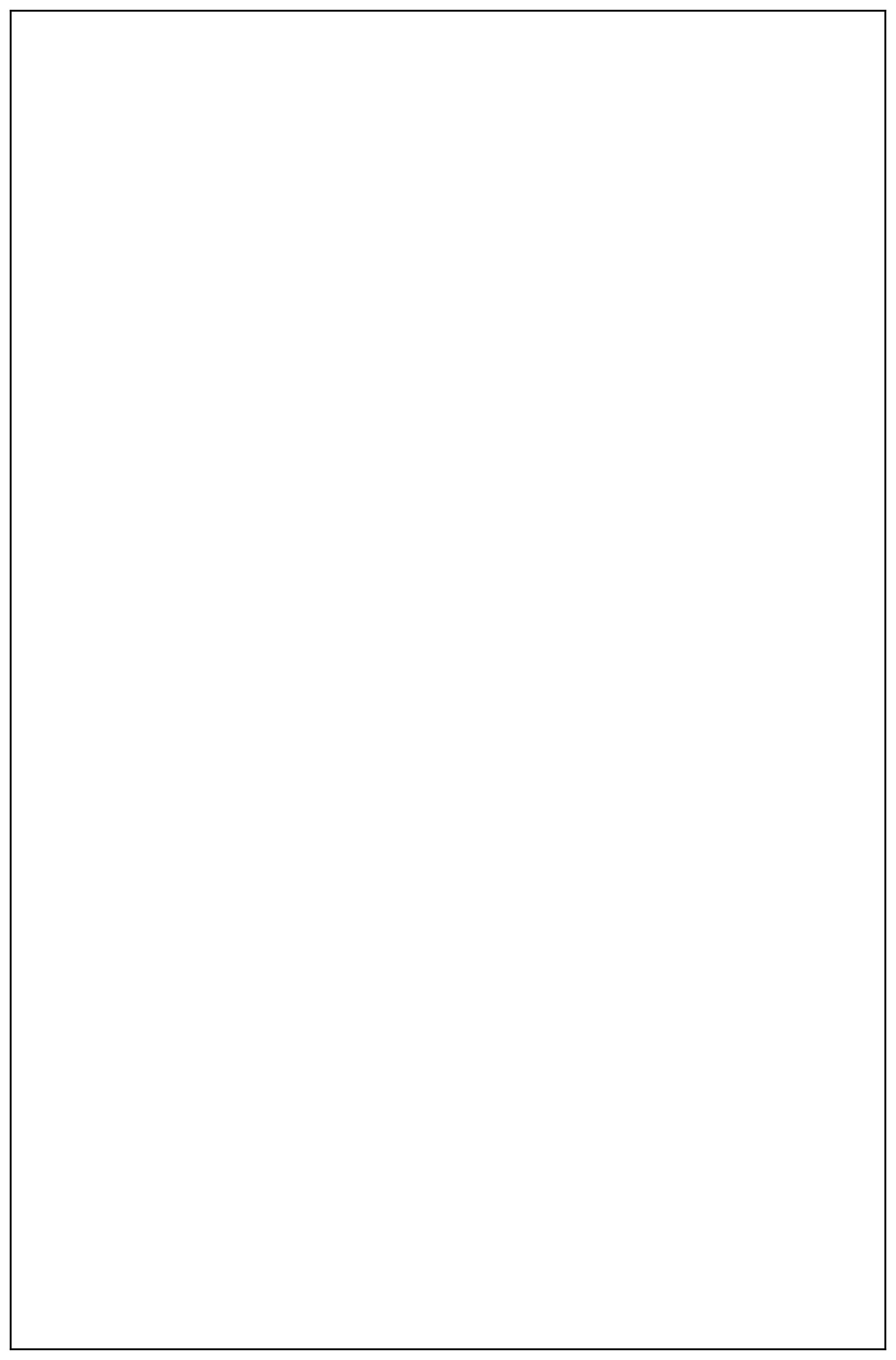
match.visitingTeamScored(vtpoints);

// Determine and print the winning team

match.winningTeam();

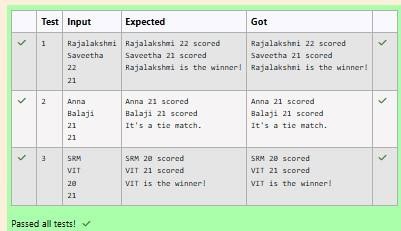
sc.close();

}



}

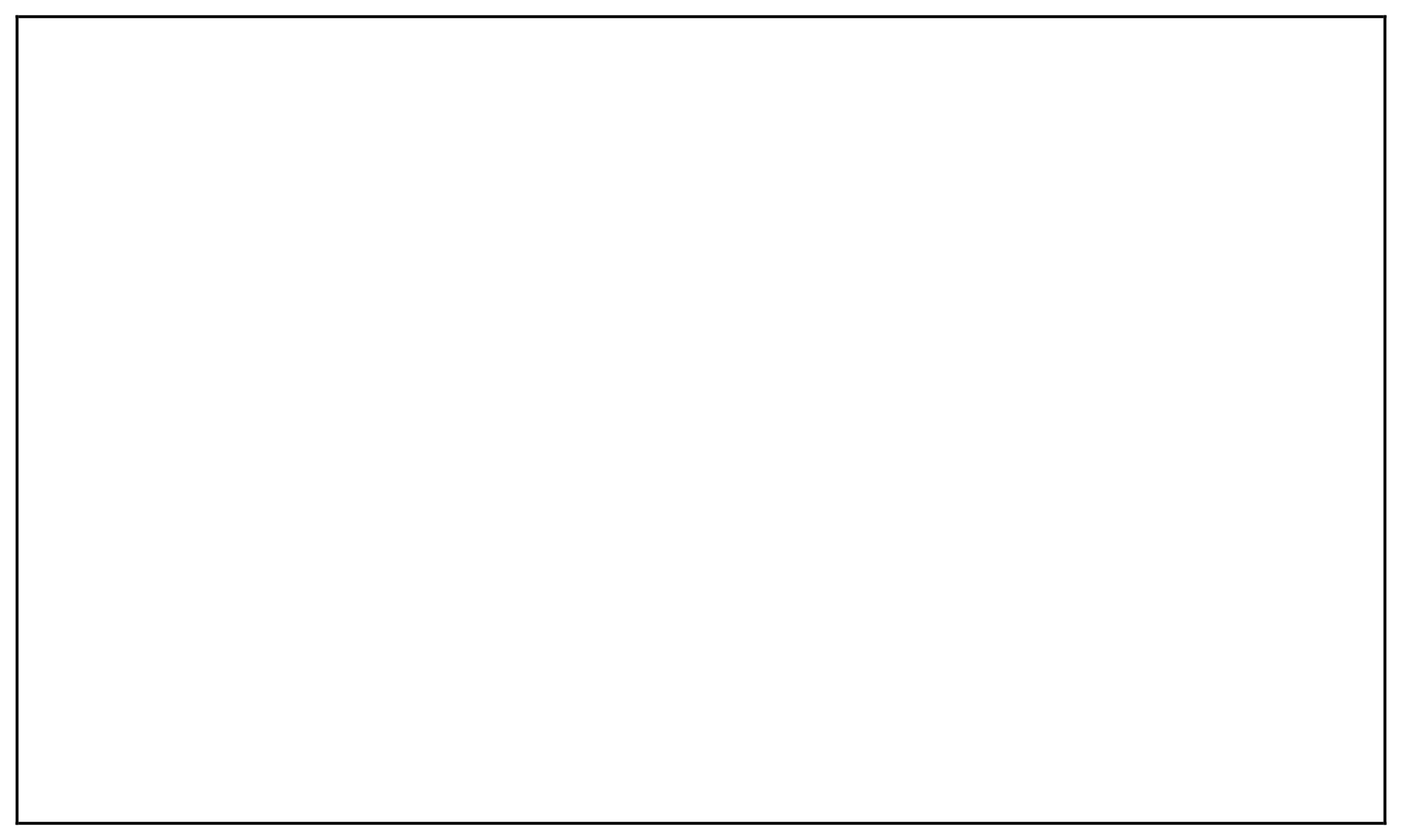
**OUTPUT :**



**3.**



**SOLUTION :**



import java.util.Scanner;

// Define the Playable interface

interface Playable {

// Abstract method to play the respective sport

void play();

}

// Football class implementing Playable interface

class Football implements Playable {

String name;

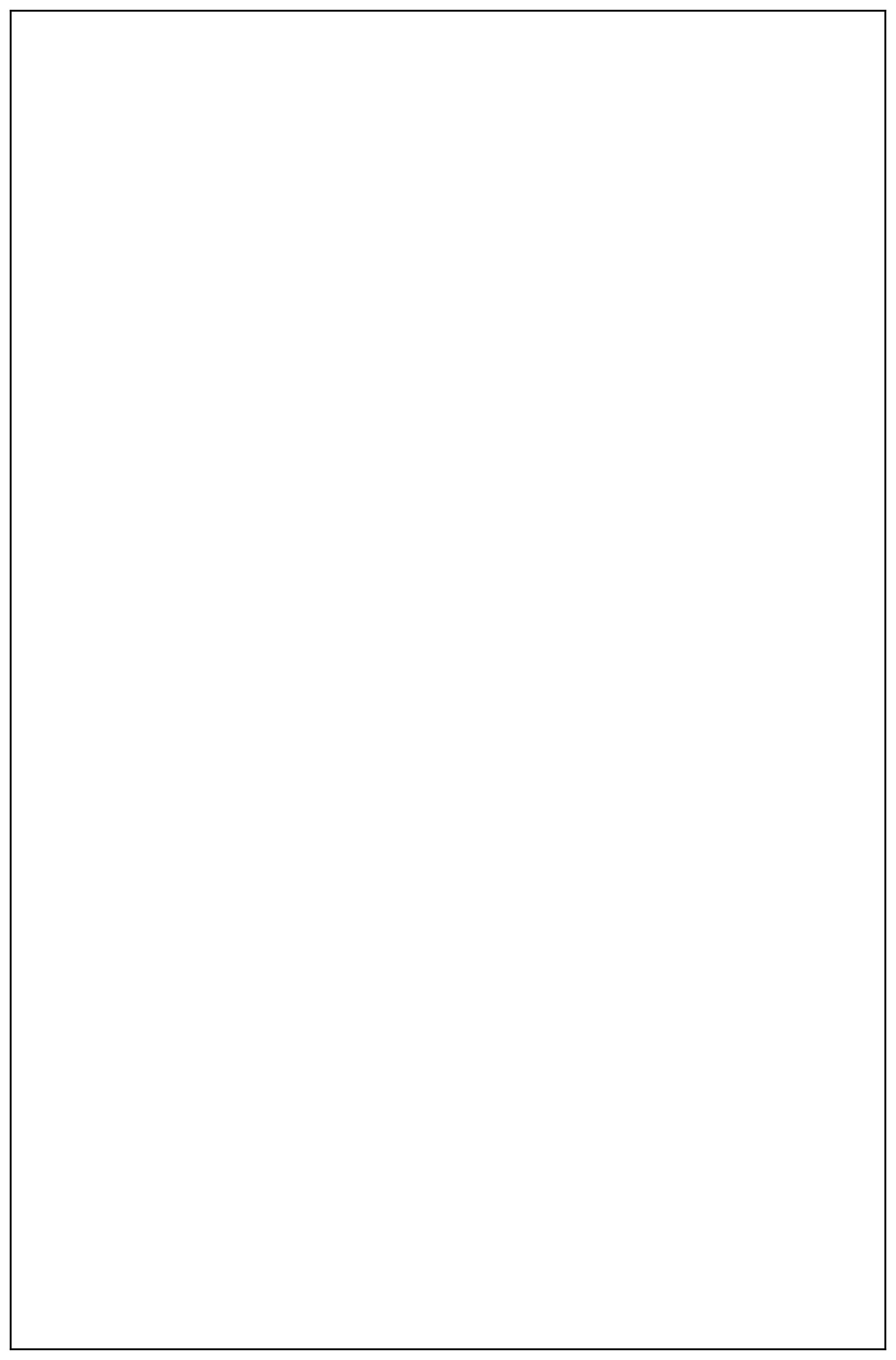
// Constructor

public Football(String name) {

this.name = name;

}

// Override the play method



public void play() {

System.out.println(name + " is Playing football");

}

}

// Volleyball class implementing Playable interface

class Volleyball implements Playable {

String name;

// Constructor

public Volleyball(String name) {

this.name = name;

}

// Override the play method

public void play() {

System.out.println(name + " is Playing volleyball");

}

}

// Basketball class implementing Playable interface

class Basketball implements Playable {

String name;

// Constructor

public Basketball(String name) {

this.name = name;

}

// Override the play method

public void play() {

System.out.println(name + " is Playing basketball");

}

}

// Main class to test the functionality

public class Main { public static void

main(String[] args) {

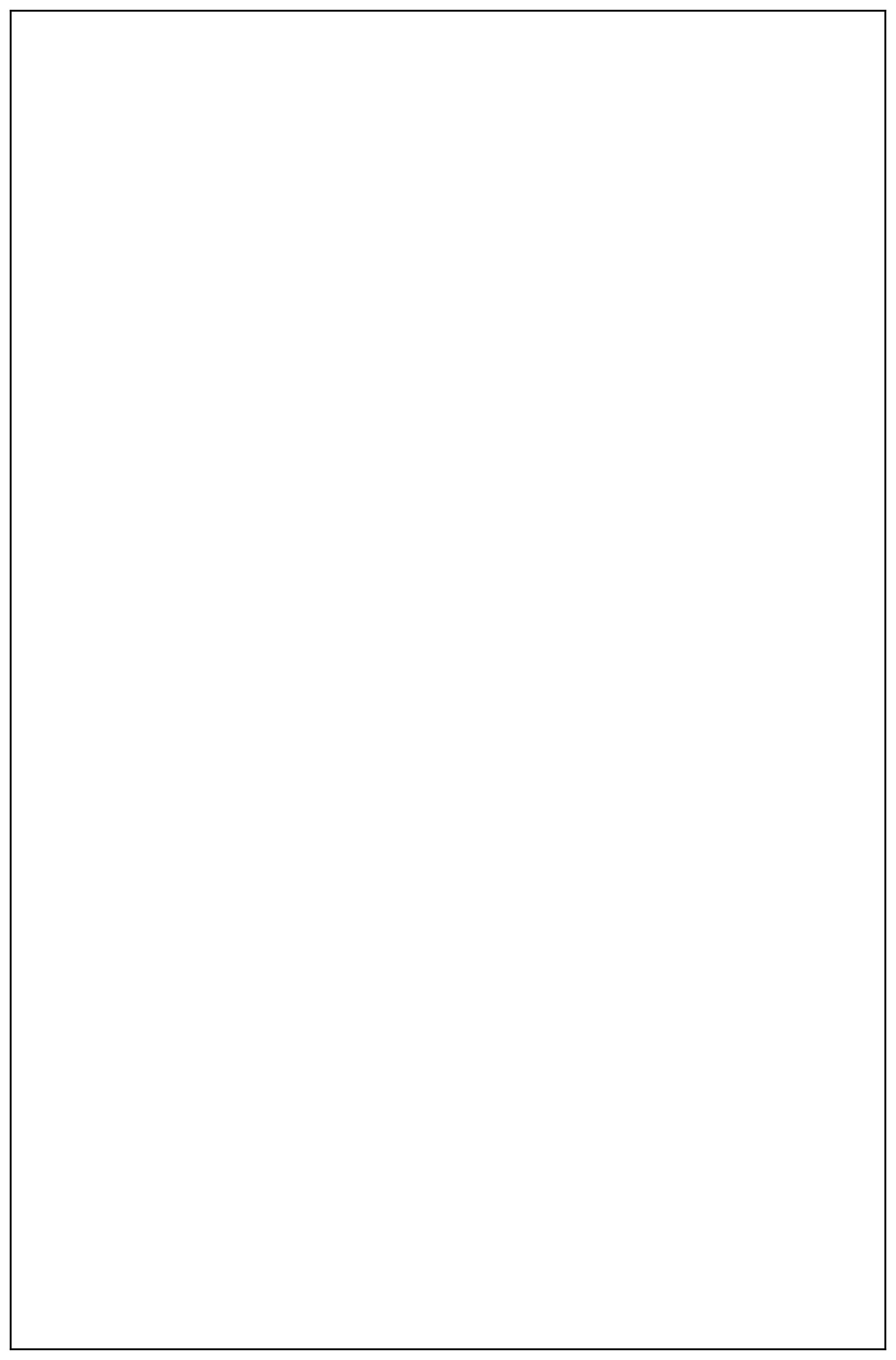
Scanner scanner = new Scanner(System.in);

// Input for Football player

String footballPlayerName = scanner.nextLine();

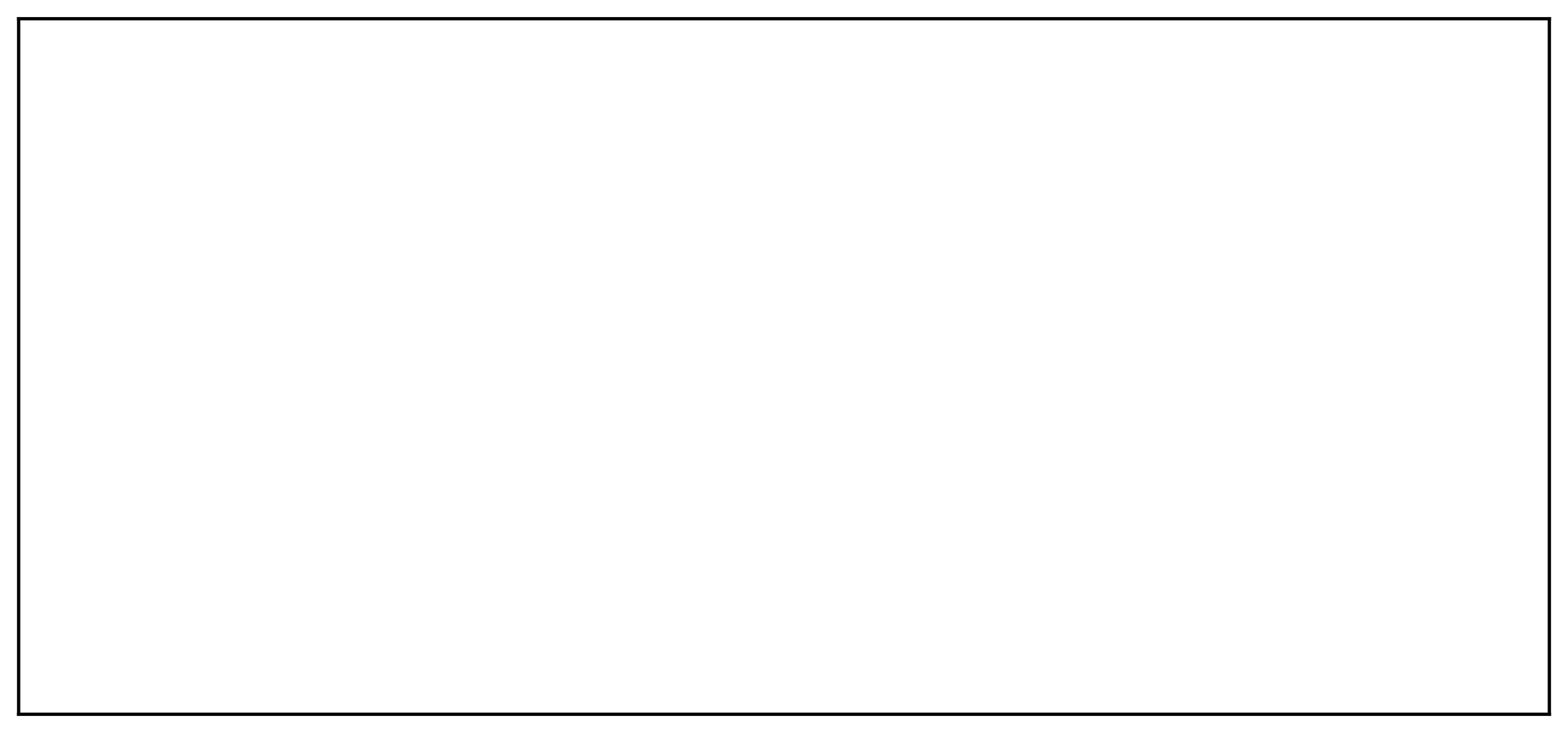
Football footballPlayer = new Football(footballPlayerName);

// Input for Volleyball player



String volleyballPlayerName = scanner.nextLine();

Volleyball volleyballPlayer = new Volleyball(volleyballPlayerName);



//

Input

for

Basketball

player

String

basketballPlayerName

=

scanner.nextLine();

Basketball basketballPlayer = new Basketball(basketballPlayerName);

// Call the play method for each player

footballPlayer.play();

volleyballPlayer.play();

basketballPlayer.play();

scanner.close();

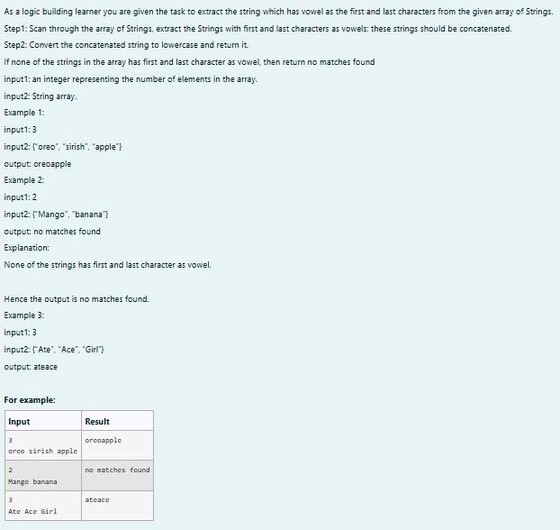
}

}

**OUTPUT :**

[**Lab-08 - Polymorphism, Abstract Classes, final Keyword**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57)

**1.**



**SOLUTION :**

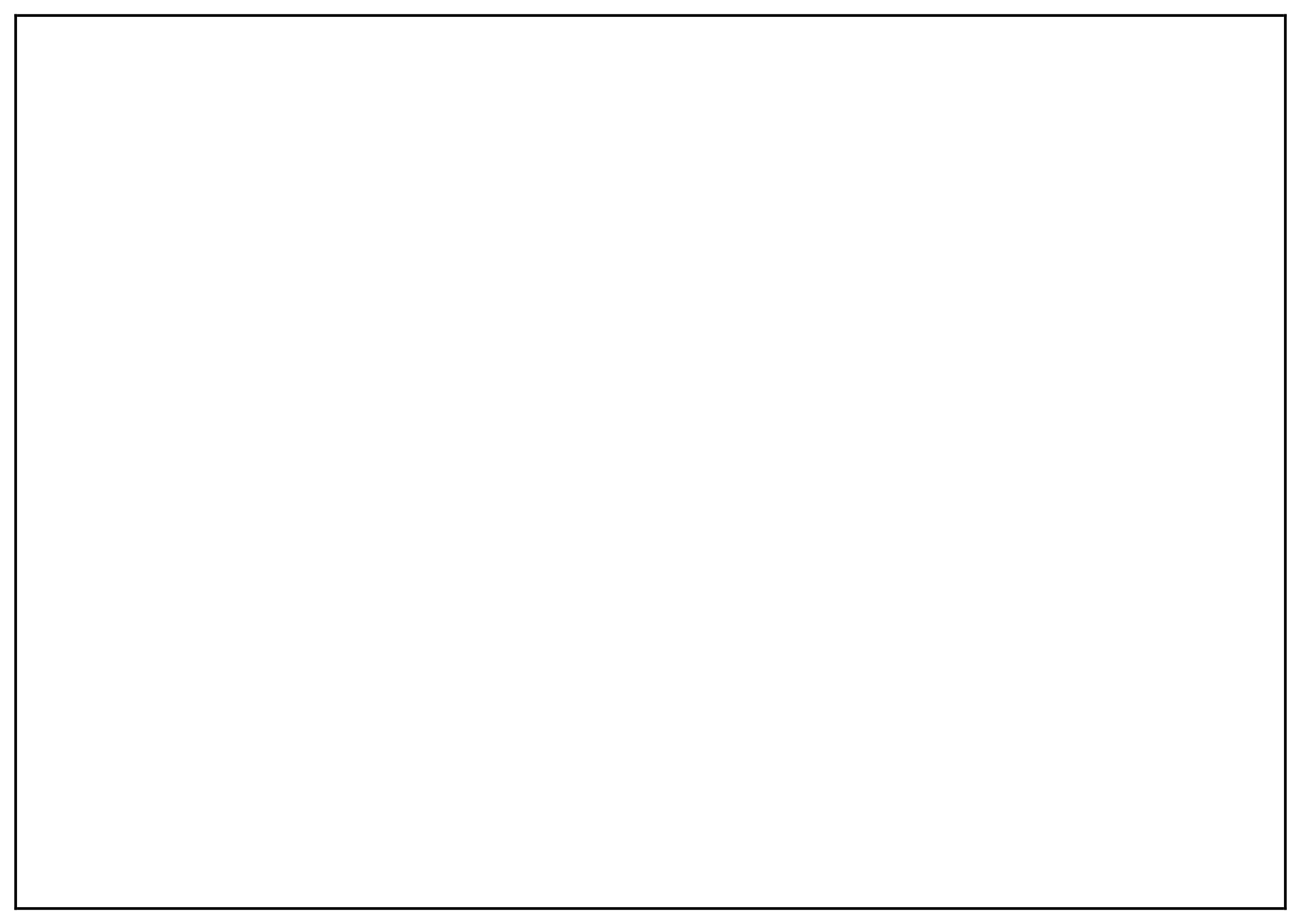
import java.util.Scanner; public class VowelStringExtractor {

// Method to extract strings with vowels as first and last characters public static String extractVowelStrings(String[] stringArray) {

StringBuilder result = new StringBuilder();

String vowels = "aeiouAEIOU"; // String containing all vowels

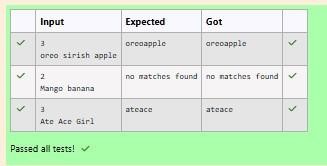
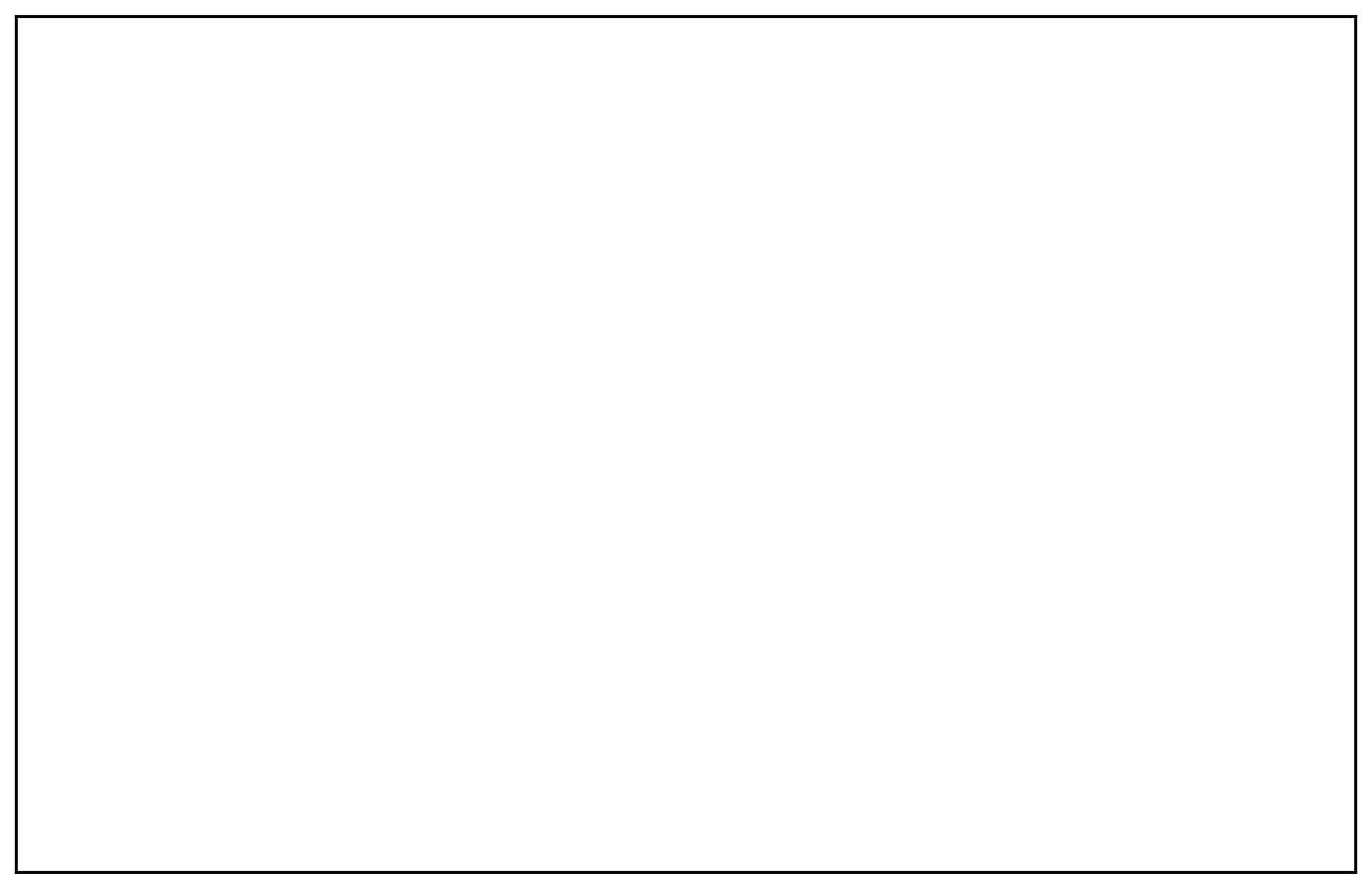
// Iterate through the array of strings for (String s : stringArray) {

// Check if the string is not empty and if both the first and last characters are vowels if (s.length() > 0 && vowels.indexOf(s.charAt(0)) != -1 &&

vowels.indexOf(s.charAt(s.length() - 1)) != -1) { result.append(s); // Append matching string to the result }

}

// Return the concatenated string in lowercase or "no matches found" return result.length() > 0 ? result.toString().toLowerCase() : "no matches found"; }



public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input for the number of strings

int n = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Input for the strings in one line

String input = scanner.nextLine();

String[] strings = input.split(" "); // Split input into an array

// Process and output the result

String result = extractVowelStrings(strings);

System.out.println(result);

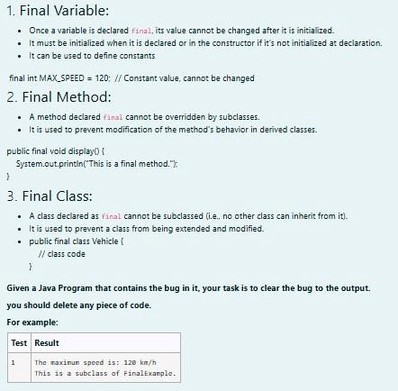
scanner.close(); // Close the scanner

}

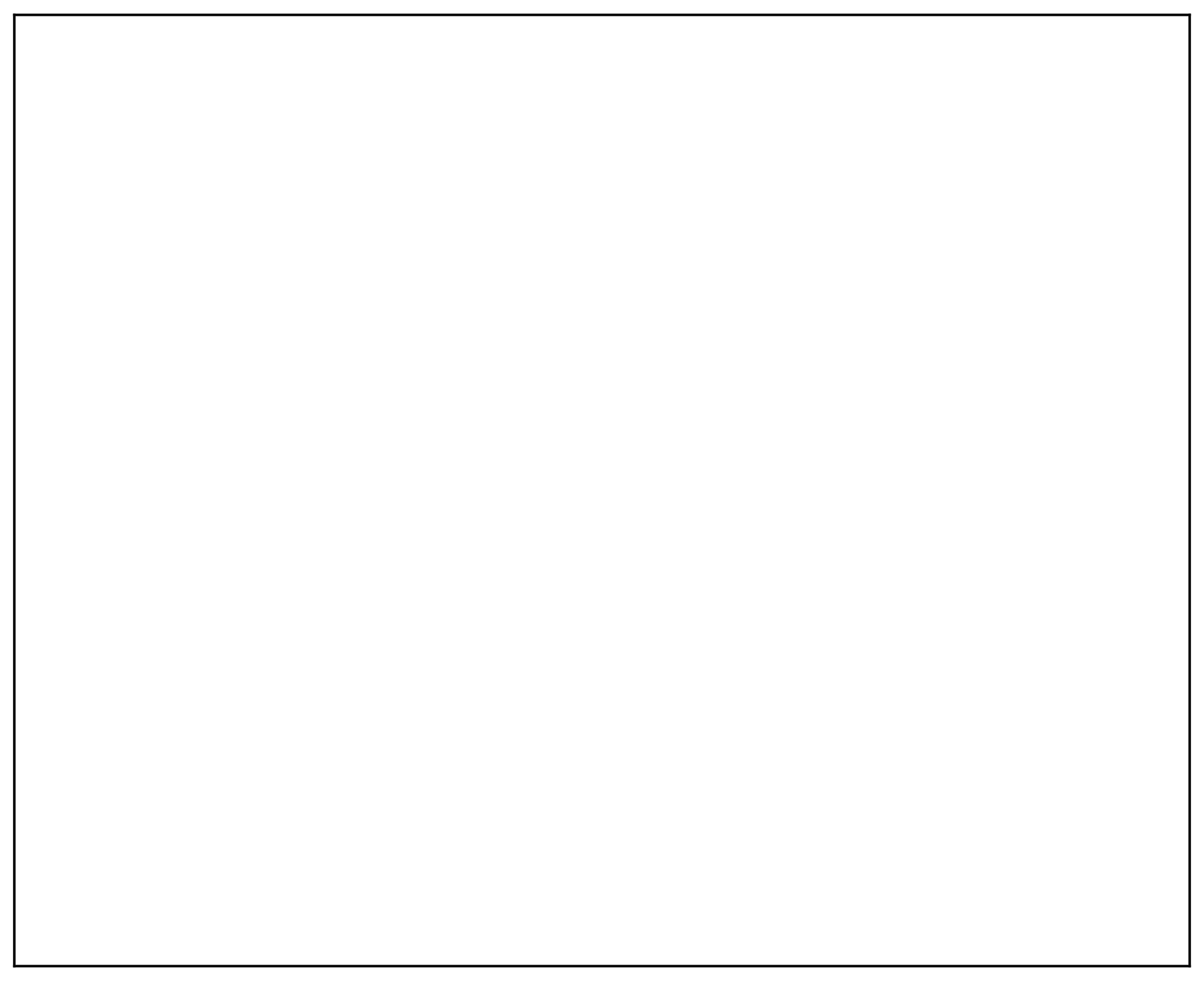
}

**OUTPUT :**

**2.**



**SOLUTION :**



// Final class definition

final class FinalExample {

// Final variable

final int MAX\_SPEED = 120; // Constant value

// Final method

public final void display() {

System.out.println("The maximum speed is: " + MAX\_SPEED + " km/h");

}

}

// Main class to test the final class public

class Test { public static void

main(String[] args) {

// Create an instance of FinalExample

FinalExample example = new FinalExample();

example.display();

// Uncommenting the following line will result in a compile-time error

// because FinalExample is a final class and cannot be subclassed. //

class SubclassExample extends FinalExample { }

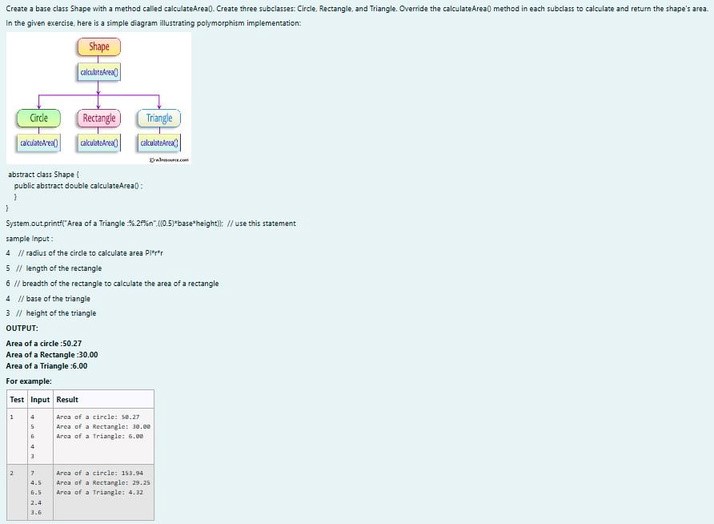
System.out.println("This is a subclass of FinalExample.");

}

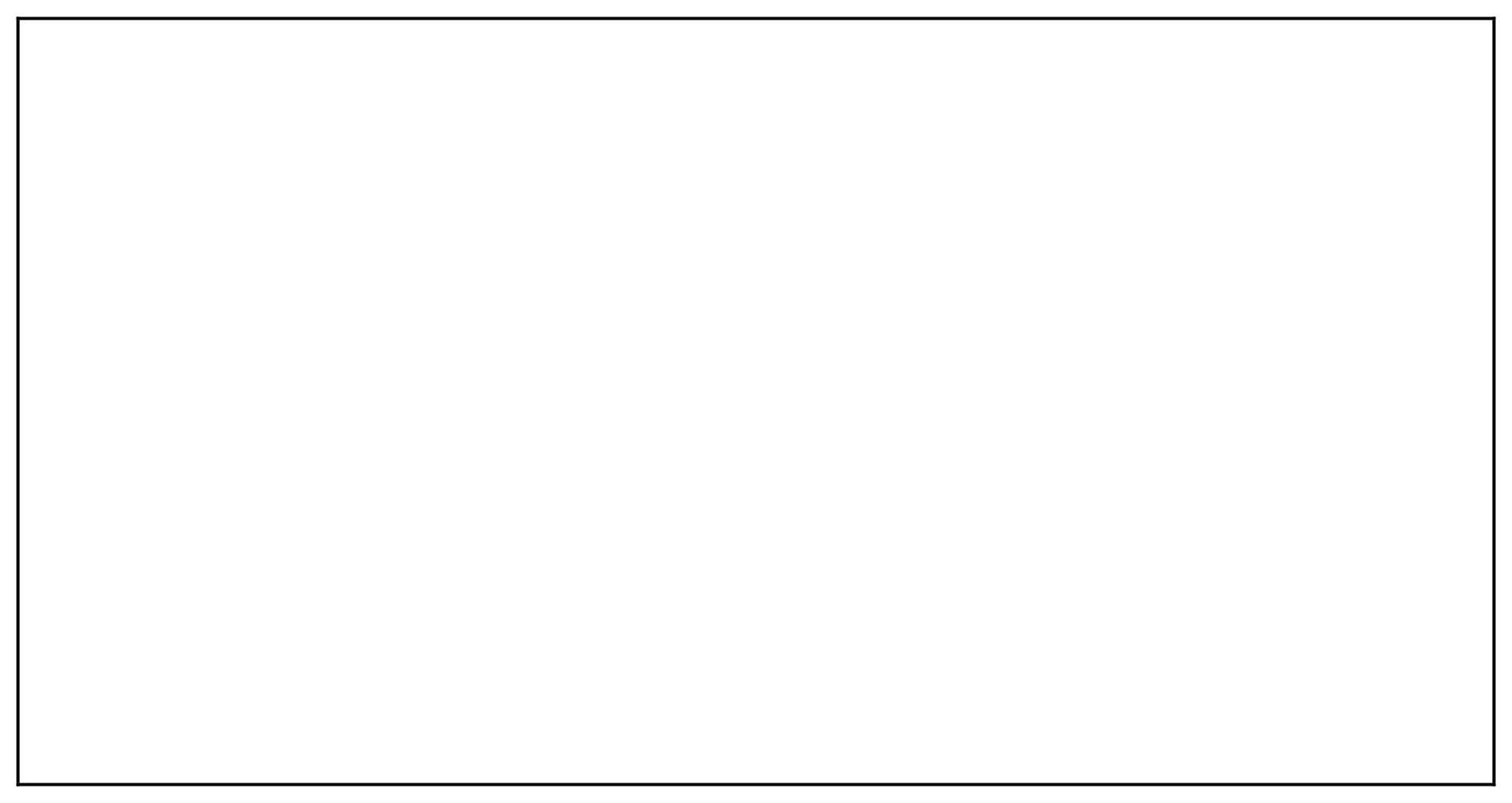
}

**OUTPUT :**

**3.**



**SOLUTION :**



import java.util.Scanner;

// Abstract class Shape abstract class

Shape { public abstract double

calculateArea();

}

// Circle class

class Circle extends Shape {

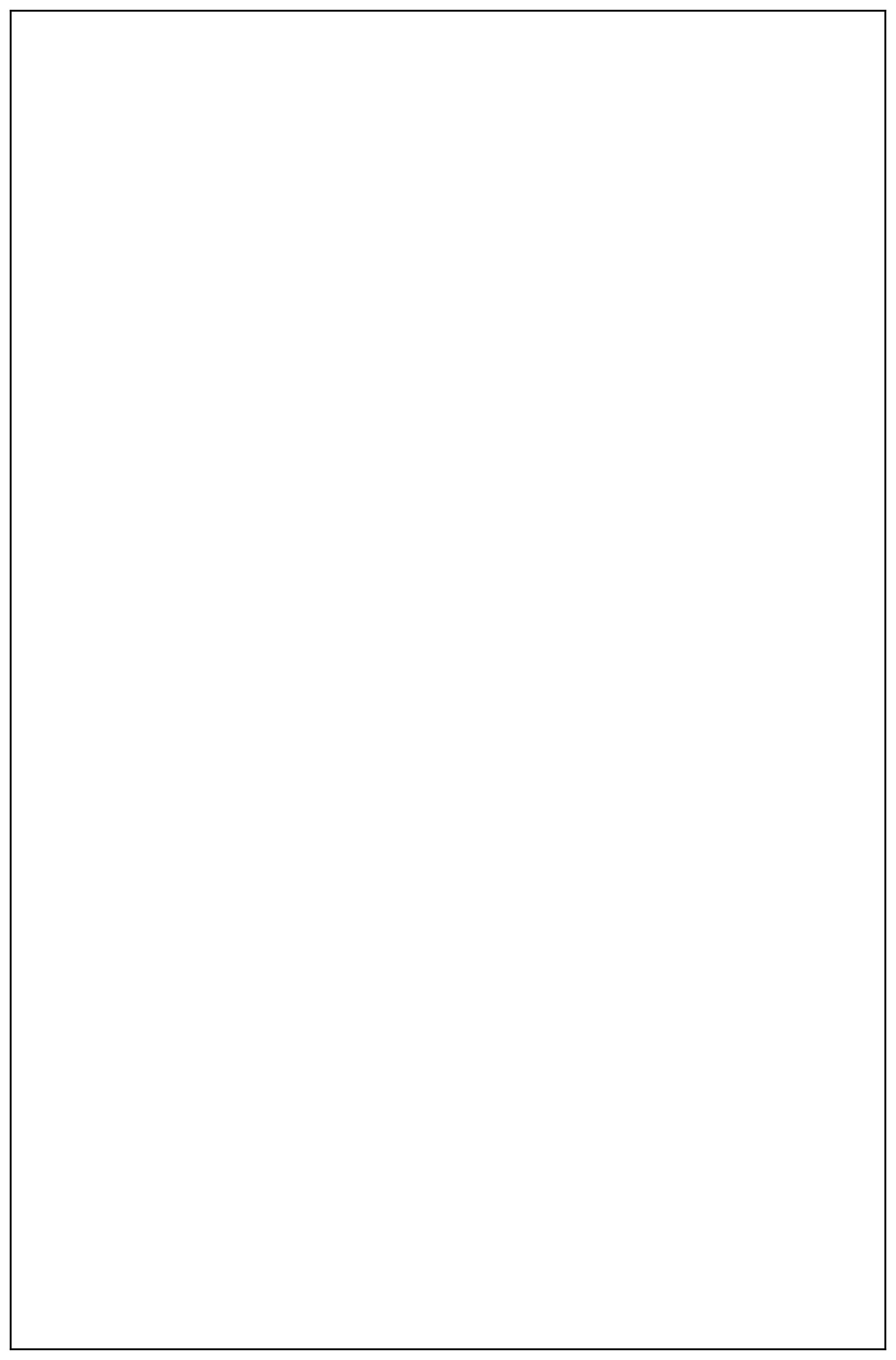
private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override



public double calculateArea() { return Math.PI \* radius

\* radius; // Area of circle: πr² }

}

// Rectangle class

class Rectangle extends Shape {

private double length; private

double breadth;

public Rectangle(double length, double breadth) {

this.length = length; this.breadth = breadth;

}

@Override

public double calculateArea() { return length \* breadth; // Area

of rectangle: length \* breadth

}

}

// Triangle class

class Triangle extends Shape {

private double base; private

double height;

public Triangle(double base, double height) {

this.base = base; this.height = height;

}

@Override

public double calculateArea() { return 0.5 \* base \* height; // Area

of triangle: 0.5 \* base \* height

}

}

// Main class to test the shapes public

class ShapeTest { public static void

main(String[] args) {

Scanner scanner = new Scanner(System.in);

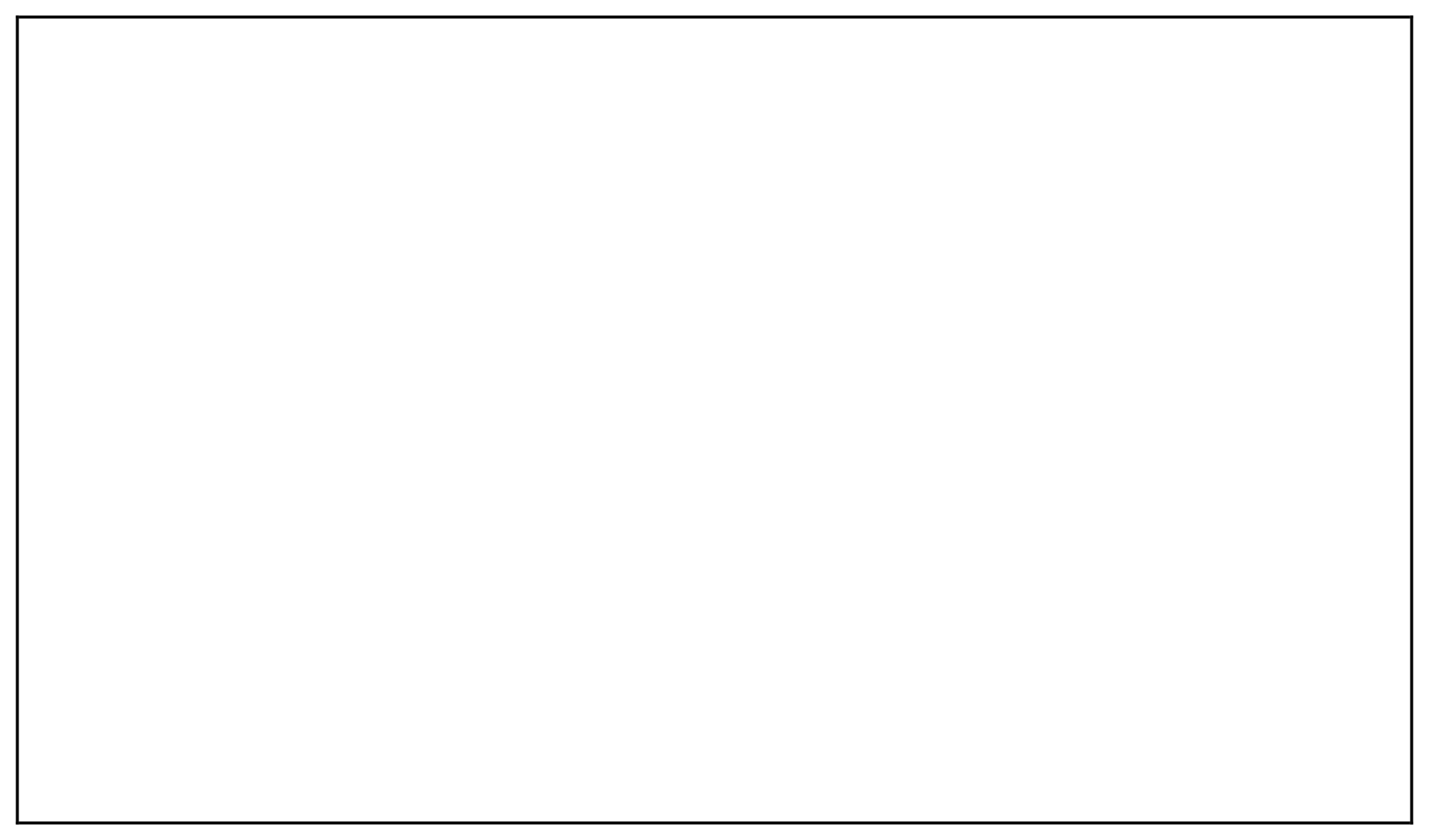
// Input for Circle

double radius = scanner.nextDouble();

Circle circle = new Circle(radius);

System.out.printf("Area of a circle: %.2f%n", circle.calculateArea());

// Input for Rectangle



double length = scanner.nextDouble();

double breadth = scanner.nextDouble();

Rectangle rectangle = new Rectangle(length, breadth);

System.out.printf("Area of a Rectangle: %.2f%n", rectangle.calculateArea());

// Input for Triangle double base =

scanner.nextDouble();

double height = scanner.nextDouble();

Triangle triangle = new Triangle(base, height);

System.out.printf("Area of a Triangle: %.2f%n", triangle.calculateArea());

scanner.close();

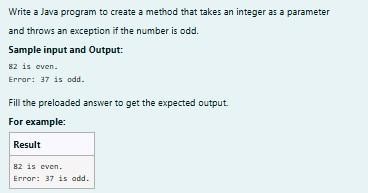
}

}

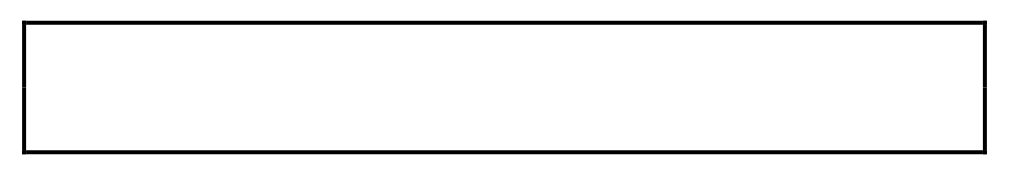
**OUTPUT :**

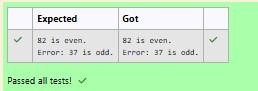
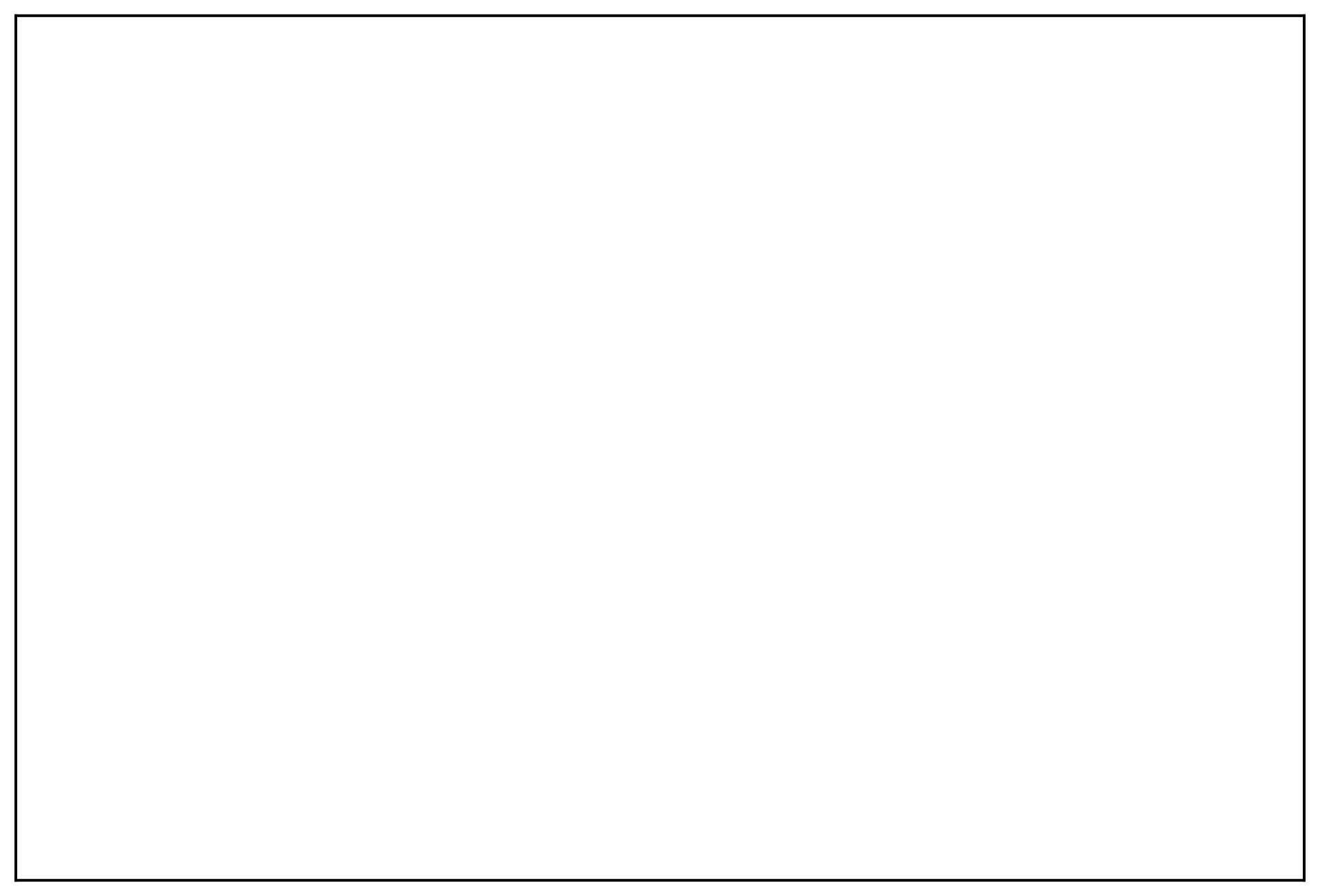
[**Lab-09-Exception Handling**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=59)

**1.**



**SOLUTION :**

class prog { public static void main(String[] args) {



**OUTPUT :**

int n = 82;

trynumber(n);

n = 37;

trynumber(n); // Call the trynumber(n);

}

public static void trynumber(int n) { try {

checkEvenNumber(n); // Call the checkEvenNumber()

System.out.println(n + " is even.");

}

catch (Exception e) { // Catch the exception

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

}

}

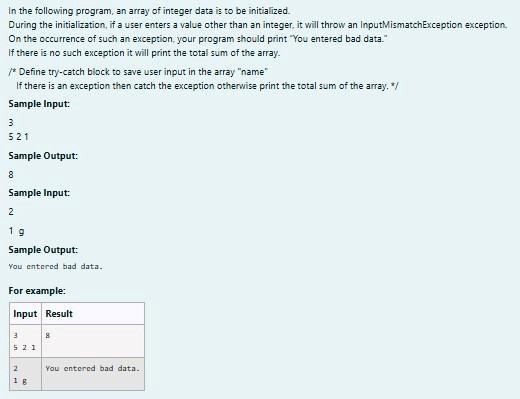
public static void checkEvenNumber(int number) { if (number % 2 != 0) { throw new

RuntimeException(number + " is odd."); // Throw a RuntimeException }

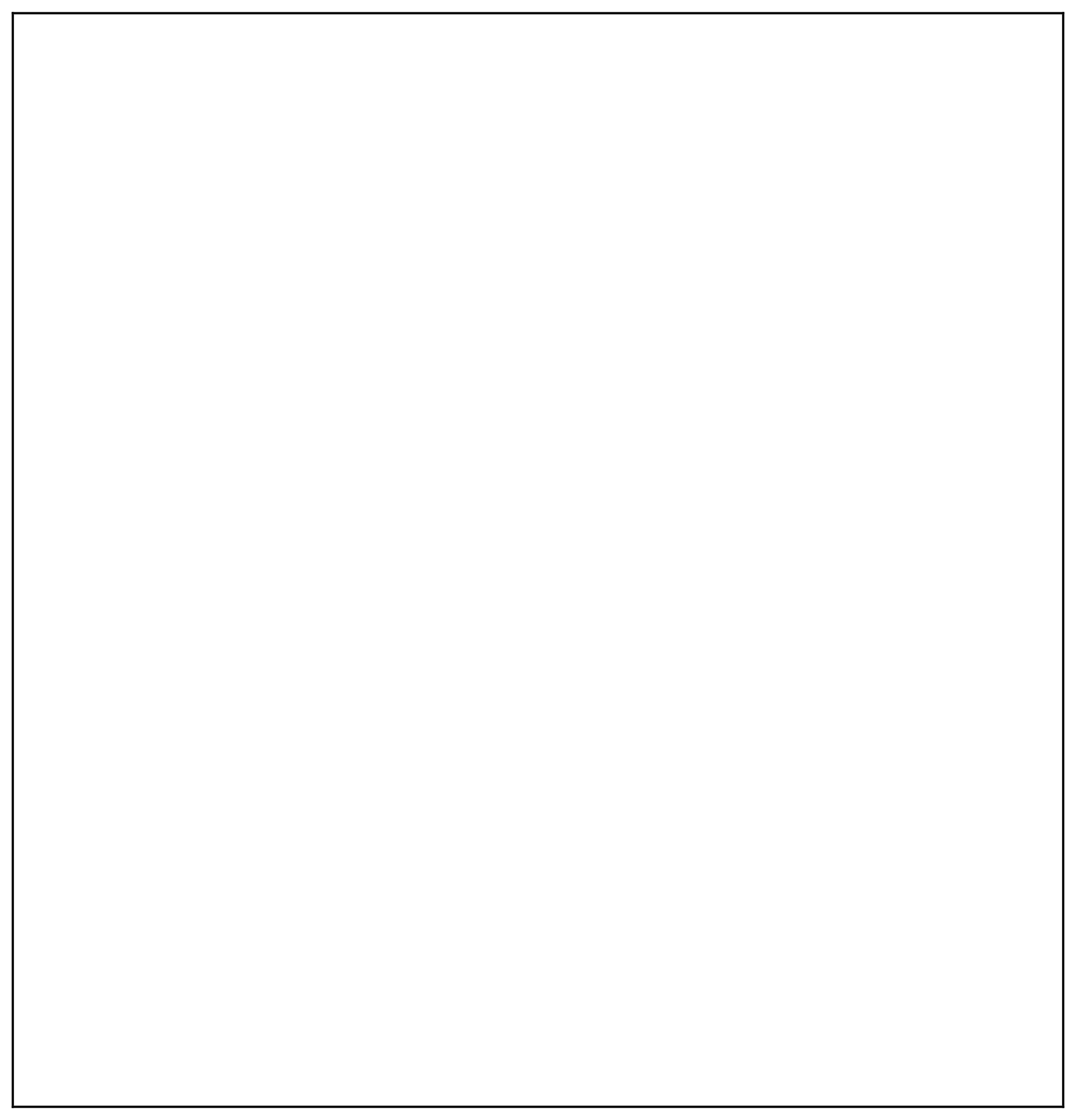
**}**

**}**

**2.**



**SOLUTION :**



import java.util.Scanner;

import java.util.InputMismatchException;

class prog { public static void

main(String[] args) { Scanner sc = new

Scanner(System.in); int length =

sc.nextInt();

// create an array to save user input int[]

name = new int[length]; int sum = 0; // save

the total sum of the array.

/\* Define try-catch block to save user input in the array "name"

If there is an exception then catch the exception otherwise print

the total sum of the array. \*/

try { for (int i = 0; i < length; i++) { name[i] =

sc.nextInt(); // save user input in the array

}

// Calculate the total sum

for (int num : name) {

sum += num;

}

// Print the total sum

System.out.println(sum);

}

catch (InputMismatchException e)

{

System.out.println("You entered bad data.");

}

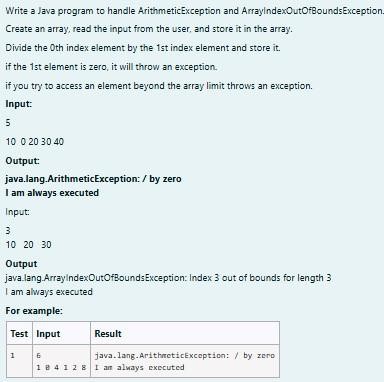
sc.close(); // Close the scanner

}

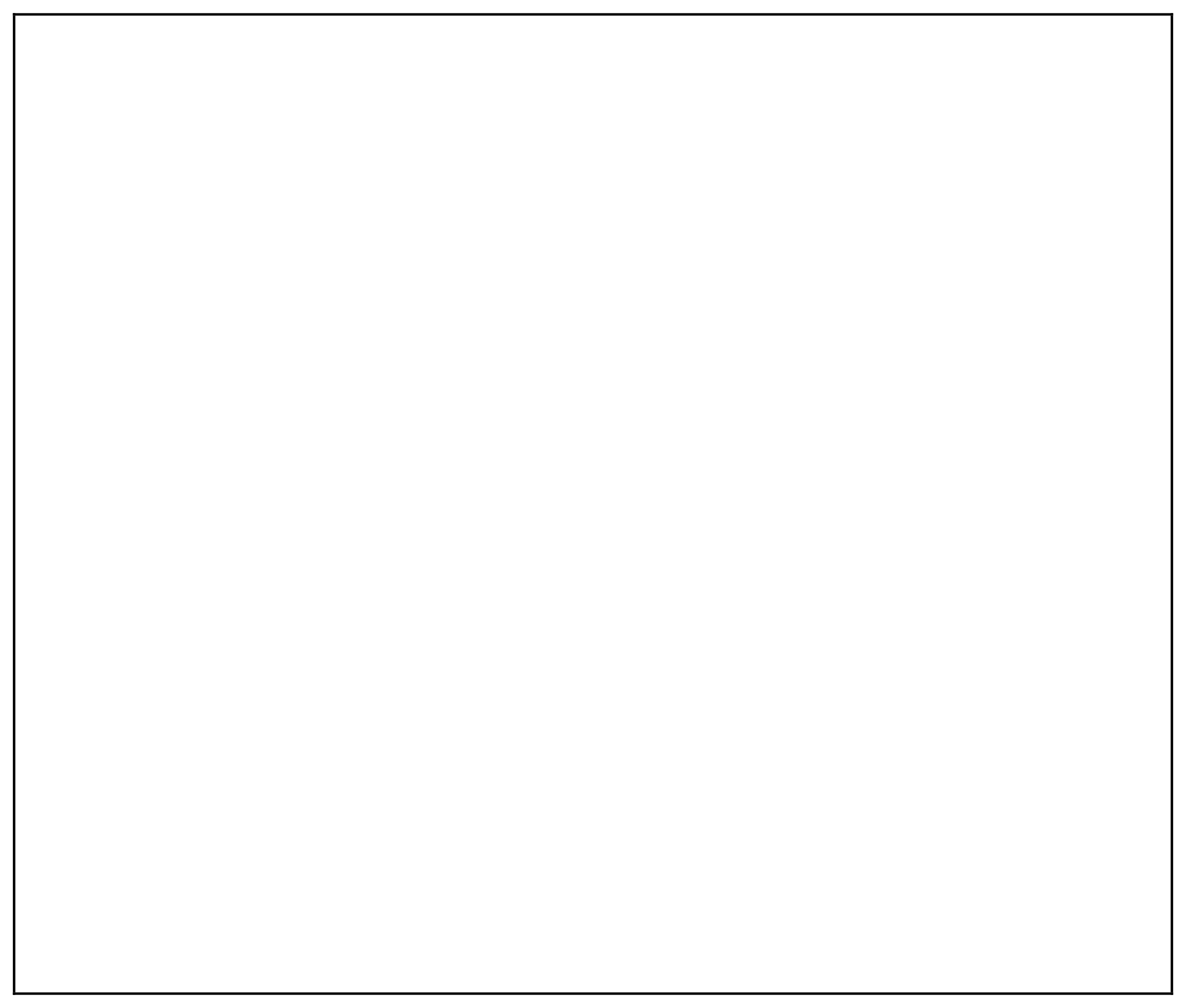
}

**OUTPUT :**

**3.**



**SOLUTION :**



import java.util.Scanner;

public class ExceptionHandlingExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Read the size of the array

int size = scanner.nextInt();

// Initialize the array int[]

numbers = new int[size];

// Read the elements into the array

for (int i = 0; i < size; i++) {

numbers[i] = scanner.nextInt();

}

try {

// Attempt to perform division

int result = numbers[0] / numbers[1]; // This may cause an ArithmeticException

}

catch (ArithmeticException e)

{

System.out.println(e); // Catch division by zero

}

catch (ArrayIndexOutOfBoundsException e)

{

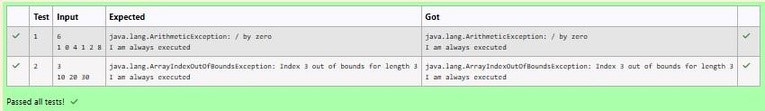
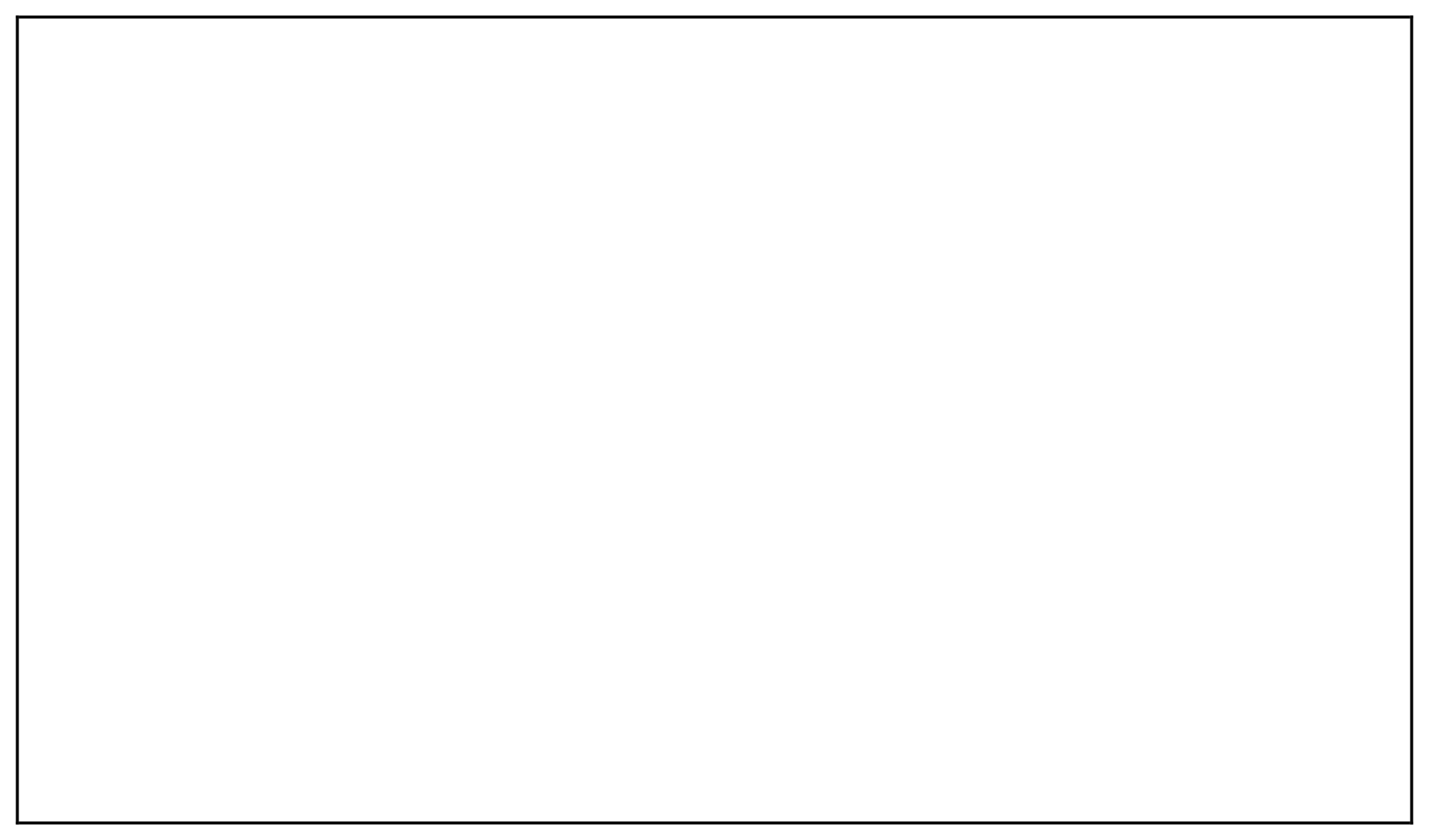
System.out.println(e); // Catch accessing out of bounds

}

catch (Exception e)

{

System.out.println(e); // Catch any other exceptions



}

finally

{

// This block is always executed

}

try {

// Attempt to access an out-of-bounds index

int outOfBoundsValue = numbers[3]; // This will trigger

ArrayIndexOutOfBoundsException if size < 4

}

catch (ArrayIndexOutOfBoundsException e)

{

System.out.println(e);

}

finally

{

// This block is always executed for the second try

System.out.println("I am always executed");

}

scanner.close();

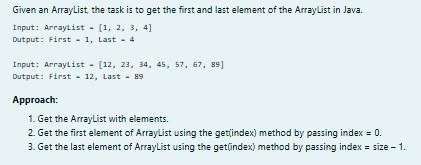
}

}

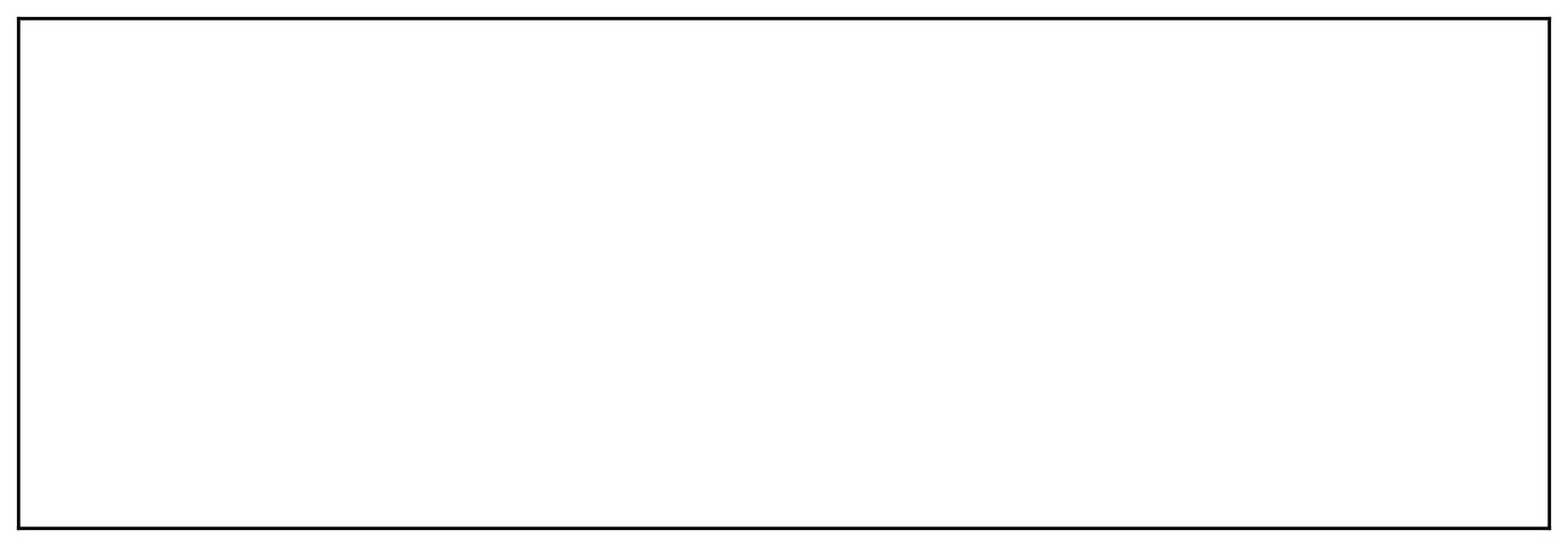
**OUTPUT :**

[**Lab-10- Collection- List**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=60)

**1.**



**SOLUTION :**



import java.util.ArrayList;

import java.util.Scanner;

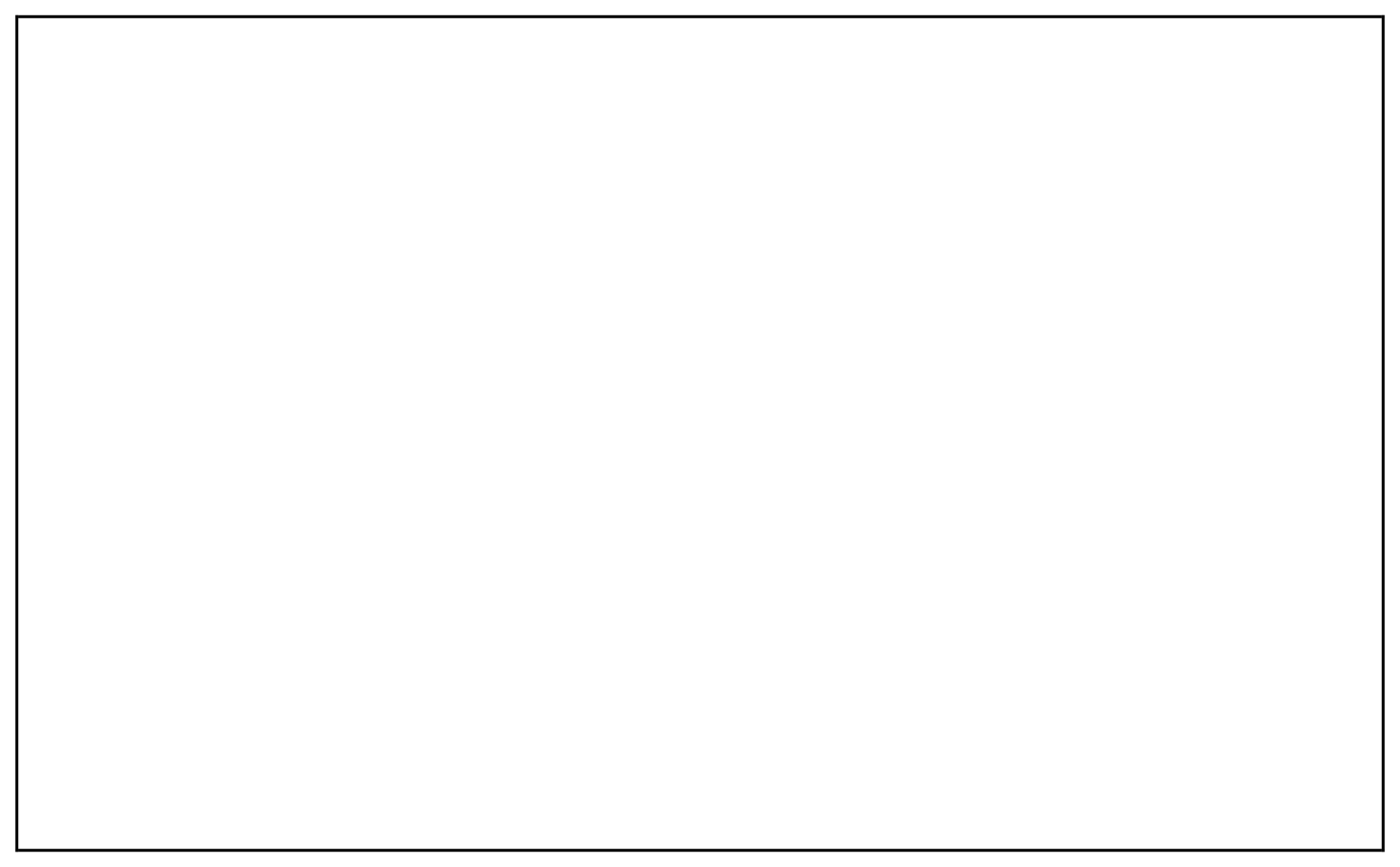
public class FirstAndLastElement {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Create an ArrayList

ArrayList<Integer> numbers = new ArrayList<>();



int numElements = scanner.nextInt();

for (int i = 0; i < numElements; i++) {

int number = scanner.nextInt();

numbers.add(number);

}

System.out.println("ArrayList: " + numbers);

// Get the first element int

firstElement = numbers.get(0);

// Get the last element

int lastElement = numbers.get(numbers.size() - 1);

// Print the results

System.out.print("First : " + firstElement);

System.out.println(", Last : " + lastElement);

}

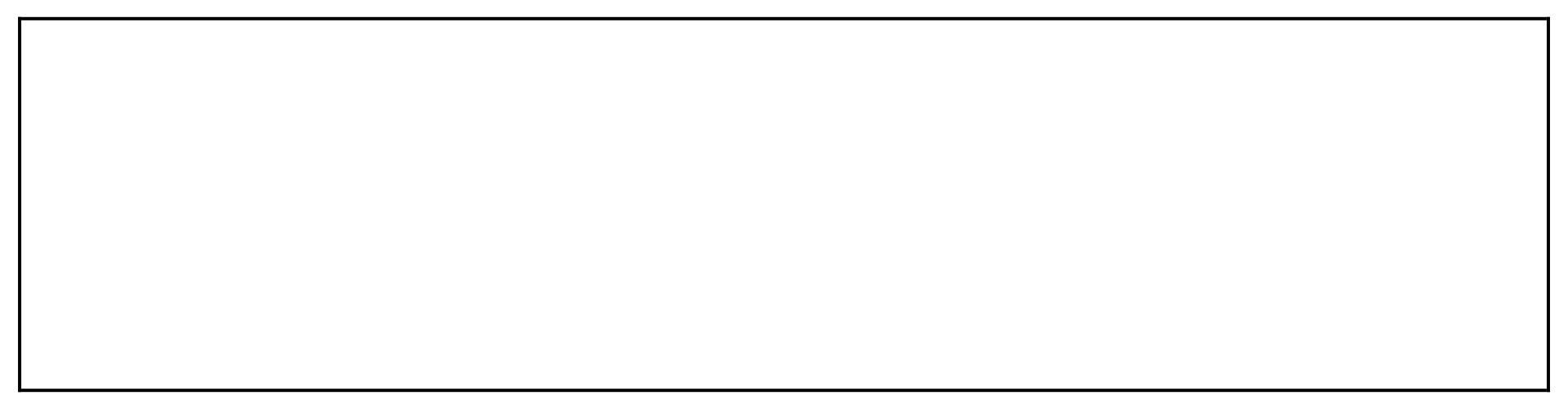
}

**OUTPUT :**

**2.**



**SOLUTION :**



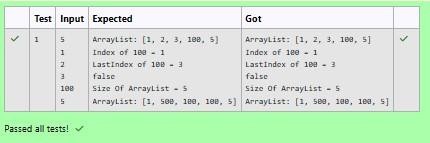
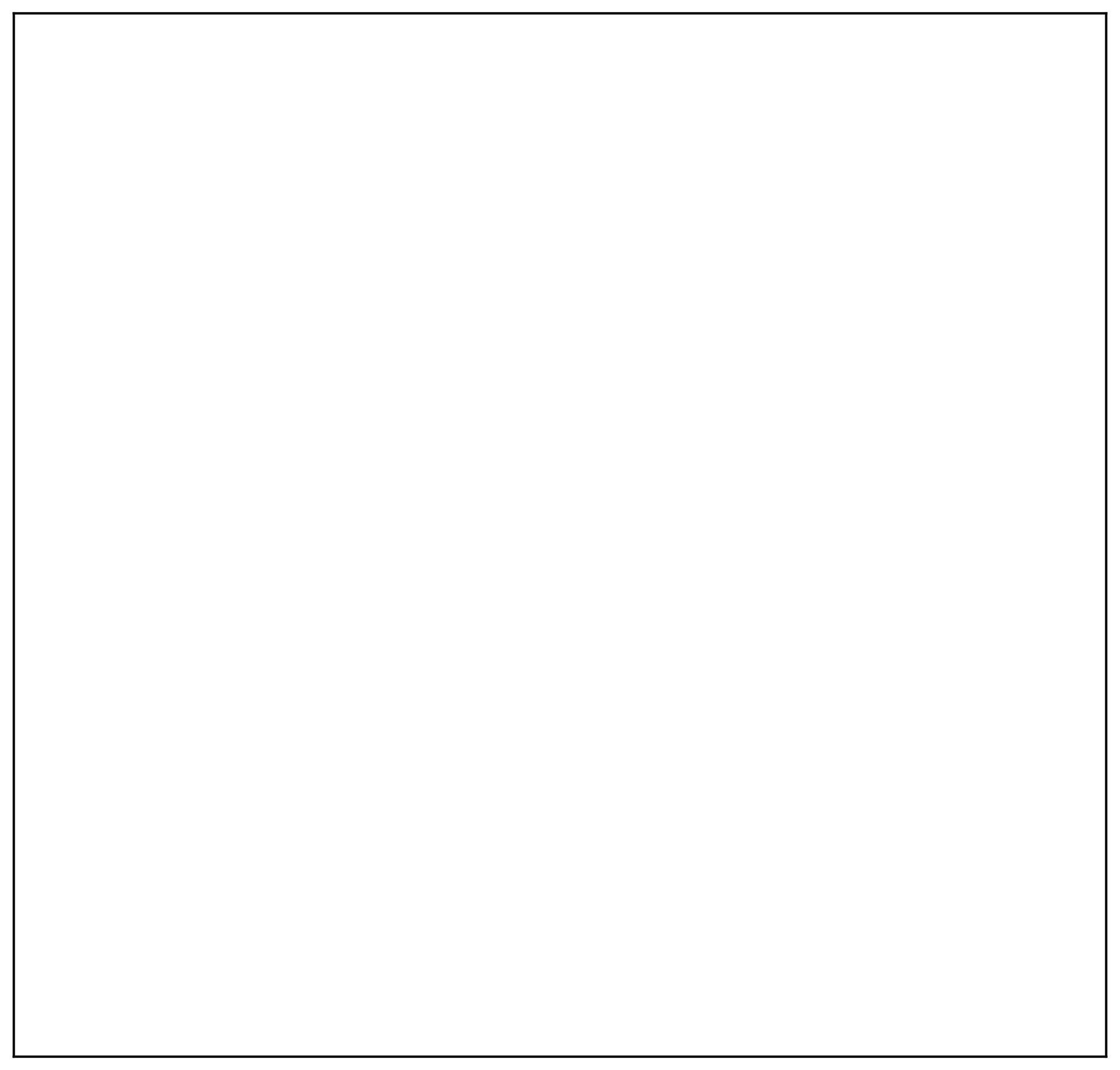
import java.util.ArrayList;

import java.util.Scanner;

public class Prog {

public static void main(String[] args)

{



Scanner sc= new Scanner(System.in);

int n = sc.nextInt();

ArrayList<Integer> list = new ArrayList<Integer>();

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

list.add(sc.nextInt());

// printing initial value ArrayList

System.out.println("ArrayList: " + list);

//Replacing the element at index 1 with 100

list.set(1,100);

//Getting the index of first occurrence of 100

System.out.println("Index of 100 = "+ list.indexOf(100)

//Getting the index of last occurrence of 100

)

;

System.out.println("LastIndex of 100 = "+ list.lastIndexOf(100));

// Check whether 200 is in the list or not

System.out.println(list.contains(200)); //Output : false

// Print ArrayList size

System.out.println("Size Of ArrayList = "+list.size() );

//Inserting 500 at index 1

list.add(1,500);

// code here

//Removing an element from position 3

list.remove(3);

// code here

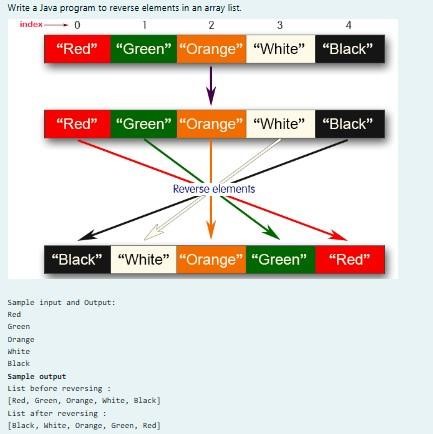
System.out.print("ArrayList: " + list);

}

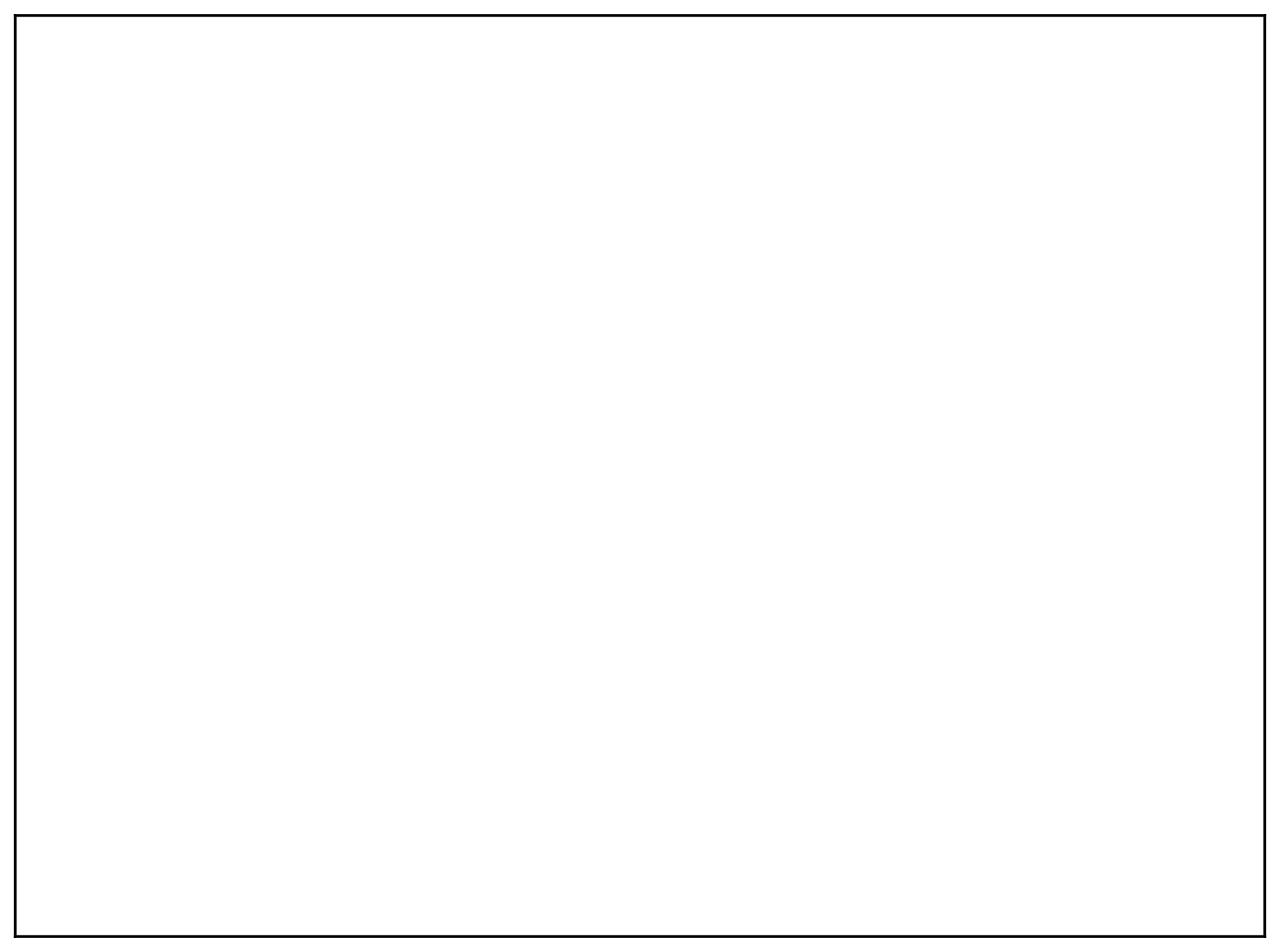
}

**OUTPUT :**

**3.**



**SOLUTION :**



import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class ReverseArrayList { public

static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

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

String element = scanner.next();

list.add(element);

}

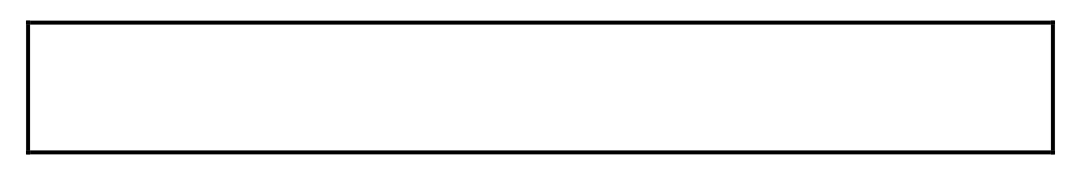
System.out.println("List before reversing : ");

System.out.println(list);

Collections.reverse(list);

System.out.println("List after reversing : ");

System.out.println(list);



}

}

**OUTPUT :**

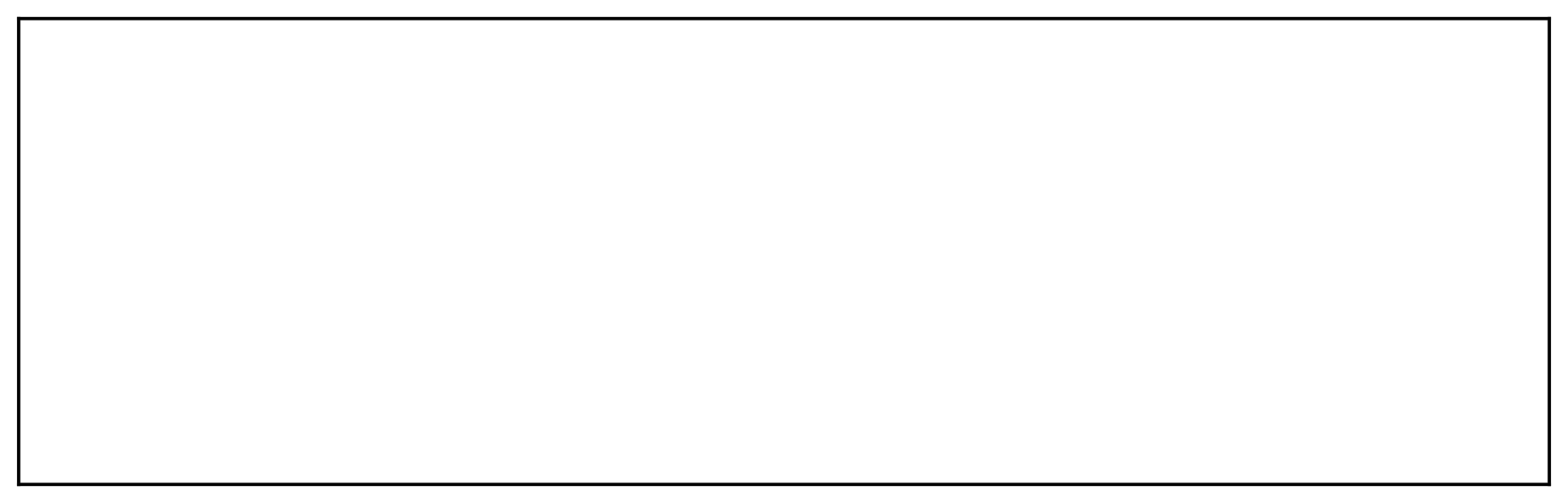


[**Lab-11-Set, Map**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=61)

**1.**



**SOLUTION :**



**import java.util.HashSet;**

**import java.util.Scanner;**

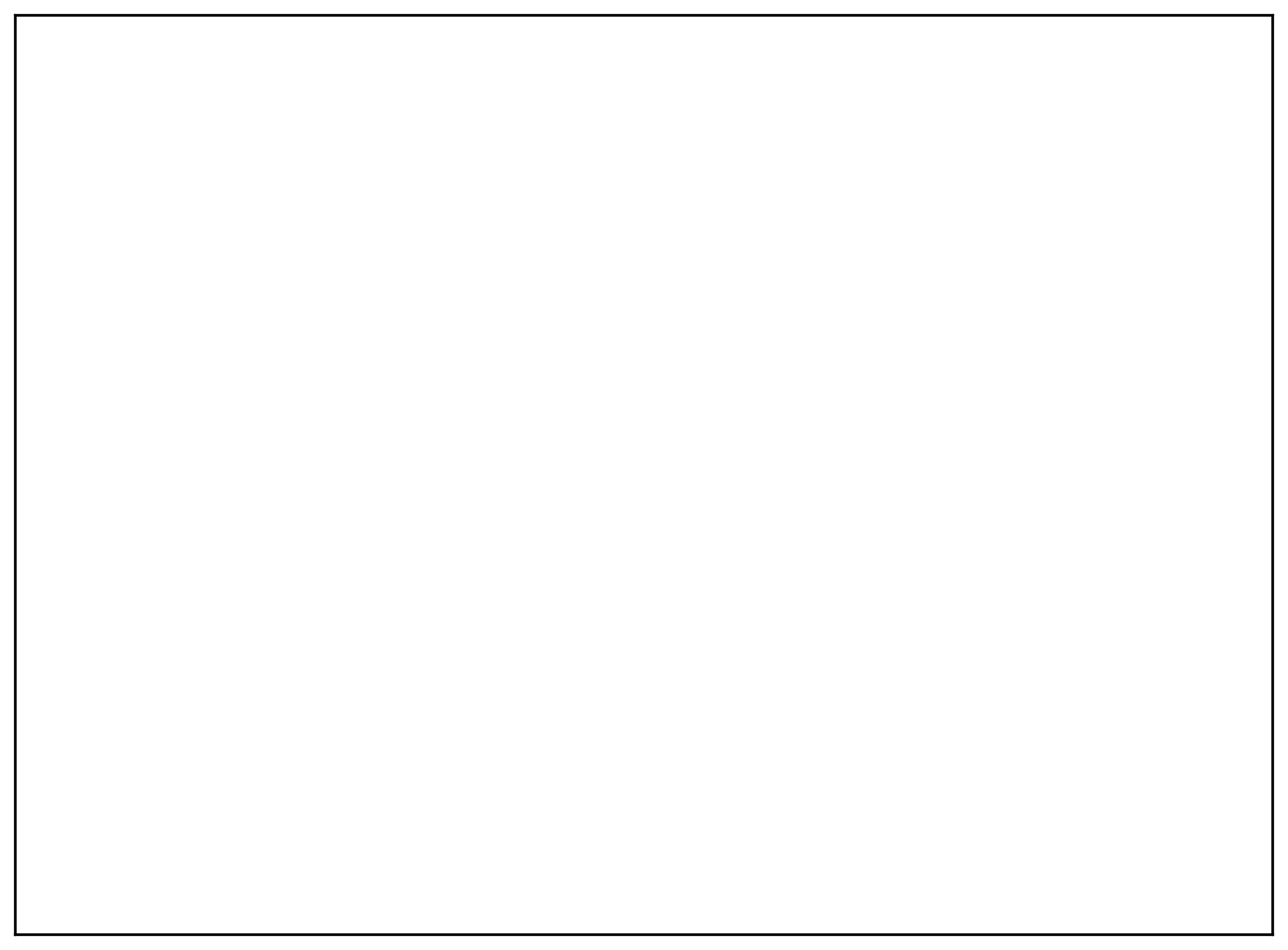
**public class Prog { public static void**

**main(String[] args) {**

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

**// Read the number of elements**

**int n = sc.nextInt();**



**// Create a HashSet object to store numbers**

**HashSet<Integer> numbers = new HashSet<>();**

**// Add numbers to the HashSet**

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

**numbers.add(sc.nextInt());**

**}**

**// Read the search key**

**int skey = sc.nextInt();**

**// Check if skey is present in the HashSet**

**if (numbers.contains(skey)) {**

**System.out.println(skey + " was found in the set.");**

**}**

**else**

**{**

**System.out.println(skey + " was not found in the set.");**

**}**

**// Close the scanner**

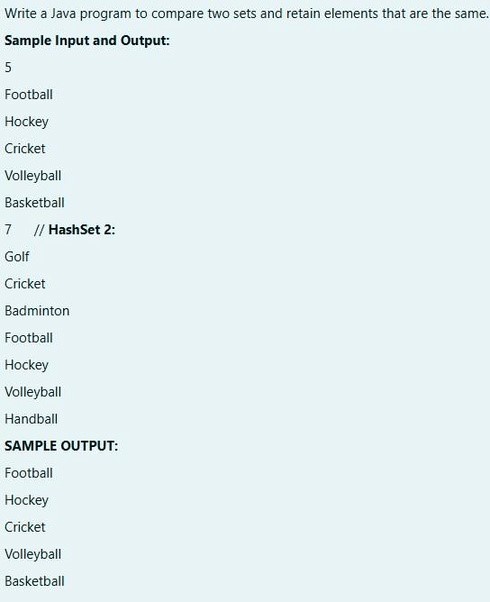
**sc.close();**

**}**

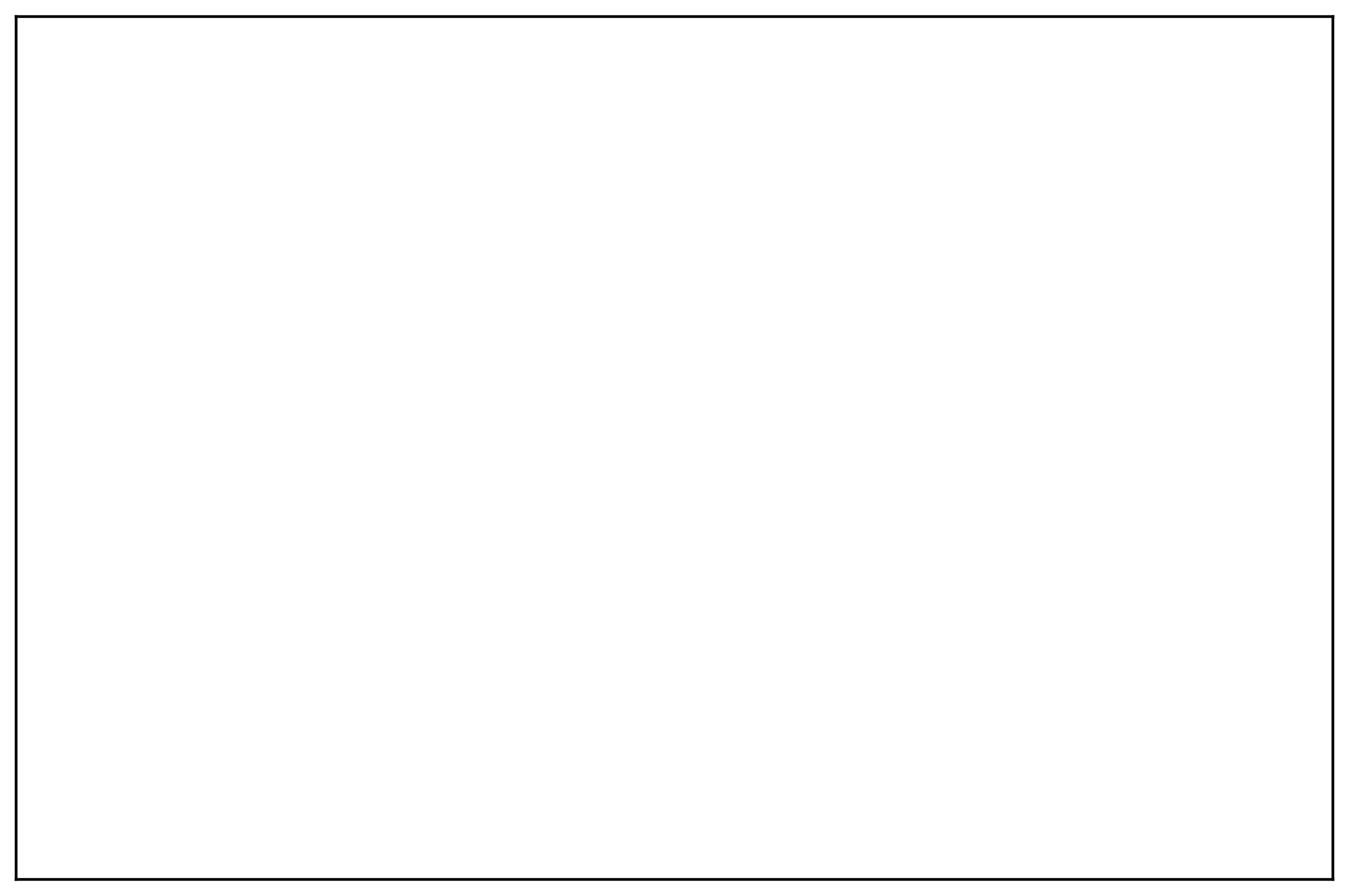
**}**

**OUTPUT :**

**2.**



**SOLUTION :**



**import java.util.HashSet;**

**import java.util.Scanner;**

**import java.util.Set;**

**public class CompareSets { public static**

**void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**// Read the size of the first set**

**int size1 = Integer.parseInt(scanner.nextLine());**

**// Create a HashSet to store the first set of elements**

**Set<String> set1 = new HashSet<>();**

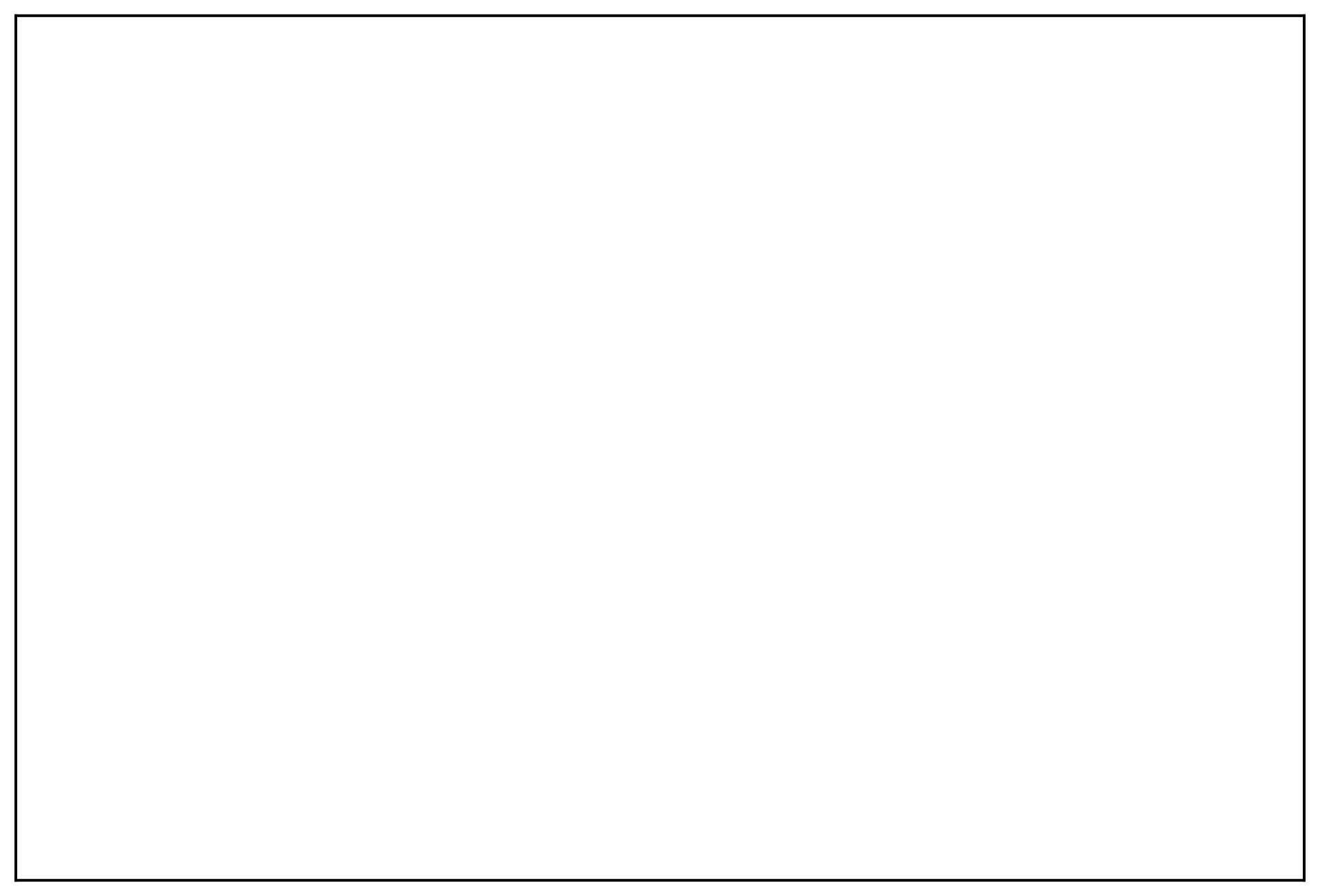
**// Read elements for the first set**

**for (int i = 0; i < size1; i++) {**

**set1.add(scanner.nextLine());**

**}**

**// Read the size of the second set**



**int size2 = Integer.parseInt(scanner.nextLine());**

**// Create a HashSet to store the second set of elements**

**Set<String> set2 = new HashSet<>();**

**// Read elements for the second set**

**for (int i = 0; i < size2; i++) {**

**set2.add(scanner.nextLine());**

**}**

**// Retain common elements using the retainAll() method**

**set1.retainAll(set2);**

**// Print the common elements**

**for (String element : set1) {**

**System.out.println(element);**

**}**

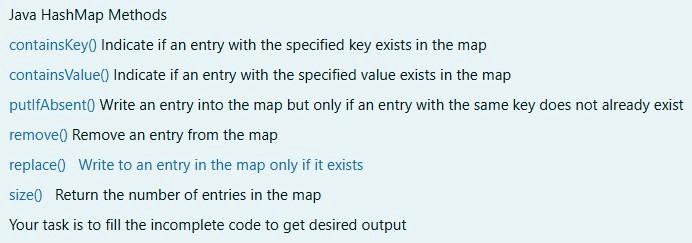
**scanner.close();**

**}**

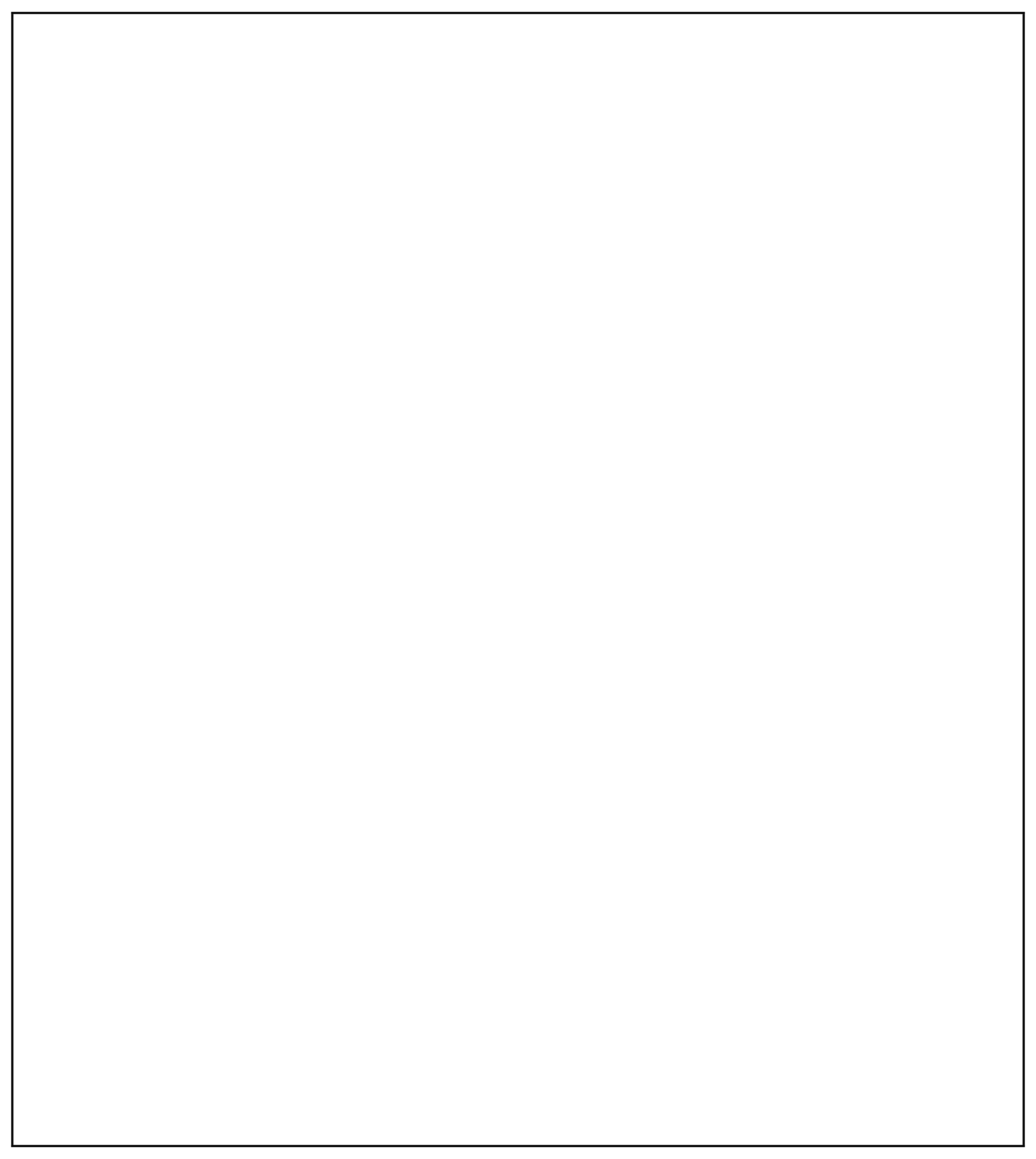
**}**

**OUTPUT :**

**3.**



**SOLUTION :**



**import java.util.HashMap;**

**import**

**java.util.Map.Entry;**

**import java.util.Scanner;**

**import java.util.Set; public**

**class Prog {**

**public static void main(String[] args) {**

**// Creating HashMap with default initial capacity and load factor**

**HashMap<String, Integer> map = new HashMap<String, Integer>();**

**String name;**

**int num;**

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

**int n = sc.nextInt();**

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

**name = sc.next(); num**

**=**

**sc.nextInt();**

**map.put(name, num);**

**}**

**// Printing key-value pairs**

**Set<Entry<String, Integer>> entrySet = map.entrySet();**

**for (Entry<String, Integer> entry : entrySet) {**

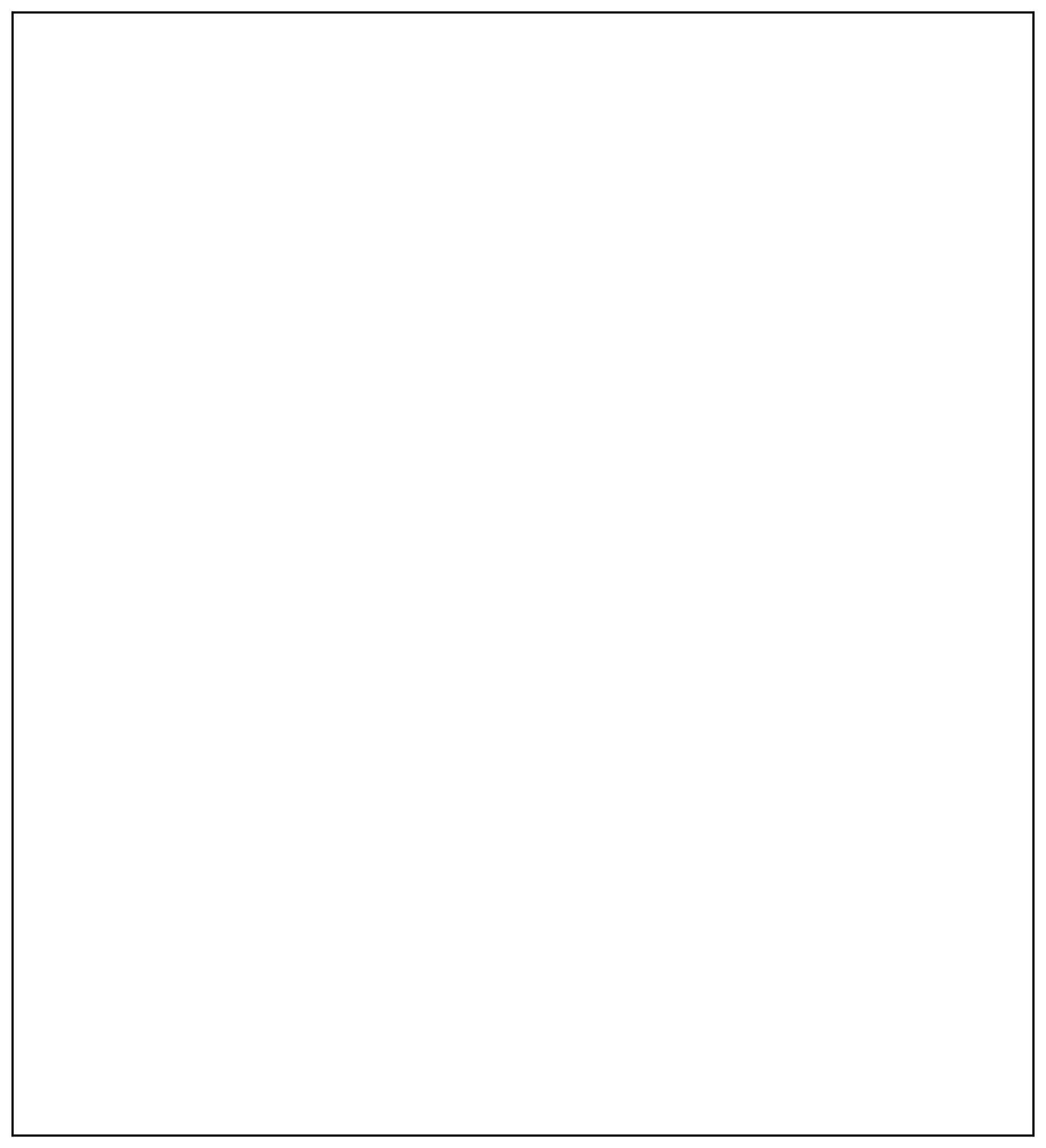
**System.out.println(entry.getKey() + " : " + entry.getValue());**

**}**

**System.out.println("----------");**

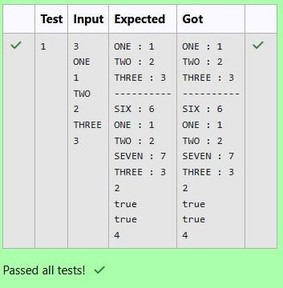
**// Creating another HashMap**

**HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();**



**// Inserting key-value pairs to anotherMap using put() method**

**anotherMap.put("SIX", 6);**



**anotherMap.put("SEVEN", 7);**

**// Inserting key-value pairs of map to anotherMap using putAll() method**

**anotherMap.putAll(map); // This line fills in the missing code**

**// Printing key-value pairs of anotherMap entrySet**

**=**

**anotherMap.entrySet();**

**for (Entry<String, Integer> entry : entrySet) {**

**System.out.println(entry.getKey() + " : " + entry.getValue());**

**}**

**// Adds key-value pair 'FIVE-5' only if it is not present in map**

**map.putIfAbsent("FIVE", 5);**

**// Retrieving a value associated with key 'TWO'**

**int value = map.get("TWO");**

**System.out.println(value); // Prints the value associated with key "TWO" (if it**

**exists)**

**// Checking whether key 'ONE' exists in map**

**System.out.println(map.containsKey("ONE")); // Prints true if "ONE" is a key,**

**false otherwise**

**// Checking whether value '3' exists in map**

**boolean valueExists = map.containsValue(3); // You can use a variable to store**

**the result**

**System.out.println(valueExists); // Prints true if value 3 exists in the map, false**

**otherwise**

**// Retrieving the number of key-value pairs present in map**

**System.out.println(map.size()); // Prints the number of entries in the map**

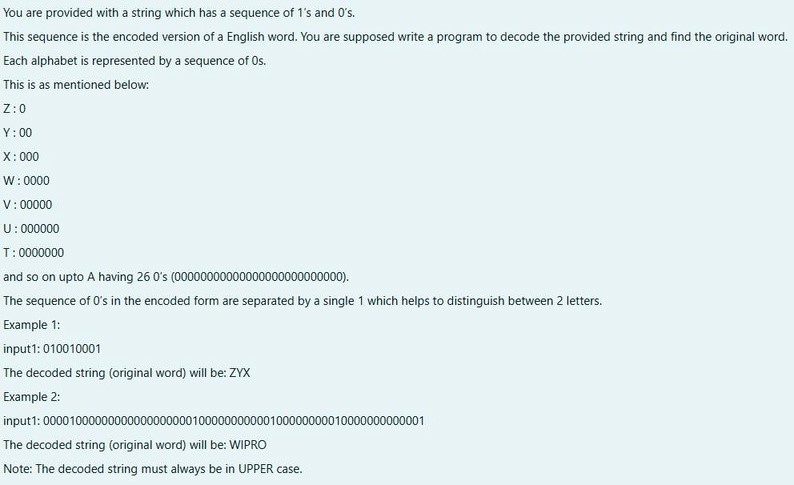
**}**

**}**

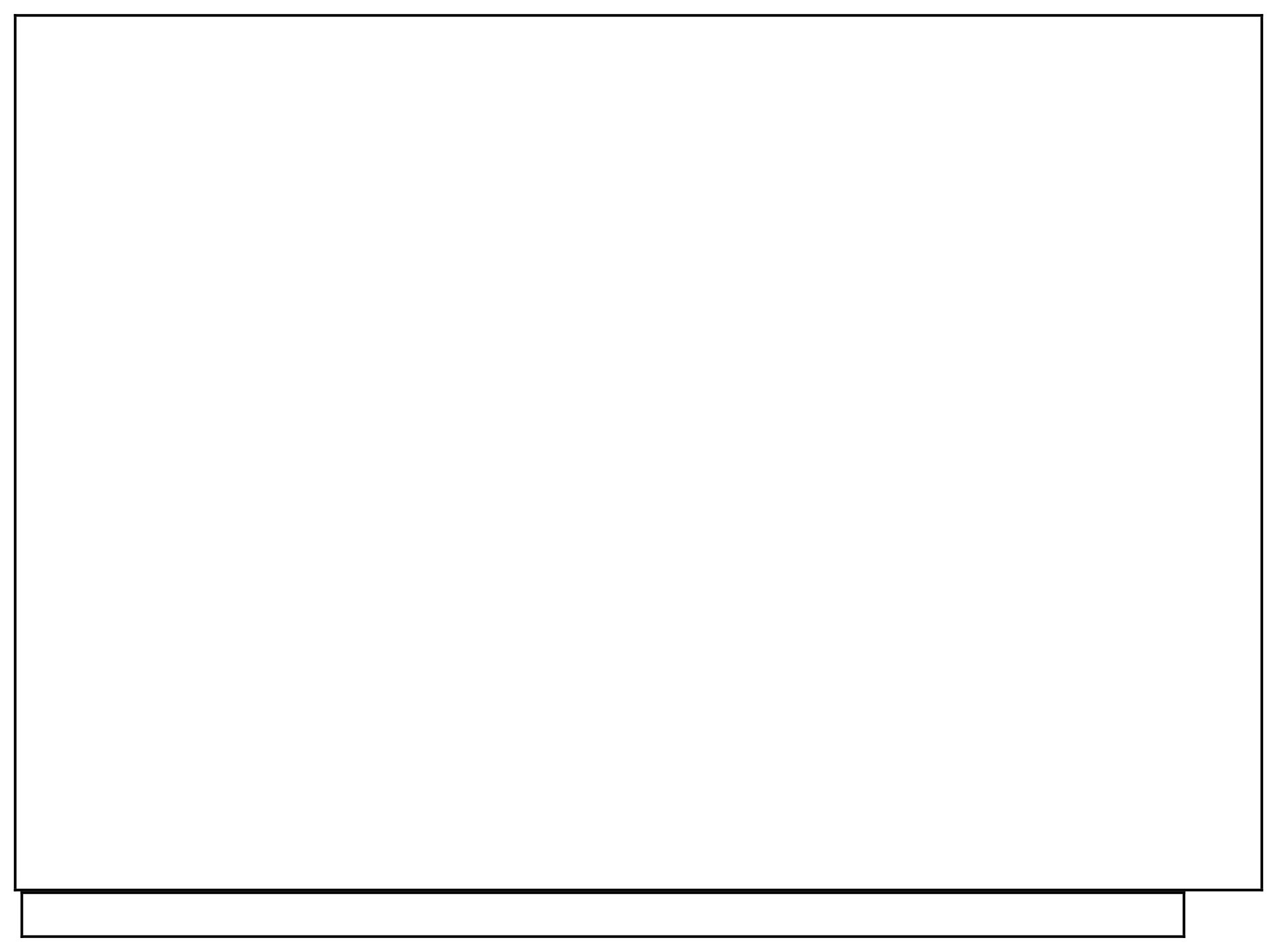
**OUTPUT :**

[**Lab-12-Introduction to I/O, I/O Operations, Object** **Serialization**](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56)

**1.**



**SOLUTION :**



**import java.util.Scanner;**

**public class DecodeString { public static**

**void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**String encodedString = scanner.nextLine();**

**StringBuilder decodedString = new StringBuilder();**

**int count = 0;**

**for (int i = 0; i < encodedString.length(); i++) {**

**if (encodedString.charAt(i) == '0') {**

**count++;**

**}**

**else { char decodedChar = (char) ('Z' - count**

**+**

**1); decodedString.append(decodedChar);**

**count = 0;**

**}**

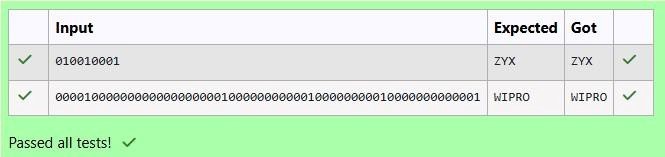
**}**

**System.out.println(decodedString.toString());**

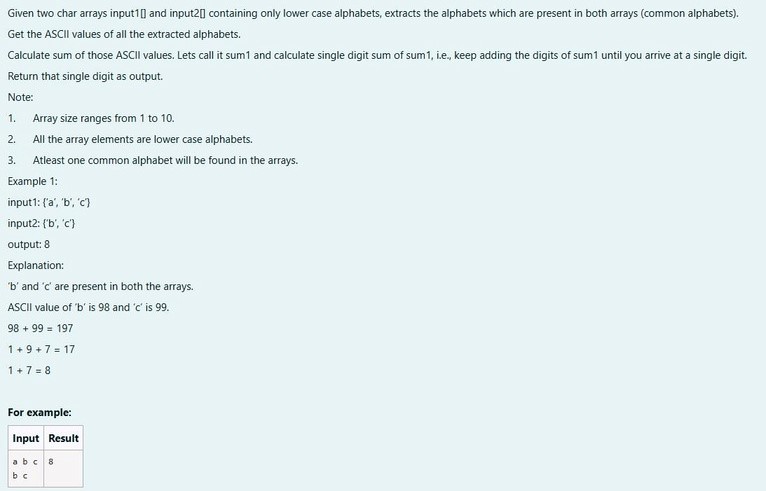
**}**

**}**

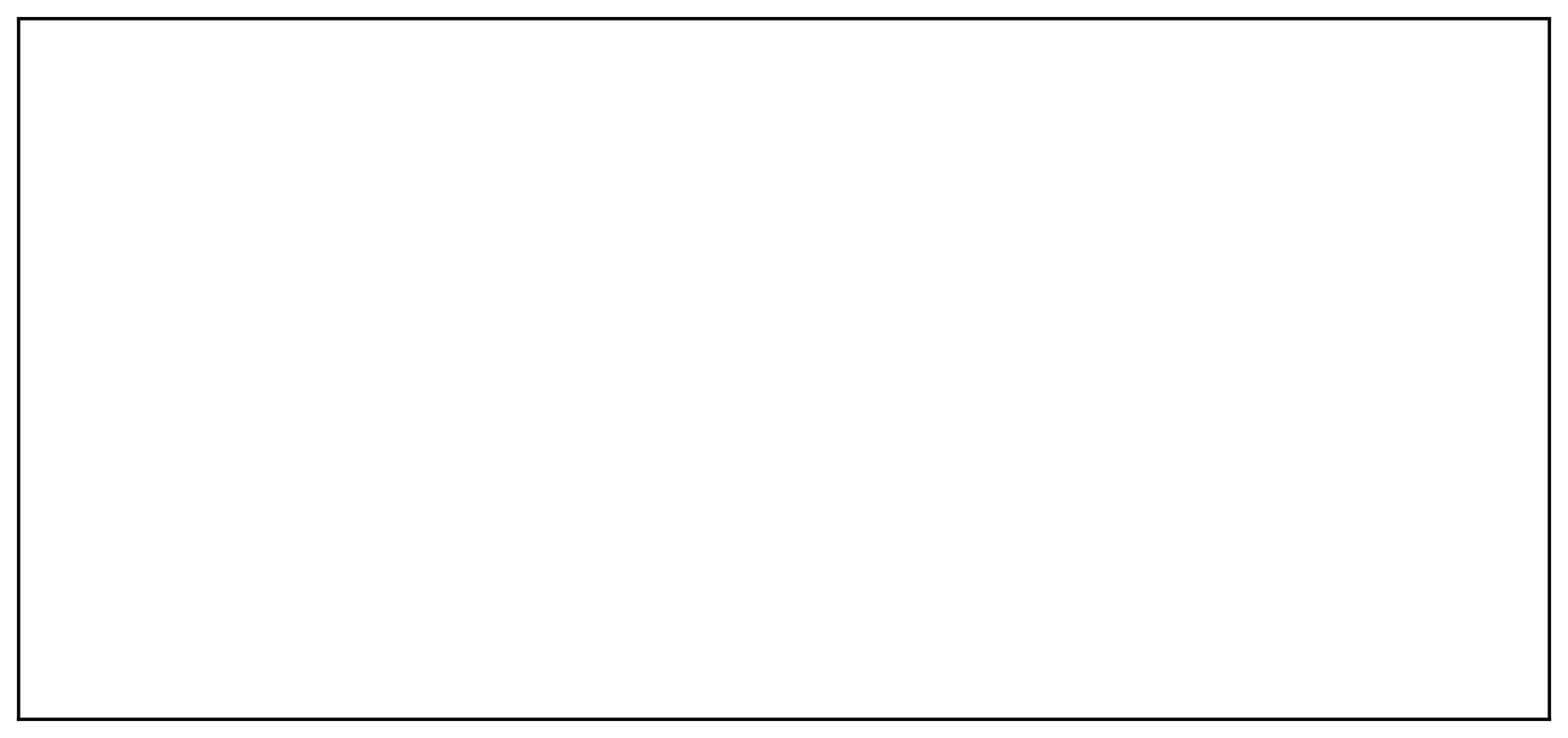
**OUTPUT :**



**2.**



**SOLUTION :**



**import**

**java.util.HashSet;**

**import**

**java.util.Set;**

**public**

**class**

**CommonAlphabetSum {**

**public static int singleDigitSum(int num) {**

**int sum = 0;**

**while (num > 0) {**

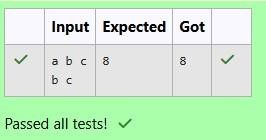
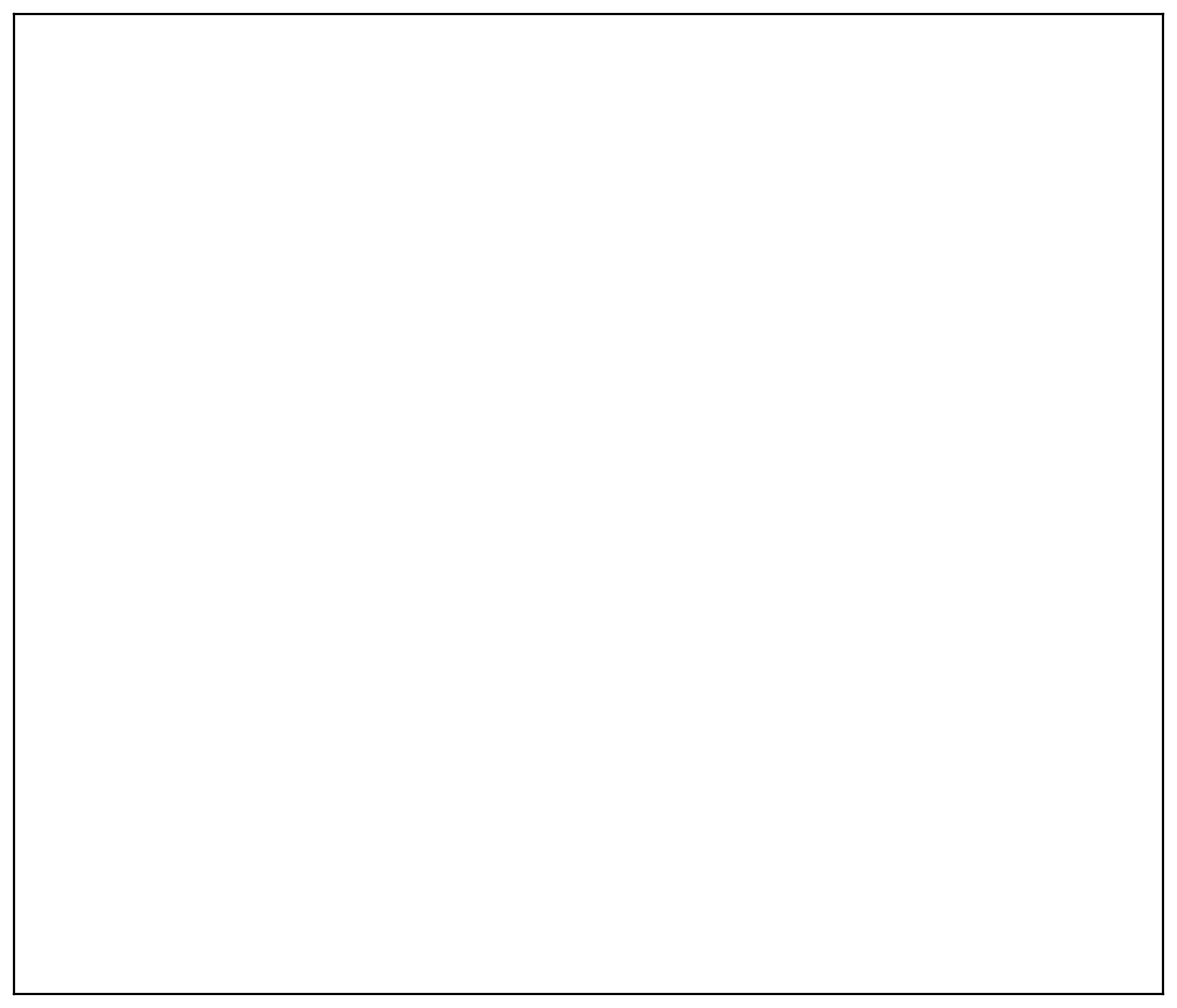
**sum += num % 10;**

**num /= 10;**

**}**

**if (sum > 9) { return**

**singleDigitSum(sum); }**



**OUTPUT :**

**return sum;**

**}**

**public static int calculateCommonAlphabetSum(char[] input1, char[] input2) {**

**Set<Character> set1 = new HashSet<>(); for (char c : input1) { set1.add(c);**

**}**

**int sum = 0; for**

**(**

**char c : input2)**

**{**

**if (set1.contains(c)) {**

**sum += c;**

**}**

**}**

**return singleDigitSum(sum);**

**}**

**public static void main(String[] args)**

**{**

**char[] input1 = {'a', 'b', 'c'};**

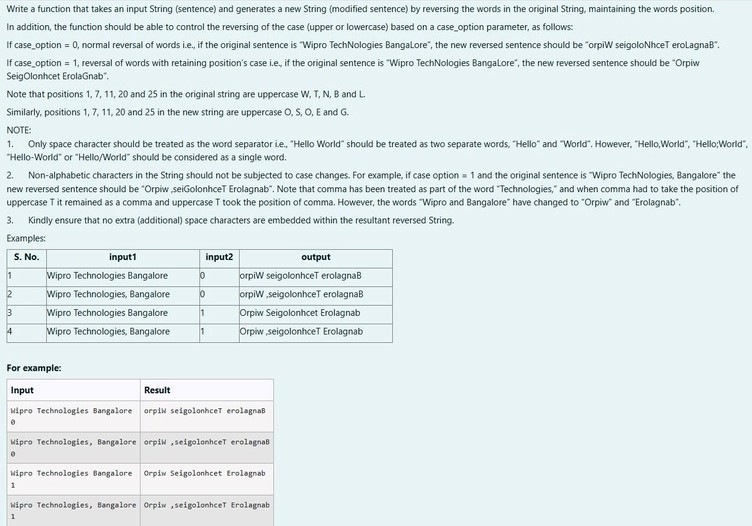
**char[] input2 = {'b', 'c', 'd'};**

**int result = calculateCommonAlphabetSum(input1, input2);**

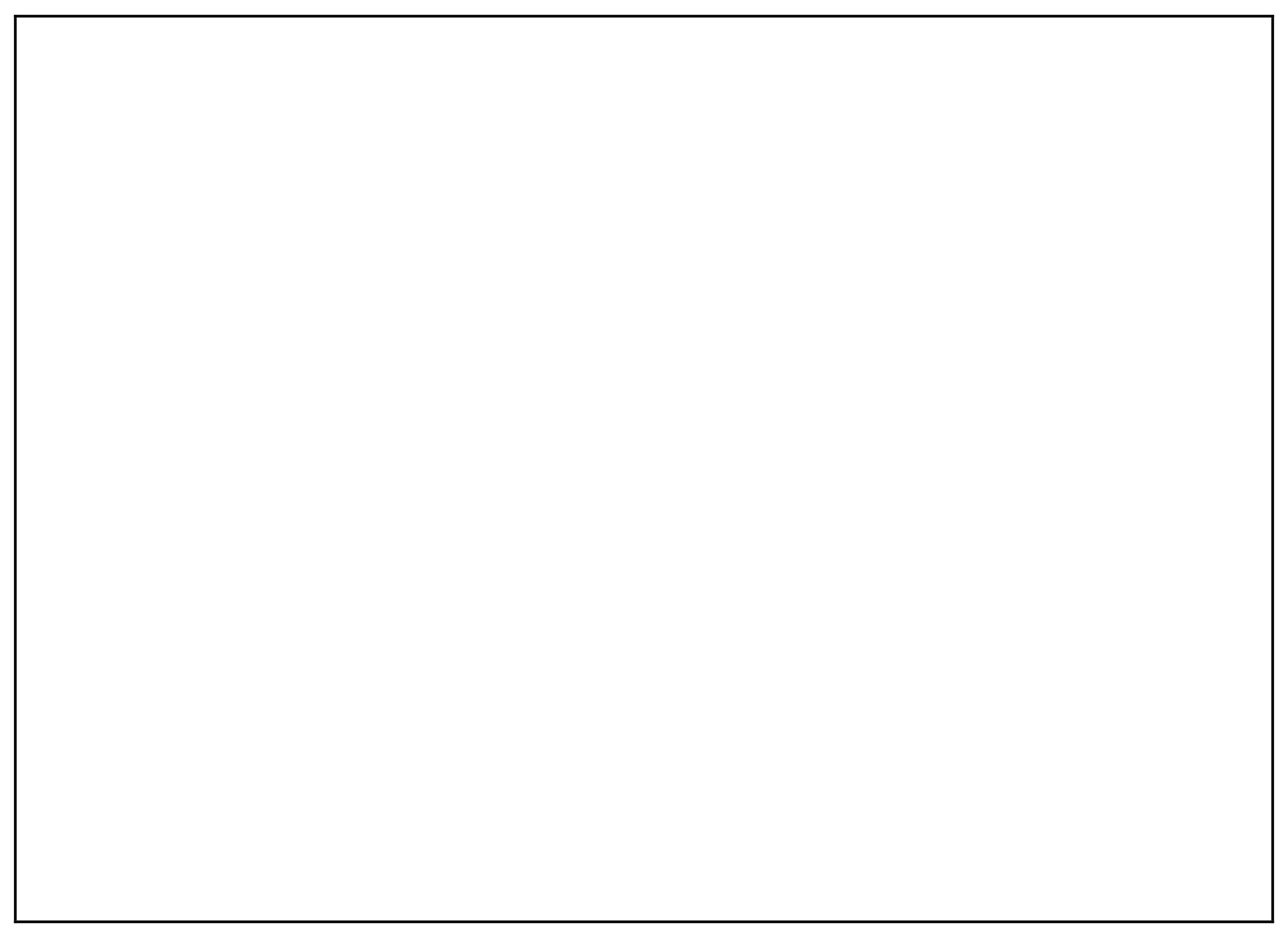
**System.out.println(result); }**

**}**

**3.**



**SOLUTION :**



**import java.util.Scanner;**

**public class WordReverser {**

**public static String reverseWordsWithCase(String sentence, int caseOption) {**

**// Split the sentence into words based on spaces**

**String[] words = sentence.split(" ");**

**// StringBuilder to store the result**

**StringBuilder result = new StringBuilder();**

**// Process each word for**

**(**

**String word : words)**

**{**

**// Reverse the word**

**String reversedWord = new StringBuilder(word).reverse().toString();**

**if (caseOption == 0) {**

**// If caseOption is 0, no case conversion, just reverse the word**

**result.append(reversedWord).append(" ");**

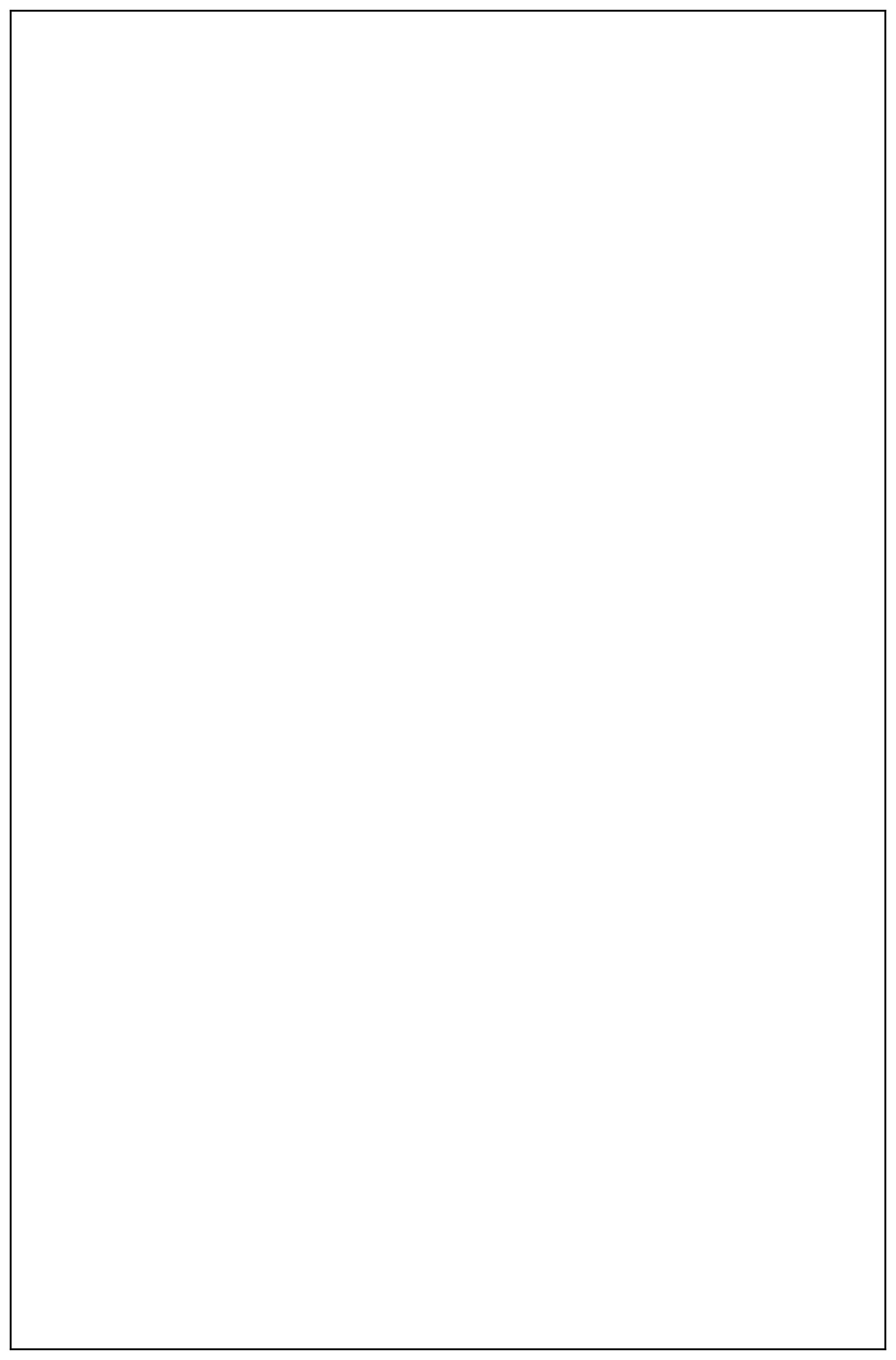
**}**

**else if (caseOption == 1)**

**{**

**// If caseOption is 1, adjust the case while maintaining original letter**

**positions**



**result.append(applyCaseConversion(reversedWord, word)).append(" ");**

**}**

**}**

**// Remove the trailing space and return the result return**

**result.toString().trim();**

**}**

**private static String applyCaseConversion(String reversedWord, String**

**originalWord) {**

**// StringBuilder to store the adjusted word**

**StringBuilder adjustedWord = new StringBuilder();**

**// Iterate over each character in the reversed word**

**for (int i = 0; i < reversedWord.length(); i++) { char**

**reversedChar = reversedWord.charAt(i); char**

**originalChar = originalWord.charAt(i);**

**if (Character.isLowerCase(originalChar)) {**

**// If the original character was lowercase, the reversed character should be**

**uppercase adjustedWord.append(Character.toLowerCase(reversedChar));**

**}**

**else if (Character.isUpperCase(originalChar))**

**{**

**// If the original character was uppercase, the reversed character should be**

**lowercase adjustedWord.append(Character.toUpperCase(reversedChar));**

**}**

**else**

**{**

**// Non-alphabetic characters remain unchanged**

**adjustedWord.append(reversedChar); }**

**}**

**return adjustedWord.toString();**

**}**

**public static void main(String[] args) {**

**// Create a Scanner object to get input from the user Scanner**

**scanner = new Scanner(System.in);**

**// Get sentence input from the user**

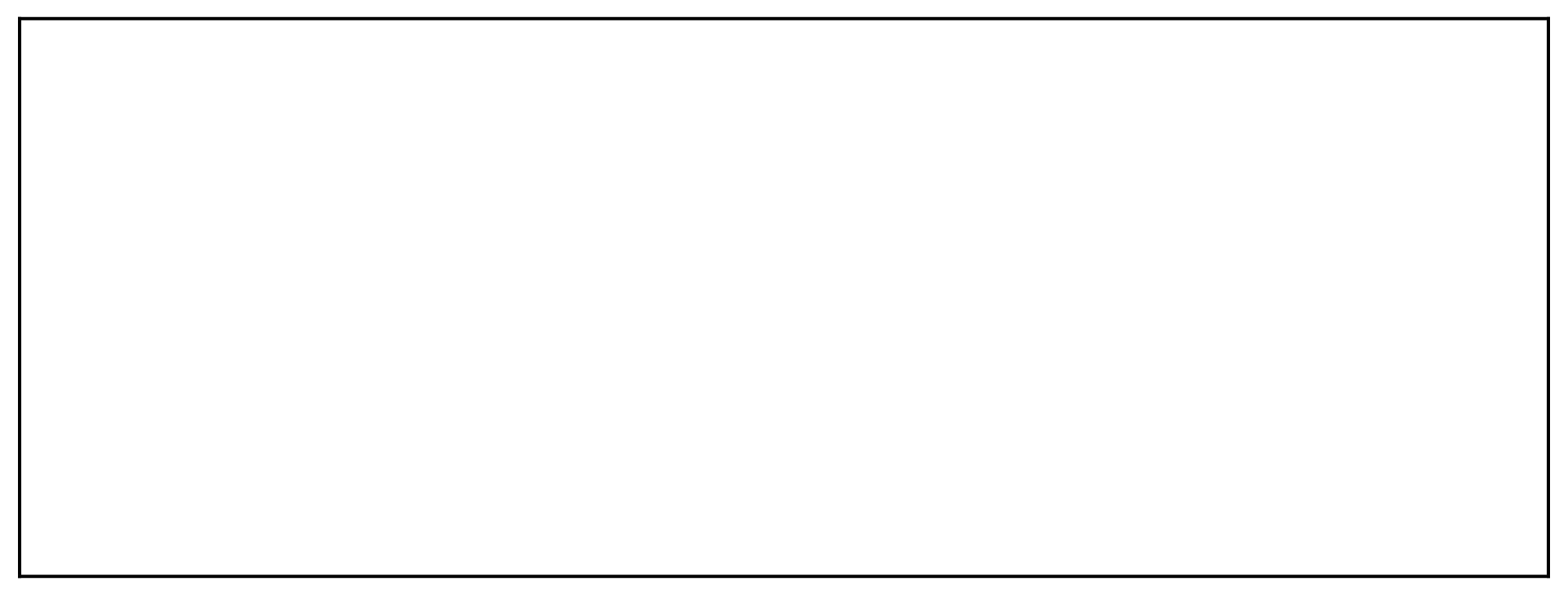
**String sentence = scanner.nextLine(); //**

**Get case option input from the user int**

**caseOption = scanner.nextInt();**

**// Validate the case option**

**if (caseOption != 0 && caseOption != 1) {**



**System.out.println("Invalid case option. Please enter 0 or 1.");**

**}**

**else**

**{**

**// Call the function and print the result**

**String result = reverseWordsWithCase(sentence, caseOption);**

**System.out.println(result);**

**}**

**// Close the scanner**

**scanner.close();**

**}**

**}**

**OUTPUT :**