

# RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI**  
**ENGINEERING COLLEGE**

**CS23333**

**OBJECT ORIENTED PROGRAMMING USING JAVA**

**Laboratory Observation Note Book**

Name : ... **CHANDRU K** .....

Year / Branch / Section : . **II.Year/. AIML-/A** .....

Register No. : ..... **231501031** .....

Semester : ..... **3RD.SEMESTER** .....

Academic Year : **2024-2025** .....

**Reg. No: 231501031**

**Year: 2nd Year**

**Name: CHANDRU K**

**Branch: AIML Sec: A**

<b>WEEK NO</b>	<b>DATE</b>	<b>TITLE</b>	<b>SIGN</b>
<b>01</b>	<b>21-09-24</b>	<b>Java Architecture, Language Basics</b>	
<b>02</b>	<b>21-09-24</b>	<b>Flow Control Statements</b>	
<b>03</b>	<b>22-09-24</b>	<b>Arrays</b>	
<b>04</b>	<b>22-09-24</b>	<b>Classes and Objects</b>	
<b>05</b>	<b>05-10-24</b>	<b>Inheritance</b>	
<b>06</b>	<b>05-10-24</b>	<b>String, String Buffer</b>	
<b>07</b>	<b>09-10-24</b>	<b>Interfaces</b>	
<b>08</b>	<b>13-10-24</b>	<b>Polymorphism, Abstract Classes, final Keyword</b>	
<b>09</b>	<b>13-10-24</b>	<b>Exception Handling</b>	
<b>10</b>	<b>31-10-24</b>	<b>Collection- List</b>	
<b>11</b>	<b>09-11-24</b>	<b>Set, Map</b>	

12	11-11-24	Introduction to 1/0, 1/0 Operations, Object Serialization	
----	----------	--	--

## Java Architecture, Language Basics

### PROGRAM 1.1:

AIM: Write a program to find whether the given input number is Odd.

If the given number is odd, the program should return 2 else It should return 1.

Note: The number passed to the program can either be negative. positive or zero. Zero should NOT be treated as Odd.

CODE:

```
import java.util.*; public class odd{

public static void main(String[] args){

Scanner in =new Scanner (System.in);

    int n = in .nextInt();

    if(n%2!=0){

        System.out.println("2");

    }

    else{

        System.out.println("1");

    }

}

}
```

OUTPUT :

	Input	Expected	Got	
✓	123	2	2	✓
✓	456	1	1	✓

Passed all tests! ✓

## PROGRAM 1.2:

AIM: Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.

The last digit should be returned as a positive number. For example,

if the given number is 197, the last digit is 7 if

the given number is -197, the last digit is 7

CODE:

```
import java.util.*; import
java.lang.*; public class
last{
    public static void main(String[] args){
Scanner in = new Scanner (System.in);      int
n= in .nextInt();
        System.out.println(Math.abs(n)%10);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✓	197	7	7	✓
✓	-197	7	7	✓

Passed all tests! ✓

## PROGRAM 1.3

AIM:

Rohit wants to add the last digits of two given numbers.

For example,

If the given numbers are 267 and 154, the output should be 11.

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

Write a program to help Rohit achieve this for any given two numbers.

Note: The sign of the input numbers should be ignored. i.e.

if the input numbers are 267 and 154, the sum of last two digits should be 11 if

the input numbers are 267 and -154, the sum of last two digits should be 11 if the

input numbers are -267 and 154, the sum of last two digits should be 11 if the

input numbers are -267 and -154, the sum of last two digits should be 11

CODE:

```
import java.util.*; import
java.lang.*; public class
last{
    public static void main(String[] args){
Scanner in =new Scanner(System.in);
int n1= in.nextInt();    int n2=
in.nextInt();
        System.out.println((Math.abs(n1)%10)+(Math.abs(n2)%10));
    }
}
```

OUTPUT:

	Input	Expected	Got	
✓	267 154	11	11	✓
✓	267 -154	11	11	✓
✓	-267 154	11	11	✓
✓	-267 -154	11	11	✓

Passed all tests! ✓

# Flow Control Statements

## PROGRAM 2.1

AIM:

Write a Java program to input a number from user and print it into words using for loop.  
How to display number in words using loop in Java programming.

Logic to print number in words in Java programming.

### Example

#### Input

1234

#### Output

One Two Three Four Input:

16 Output: one

six

CODE:

```
import java.util.*; public
class number{
    public static void main(String[] args){
        Scanner in =new Scanner(System.in);
String n=in.next();    for(int
i=0;i<n.length();i++){
switch(n.charAt(i)){        case '1':
            System.out.print("One ");
break;        case '2':
            System.out.print("Two ");
break;        case '3':
            System.out.print("Three ");
break;
        case '4':
            System.out.print("Four ");
break;        case '5':
```



```

        System.out.print("Five ");
break;        case '6':
        System.out.print("Six ");
break;        case '7':
        System.out.print("Seven ");
break;        case '8':
        System.out.print("Eight ");
break;        case '9':
        System.out.print("Nine ");
break;        case '0':
        System.out.print("Zero ");
break;
    }

}

}
}

```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	45	Four Five	Four Five	✓
✓	2	13	One Three	One Three	✓
✓	3	87	Eight Seven	Eight Seven	✓
Passed all tests! ✓					

PROGRAM 2.2:

Consider a sequence of the form 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149...

Write a method program which takes as parameter an integer n and prints the nth term of the above sequence. The nth term will fit in an integer value.

Example Input:

5

Output:

4

Example Input:

CHANDRU K 231501031

8

Output:

24

Example Input:

11

Output:

149

PROGRAM:

```
import java.util.*;
public class triple{
    public static void main(String[] args){
Scanner in = new Scanner(System.in);
int n= in.nextInt();    int
a=0,b=1,c=1,d=0;    if (n==0){
    System.out.println(a);
    }
    if (n==1){
    System.out.println(b);
    }
    if (n==2){
    System.out.println(c);
    }
    for (int i=3;i<n;i++){
d=a+b+c;    a=b;
b=c;    c=d;
    }
    System.out.println(d);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✓	5	4	4	✓
✓	8	24	24	✓
✓	11	149	149	✓

Passed all tests! ✓

PROGRAM 2.3:

Write a program that takes as parameter an integer n.

You have to print the number of zeros at the end of the factorial of n.

For example,  $3! = 6$ . The number of zeros are 0.  $5! = 120$ . The number of zeros at the end are 1. Note:  $n! < 10^5$  Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

253

CODE:

```
import java.util.Scanner;

class prog {    static int
findTrailingZeros(int n) {        if (n
< 0)            return -1;

        int count = 0;
        for (int i = 5; n / i >= 1; i *= 5)
count += n / i;

        return count;
    }

    public static void main(String[] args) {
        int n;
        Scanner sc = new Scanner(System.in);
n = sc.nextInt();
        System.out.println(findTrailingZeros(n));
    }
}
```

OUTPUT:

	Input	Expected	Got	
✓	3	0	0	✓
✓	60	14	14	✓
✓	100	24	24	✓
✓	1024	253	253	✓

Passed all tests! ✓

## Arrays

PROGRAM 3.1:

Given an array of numbers, you are expected to return the sum of the longest sequence of POSITIVE numbers in the array.

CHANDRU K 231501031

If there are NO positive numbers in the array, you are expected to return -1.

In this question's scope, the number 0 should be considered as positive.

Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all those POSITIVE numbers (see example 3 below). input1 represents the number of elements in the array. input2 represents the array of integers.

Example 1: input1 = 16 input2 = {-12, -16, 12, 18, 18, 14, -4, -12, -13,

32, 34, -5, 66, 78, 78, -79}

Expected output = 62

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers =  $12 + 18 + 18 + 14 = 63$ .

Example 2: input1 = 11 input2 = {-22, -24, 16, -1, -17, -

19, -37, -25, -19, -93, -61}

Expected output = -1

Explanation:

There are NO positive numbers in the input array. Therefore, the expected output for such cases = -1.

Example 3: input1

= 16 input2 = {-58,

32, 26, 92, -10, -4,

12, 0, 12, -2, 4, 32,

-9, -7, 78, -79}

Expected output = 174

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "32, 26, 92", "12, 0, 12", "4, 32", and "78". The first and second sequences "32, 26, 92" and "12, 0, 12" are the longest of the

four as they contain 4 elements each. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers =  $(32 + 26 + 92) + (12 + 0 + 12) = 174$ .

CODE:

```
import java.util.*; public class sum{

public static void main(String[] args){

Scanner in = new Scanner(System.in);

int n= in.nextInt();    int [] a = new int

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

a[i]=in.nextInt();

    }

    int cur_pos=0;    int

cur_tot=0;    int max=0;

int f=0;    int tot=0;

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

if(a[i]>=0){

cur_pos++;

cur_tot=cur_tot+a[i];

f=1;    }    else{

f=0;    }    if(f==0){

if (cur_pos>max){

max=cur_pos;

tot+=cur_tot;

    }

    else if(cur_pos==max){

tot+=cur_tot;

    }

}
```

```

        cur_pos=0;
cur_tot=0;
    }
    }    if
(tot>0){
        System.out.println(tot);
    }
else{
        System.out.println("-1");
    }
}
}

```

OUTPUT:

	Input	Expected	Got	
✓	16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62	62	✓
✓	11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1	-1	✓
✓	16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174	174	✓

Passed all tests! ✓

### PROGRAM 3.2:

Given an integer array as input, perform the following operations on the array, in the below specified sequence.

1. Find the maximum number in the array.
2. Subtract the maximum number from each element of the array.
3. Multiply the maximum number (found in step 1) to each element of the resultant array.

After the operations are done, return the resultant array.

Example 1: input1 = 4 (represents the number of elements in the input1 array) input2 = {1, 5, 6, 9}

Expected Output = {-72, -36, 27, 0} Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

$\{(1 - 9), (5 - 9), (6 - 9), (9 - 9)\} = \{-8, -4, -3, 0\}$

Step 3: Multiplying the maximum number 9 to each of the resultant array:

$\{(-8 \times 9), (-4 \times 9), (3 \times 9), (0 \times 9)\} = \{-72, -36, -27, 0\}$

So, the expected output is the resultant array {-72, -36, -27, 0}.

Example 2: input1 = 5 (represents the number of elements in the input1 array) input2 = {10, 87, 63, 42, 2}

Expected Output = {-6699, 0, -2088, -3915, -7395}

Explanation:

Step 1: The maximum number in the given array is 87.

Step 2: Subtracting the maximum number 87 from each element of the array:

$\{(10 - 87), (87 - 87), (63 - 87), (42 - 87), (2 - 87)\} = \{-77, 0, -24, -45, -85\}$

Step 3: Multiplying the maximum number 87 to each of the resultant array:

$\{(-77 \times 87), (0 \times 87), (-24 \times 87), (-45 \times 87), (-85 \times 87)\} = \{-6699, 0, -2088, -3915, -7395\}$  So, the expected output is the resultant array {-6699, 0, -2088, -3915, -7395}.

Example 3: input1 = 2 (represents the number of elements in the input1 array) input2 = {-9, 9}

Expected Output = {-162, 0} Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

$\{(-9 - 9), (9 - 9)\} = \{-18, 0\}$

Step 3: Multiplying the maximum number 9 to each of the resultant array:

$\{(-18 \times 9), (0 \times 9)\} = \{-162, 0\}$

So, the expected output is the resultant array {-162, 0}.

Note: The input array will contain not more than 100 elements

CODE:

```
import java.util.*;
```

```
public class sum{
```

```
CHANDRU K 231501031
```



```

    public static void main(String[] args){
Scanner in = new Scanner(System.in);
int n= in .nextInt();    int [] a = new
int[n];    for(int i=0;i<n;i++){
a[i]=in.nextInt();
    }
    int max = a[0];
for(int i=0;i<n;i++){
if (max<a[i]){
max=a[i];
    }
    }
    for(int i=0;i<n;i++){
a[i]=a[i]-max;
a[i]=a[i]*max;
        System.out.print(a[i]+" ");
    }
    }
}

```

OUTPUT:

	Input	Expected	Got	
✓	5 1 51 436 7860 41236	107	107	✓
✓	5 1 5 423 310 61540	53	53	✓

Passed all tests! ✓

PROGRAM 3.3:

AIM:

You are provided with a set of numbers (array of numbers).

You have to generate the sum of specific numbers based on its position in the array set provided to you.

This is explained below:

CHANDRU K 231501031

Example 1:

Let us assume the encoded set of numbers given to you is:

input1:5 and input2: {1, 51, 436, 7860, 41236} Step 1:

Starting from the 0<sup>th</sup> index of the array pick up digits as per below:

0<sup>th</sup> index – pick up the units value of the number (in this case is 1).

1<sup>st</sup> index - pick up the tens value of the number (in this case it is 5).

2<sup>nd</sup> index - pick up the hundreds value of the number (in this case it is 4).

3<sup>rd</sup> index - pick up the thousands value of the number (in this case it is 7).

4<sup>th</sup> index - pick up the ten thousands value of the number (in this case it is 4).

(Continue this for all the elements of the input array).

The array generated from Step 1 will then be – {1, 5, 4, 7, 4}.

Step 2:

Square each number present in the array generated in Step 1.

{1, 25, 16, 49, 16} Step

3:

Calculate the sum of all elements of the array generated in Step 2 to get the final result.

The result will be = 107.

Note:

1) While picking up a number in Step1, if you observe that the number is smaller than the required position then use 0.

2) In the given function, input1[] is the array of numbers and input2 represents the number of elements in input1.

Example 2: input1: 5 and input1: {1, 5, 423,

310, 61540} Step 1:

Generating the new array based on position, we get the below array:

{1, 0, 4, 0, 6}

In this case, the value in input1 at index 1 and 3 is less than the value required to be picked up based on position, so we use a 0.

Step 2: {1, 0,

16, 0, 36} Step

3:

The final result = 53.

CODE:

CHANDRU K 231501031

```

import java.util.*; public
class sum{
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        int n=in.nextInt();
        int [] a = new int[5];
        for(int i=0;i<n;i++){
            a[i]=in.nextInt();
        }
        int j=1,sum=0;
        for(int i=0;i<n;i++){
            a[i]=(a[i]/j)%10;
            sum=sum+(a[i]*a[i]);
            j=j*10;
        }
        System.out.print(sum);
    }
}

```

OUTPUT:

	Input	Expected	Got	
✓	5 1 51 436 7860 41236	107	107	✓
✓	5 1 5 423 310 61540	53	53	✓

Passed all tests! ✓

## Classes and Objects

PROGRAM 4.1:

AIM:

CHANDRU K 231501031

Create a Class Mobile with the attributes listed below, private

String manufacturer;

private String operating\_system;

public

String color; private int cost;

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer

is void setManufacturer(String manufacturer){

this.manufacturer= manufacturer;

}

CODE:

```
import java.util.*;
```

```
public class student{
```

```
String Name=null;
```

```
int Rollno=0;    public
```

```
student(){
```

```
    System.out.println("No-arg constructor is invoked");
```

```
}
```

```
public student(String name){
```

```
    System.out.println("1 arg constructor is invoked");
```

```
    Name=name;
```

```
}
```

```
public student(String name,int rollno){
```

```
    System.out.println("2 arg constructor is invoked");
```

```
    Name=name;
```

```
    Rollno=rollno;
```

```
}
```

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

```
student obj1=new student();    student
```

```
obj2=new student("Rajalakshmi");    student
```

```
obj3=new student("Lakshmi",101);
```

CHANDRU K 231501031

```

        System.out.println("Name =" + obj1.Name + " , Roll no = " + obj1.Rollno);
        System.out.println("Name =" + obj2.Name + " , Roll no = " + obj2.Rollno);
        System.out.println("Name =" + obj3.Name + " , Roll no = " + obj3.Rollno);
    }
}

```

OUTPUT:

	Test	Expected	Got	
✓	1	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	✓

Passed all tests! ✓

#### PROGRAM 4.2:

AIM: Create a class called "Circle" with a radius attribute. You can access and modify this attribute using getter and setter methods. Calculate the area and circumference of the circle.

CODE:

```

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()));

}

}

```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	4	Area = 50.27 Circumference = 25.13	Area = 50.27 Circumference = 25.13	✓
✓	2	6	Area = 113.10 Circumference = 37.70	Area = 113.10 Circumference = 37.70	✓
✓	3	2	Area = 12.57 Circumference = 12.57	Area = 12.57 Circumference = 12.57	✓

Passed all tests! ✓

PROGRAM 4.3:

AIM: Create a class Student with two private attributes, name and roll number. Create three objects by invoking different constructors available in the class Student.

Student()

Student(String name)

Student(String name, int rollno)

CODE:

```

import java.util.*;

public class student{

```

CHANDRU K 231501031

```

String Name=null;
int Rollno=0;    public
student(){
    System.out.println("No-arg constructor is invoked");
}
public student(String name){
    System.out.println("1 arg constructor is invoked");
    Name=name;
}
public student(String name,int rollno){
    System.out.println("2 arg constructor is invoked");
    Name=name;
    Rollno=rollno;
}
public static void main(String[] args){
student obj1=new student();    student
obj2=new student("Rajalakshmi");    student
obj3=new student("Lakshmi",101);
    System.out.println("Name =" +obj1.Name+" , Roll no = "+obj1.Rollno);
    System.out.println("Name =" +obj2.Name+" , Roll no = "+obj2.Rollno);
System.out.println("Name =" +obj3.Name+" , Roll no = "+obj3.Rollno);
}
}

```

OUTPUT:



	Test	Expected	Got	
✓	1	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	✓
Passed all tests! ✓				

## Inheritance

### PROGRAM 5.1:

AIM: Create a class known as "BankAccount" with methods called deposit() and withdraw().

Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

CODE:

```
class BankAccount {

    // Private field to store the account number    private

    String accountNumber;
```

CHANDRU K 231501031

```

// Private field to store the balance    private
double balance;

// Constructor to initialize account number and balance
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
this.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

this.balance;

}

}

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);
    }
}
}

class prog {

    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:

Expected	Got
✓ Create a Bank Account object (A/c No. BA1234) with initial balance of \$500: Deposit \$1000 into account BA1234: New balance after depositing \$1000: \$1500.0 Withdraw \$600 from account BA1234: New balance after withdrawing \$600: \$900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of \$300: Try to withdraw \$250 from SA1000! Minimum balance of \$100 required! Balance after trying to withdraw \$250: \$300.0	✓ Create a Bank Account object (A/c No. BA1234) with initial balance of \$500: Deposit \$1000 into account BA1234: New balance after depositing \$1000: \$1500.0 Withdraw \$600 from account BA1234: New balance after withdrawing \$600: \$900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of \$300: Try to withdraw \$250 from SA1000! Minimum balance of \$100 required! Balance after trying to withdraw \$250: \$300.0

Passed all tests! ✓

## PROGRAM 5.2:

**AIM:** create a class called College with attribute String name, constructor to initialize the name attributes, a method called Admitted (). Create a subclass called CSE that extends

CHANDRU K 231501031

Student class, with department attribute, Course () method to sub class. Print the details of the Student.

CODE:

```
class College
{
    protected 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 depart) {
        // initialize the instance variables
        super(collegeName); this.studentName=studentName;
        this.department=depart;
    }

    public String toString(){
        // return the details of the student return "CollegeName :
        "+collegeName + "\nStudentName : "+studentName
        +"\nDepartment : "+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:

	Expected	Got	
✓	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	✓
Passed all tests! ✓			

#### PROGRAM 5.3:

AIM: Create a class Mobile with constructor and a method basicMobile().

Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().

Create a subclass AndroidMobile which extends CameraMobile, with constructor and a method androidMobile().

display the details of the Android Mobile class by creating the instance.

#### CODE:

```
class Mobile{
public Mobile(){
    System.out.println("Basic Mobile is Manufactured");
}
    public void basicMobile(){
}
}
class CameraMobile extends Mobile{
public CameraMobile(){
    System.out.println("Camera Mobile is Manufactured");
}
```

```

    public void newfeature(){
        System.out.println("Camera Mobile with 5MG px");
    }
}
class AndroidMobile extends CameraMobile{
    public AndroidMobile(){
        System.out.println("Android Mobile is Manufactured");
    }
    public void androidMobile(){
        System.out.println("Touch Screen Mobile is Manufactured");
    }
}
public class main{
    public static void main(String[] args){
        AndroidMobile s=new AndroidMobile();
        s.newfeature();
        s.androidMobile();
    }
}

```

OUTPUT:

	Expected	Got	
✓	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	✓

Passed all tests! ✓



## String, String Buffer

PROGRAM 6.1:

AIM: Given 2 strings input1 & input2.

- Concatenate both the strings.
- Remove duplicate alphabets & white spaces.
- Arrange the alphabets in descending order.

CODE:

```
import java.util.*;

public class a {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String s1 = in.nextLine().trim();
        String s2 = in.nextLine().trim();
        if(s1.isEmpty() && s2.isEmpty()){
            System.out.print("null");
        }
        else{
            String s3 = s1 + s2;
```

```

        Set<Character> s4 = new HashSet<>();
for (char x : s3.toCharArray()) {
    s4.add(x);
}
        List<Character> sortedList = new ArrayList<>(s4);
Collections.sort(sortedList, Collections.reverseOrder());        for(char
a: sortedList){
            System.out.print(a);
        }
    }
}
}

```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	apple orange	rponlgea	rponlgea	✓
✓	2	fruits are good	utsroigfeda	utsroigfeda	✓
✓	3		null	null	✓

Passed all tests! ✓

PROGRAM 6.2:

AIM: Given a String input1, which contains many number of words separated by : and each word contains exactly two lower case alphabets, generate an output based upon the below 2 cases. Note:

All the characters in input 1 are lowercase alphabets.

Input 1 will always contain more than one word separated by :

Output should be returned in uppercase.

CODE:

```

import java.util.*; import
java.lang.*; public class
alpha{
    public static void main(String[] args) {
        Scanner in=new Scanner(System.in);
        String s=in.nextLine();
        String[] str=s.split(":");
        ArrayList<String> l1=new ArrayList<>();
        StringBuilder sb = new StringBuilder();
        String alpha=new String("abcdefghijklmnopqrstuvwxyz");
        for(String x:str){
            l1.add(x);
        }
        for(String a:l1){
            if(a.charAt(0)==a.charAt(1)){
                sb.append(a.charAt(0));
            }
            else{
                int q = a.charAt(0)-a.charAt(1);
                sb.append(alpha.charAt(q-1));
            }
        }
        System.out.print(sb.toString().toUpperCase());
    }
}

```

OUTPUT:

	Input	Expected	Got	
✓	ww:ii:pp:rr:oo	WIPRO	WIPRO	✓
✓	zx:za:ee	BYE	BYE	✓

Passed all tests! ✓

### PROGRAM 6.3:

You are provided a string of words and a 2-digit number. The two digits of the number represent the two words that are to be processed.

For example:

If the string is "Today is a Nice Day" and the 2-digit number is 41, then you are expected to process the 4th word ("Nice") and the 1st word ("Today").

### CODE:

```
import java.util.*;

public class a {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String s = in.nextLine();    int a =
        in.nextInt();

        String b = String.valueOf(a);
        char[] n_b = b.toCharArray();
        String[] str = s.split(" ");
        ArrayList<String> n_str = new ArrayList<>();

        for (char i : n_b) {
            int n = Character.getNumericValue(i);
            if (n - 1 < str.length) {
                n_str.add(str[n - 1]);
            }
        }

        for (String x : n_str) {
            int mid = x.length() / 2;

            if (x.length() % 2 == 0) {
```

```

        StringBuilder reversed = new StringBuilder(x.substring(0, mid)).reverse();
        String slice = reversed.toString() + x.substring(mid, x.length());
        System.out.print(slice + " ");
    } else {
        StringBuilder reversed = new StringBuilder(x.substring(0, mid + 1)).reverse();
        String slice = reversed.toString() + x.substring(mid, x.length());
        System.out.print(slice + " ");
    }
}
System.out.println();
}
}

```

OUTPUT:

	Input	Expected	Got	
✓	Today is a Nice Day 41	iNce doTday	iNce doTday	✓
✓	Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes	naMngo arGpes	✓

Passed all tests! ✓

# Interfaces

PROGRAM 7.1:

AIM: RBI issues all national banks to collect interest on all customer loans.

Create an RBI interface with a variable String parentBank="RBI" and abstract method rateOfInterest().

RBI interface has two more methods default and static method.

CODE:

```
interface rbi{
    String parentBank="RBI";
    abstract void rateofinterest();
    default void policynote(){
        System.out.println("RBI has a new Policy issued in 2023");
    }
    static void regulation(){
        System.out.println("RBI has updated new regulations in 2024.");
    }
}

class sbi implements rbi{
    public void rateofinterest(){
        System.out.println("SBI rate of interest: 7.6 per annum.");
    }
    public void def(){
        policynote();
    }
}

class karur implements rbi{
    public void rateofinterest(){
        System.out.println("Karur rate of interest: 7.4 per annum.");
    }
}

public class main{
    public static void
    main(String[] args){
        sbi s = new
        sbi();
    }
}
```

```

        karur r = new karur();
s.def();
rbi.regulation();
    s.rateofinterest();
    r.rateofinterest();
    }
}

```

OUTPUT:

	Test	Expected	Got	
✓	1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓
Passed all tests! ✓				

PROGRAM 7.2:

AIM:

Create interfaces shown below.

```

interface Sports { public void
setHomeTeam(String name); public void
setVisitingTeam(String name);
}

```

interface Football extends Sports { public void homeTeamScored(int points); public void visitingTeamScored(int points);} create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.

CODE:

```

public void setHomeTeam(String name); public

void setVisitingTeam(String name);

}

```

CHANDRU K 231501031

```

interface Football extends Sports {

public void homeTeamScored(int

points); public void

visitingTeamScored(int points);

}

class College implements Football {

    String homeTeam;

    String visitingTeam;


    public void setHomeTeam(String name){        this.homeTeam=name;

    }

    public void setVisitingTeam(String name){    visitingTeam=name;

    }

    public void homeTeamScored(int points){

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

    }

    public void visitingTeamScored(int points){

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

    }

    public void winningTeam(int p1, int p2){

    if(p1>p2)

        {

```



```

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

    else if(p1<p2)
    {
        System.out.println(visitingTeam+" is the winner!");
    }

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

}

}

```

```

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

String hname;

        Scanner sc= new
Scanner(System.in);

hname=sc.nextLine();    String
vteam=sc.nextLine();    int
htpoints=sc.nextInt();    int
vtpoints=sc.nextInt();    College s= new
College();

```

CHANDRU K 231501031

```

s.setHomeTeam(hname);

s.setVisitingTeam(vteam);

s.homeTeamScored(htpoints);

s.visitingTeamScored(vtpoints);

s.winningTeam(htpoints,vtpoints);

}

}

```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	✓
✓	2	Anna Balaji 21 21	Anna 21 scored Balaji 21 scored It's a tie match.	Anna 21 scored Balaji 21 scored It's a tie match.	✓
✓	3	SRM VIT 20 21	SRM 20 scored VIT 21 scored VIT is the winner!	SRM 20 scored VIT 21 scored VIT is the winner!	✓

Passed all tests! ✓

### PROGRAM 7.3:

AIM: create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.

CODE:

```

import java.util.*; interface

playable{    abstract void

play();

}

class football implements

playable{    String name;    public

```

```

football(String name){
    this.name=name;
}

    public void play(){
        System.out.println(name+" is Playing football");
    }
}

class volleyball implements
playable{    String name;    public
volleyball(String name){
    this.name=name;
}

    public void play(){
        System.out.println(name+" is Playing volleyball");
    }
}

class basketball implements
playable{    String name;    public
basketball(String name){
    this.name=name;
}

    public void play(){
        System.out.println(name+" is Playing basketball");
    }
}

```

```

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

    Scanner in = new Scanner(System.in);

    String a= in.next();

    String b= in.next();    String
c= in.next();    playable s1 = new
football(a);    playable s2 = new
volleyball(b);    playable s3 = new
basketball(c);

    s1.play();
s2.play();
s3.play();

    }
}

```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	✓
✓	2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	✓

Passed all tests! ✓

## Polymorphism, Abstract Classes, final Keyword

PROGRAM 8.1:

AIM: Given a Java Program that contains the bug in it, your task is to clear the bug to the output. You should delete any piece of code.

CODE:

```
public class FinalExample
```

CHANDRU K 231501031

```

{
    final int MAX_SPEED=120;

    public final void display()
    {
        System.out.println("The maximum speed is: "+ MAX_SPEED+ " km/h");
    }

    public static void main(String[] args)
    {
        SubExample obj=new SubExample();

        obj.display();
    }
}

obj.show();

}

}

class SubExample extends FinalExample
{
    public void show()
    {
        System.out.println("This is a subclass of FinalExample.");
    }
}

```

OUTPUT:

	Test	Expected	Got	
✓	1	The maximum speed is: 120 km/h This is a subclass of FinalExample.	The maximum speed is: 120 km/h This is a subclass of FinalExample.	✓
Passed all tests! ✓				

## PROGRAM 8.2:

### AIM:

Create a base class Shape with a method called calculateArea(). Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

### CODE:

```
import java.util.*; import java.lang.*;
abstract class shape{    public
abstract void calculateArea();
}
class circle extends shape{
double r;    public
circle(double r){
this.r=r;
}
public void calculateArea(){
double area=Math.PI*r*r;
System.out.println("Area of a circle: "+String.format("%.2f", area));
}
}
class rectangle extends shape{
double l,b;    public rectangle(double
l,double b) {
```

```

        this.l=l;
this.b=b;
    }
    public void calculateArea(){
double area=l*b;
        System.out.println("Area of a Rectangle: "+String.format("%.2f", area));

    }
}
class triangle extends shape{
double b,h;    public triangle(double
b,double h){        this.b=b;
this.h=h;
    }
    public void calculateArea(){
double area=0.5*b*h;
        System.out.println("Area of a Triangle: "+String.format("%.2f", area));
    }
}
public class main{
    public static void main(String[] args){
Scanner in = new Scanner(System.in);
double a=in.nextDouble();    double
b=in.nextDouble();    double
c=in.nextDouble();    double
d=in.nextDouble();    double
e=in.nextDouble();    circle obj1=new
circle(a);    rectangle obj2=new
rectangle(b,c);    triangle obj3=new
triangle(d,e);    obj1.calculateArea();
obj2.calculateArea(); obj3.calculateArea();
    }
}

```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	✓
✓	2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	✓

Passed all tests! ✓

### PROGRAM 8.3:

#### AIM:

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings, extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found

#### CODE:

```
import java.util.*; public
class main{
    public static void main(String[] args){
        Scanner in =new Scanner(System.in);
        int n= in.nextInt();
        String [] a =new String[n];
        for(int i=0;i<n;i++){
            a[i]=in.next().toLowerCase();
        }
    }
}
```



```

        ArrayList<String> l1 = new ArrayList<String>();
for(String i: a){
    char f=i.charAt(0);
char l =i.charAt(i.length()-1);
    if((f=='a'||f=='e'||f=='i'||f=='o'||f=='u')&&(l=='a'||l=='e'||l=='i'||l=='o'||l=='u')){
l1.add(i);
    }
}
if(l1.isEmpty()){
    System.out.print("no matches found");
}
else{
    for(String i : l1){
        System.out.print(i);
    }
}
}
}

```

OUTPUT:

	Input	Expected	Got	
✓	3 oreo sirish apple	oreoapple	oreoapple	✓
✓	2 Mango banana	no matches found	no matches found	✓
✓	3 Ate Ace Girl	ateace	ateace	✓

Passed all tests! ✓

## Exception Handling

PROGRAM 9.1:

AIM:

Write a Java program to handle ArithmeticException and ArrayIndexOutOfBoundsException.

CODE:

CHANDRU K 231501031

```

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(); sum=sum+name[i];
        }

        System.out.println(sum);

    }

    catch(InputMismatchException e)
    {
        System.out.println("You entered bad data.");
    }
}
}

```

OUTPUT:

Test	Input	Expected	Got
1	6 1 0 4 1 2 8	java.lang.ArithmeticException: / by zero I am always executed	java.lang.ArithmeticException: / by zero I am always executed
2	3 10 20 30	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed

Passed all tests! ✓

PROGRAM 9.2:

AIM:

Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.

CODE:

```
import java.util.*; class
prog {
    public static void main(String[] args) {
int n = 82;    trynumber(n);    n = 37;
        // call the trynumber(n);
        trynumber(n);

    }

    public static void trynumber(int n) {
try {
        //call the checkEvenNumber()
checkEvenNumber (n);
        System.out.println(n + " is even.");
    } catch (IllegalArgumentException e) {
        System.out.println("Error: " + e.getMessage());
    }
}
```

CHANDRU K 231501031

```

public static void checkEvenNumber(int number) {
if (number % 2 != 0) {
    throw new IllegalArgumentException(number + " is odd.");
}
}
}

```

OUTPUT:

	Expected	Got	
✓	82 is even. Error: 37 is odd.	82 is even. Error: 37 is odd.	✓

Passed all tests! ✓

### PROGRAM 9.3:

In the following program, an array of integer data is to be initialized.

During the initialization, if a user enters a value other than an integer, it will throw an `InputMismatchException` exception.

On the occurrence of such an exception, your program should print "You entered bad data." If there is no such exception it will print the total sum of the array.

### CODE:

```

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();
sum=sum+name[i];
        }
        System.out.println(sum);

    }
    catch(InputMismatchException e)
    {
        System.out.println("You entered bad data.");

    }

}

}
}

```

OUTPUT:

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓
✓	2 1 g	You entered bad data.	You entered bad data.	✓

Passed all tests! ✓

## Collection- List

### PROGRAM 10.1:

AIM: Given an ArrayList, the task is to get the first and last element of the ArrayList in Java.

Input: ArrayList = [1, 2, 3, 4]

Output: First = 1, Last =  
4

### CODE:

```
import java.util.*; public class main{  
    public static void main(String[] args){  
        Scanner in = new  
Scanner(System.in);  
        List<Integer> li=new ArrayList<>();  
        int n = in.nextInt();    for(int  
i=0;i<n;i++){        int k = in.nextInt();  
        li.add(k);  
    }  
        System.out.println("ArrayList: "+li);  
        System.out.println("First : "+li.get(0)+","+" Last : "+li.get(n-1));  
    }  
}
```

### OUTPUT:

	Test	Input	Expected	Got	
✓	1	6 30 20 40 50 10 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	✓
✓	2	4 5 15 25 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	✓

Passed all tests! ✓

## PROGRAM 10.2:

AIM: The given Java program is based on the ArrayList methods and its usage. The Java program is partially filled. Your task is to fill in the incomplete statements to get the desired output.

list.set();

list.indexOf();

list.lastIndexOf()

list.contains()

list.size()); list.add();

list.remove();

The above methods are used for the below Java program.

CODE:

```
import java.util.*; public class Prog {
public static void main(String[] args)
{
Scanner sc= new Scanner(System.in);
int n = sc.nextInt();

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

```

        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:



	Test	Input	Expected	Got	
✓	1	5 1 2 3 100 5	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3 false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3 false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	✓
Passed all tests! ✓					

### PROGRAM 10.3:

AIM:

Write a Java program to reverse elements in an array list.

CODE:

```
import java.util.*;
public class main{
    public static void main(String[] args){
        Scanner in = new Scanner (System.in);
        List<String> l1 = new ArrayList<>();
        List<String> l2 = new ArrayList<>();
        int n = in.nextInt();    for(int
        i=0;i<n;i++){           String str =
        in.next();              l1.add(str);
        }
        for(int i=n-1;i>=0;i--){
        l2.add(l1.get(i));
        }
        System.out.println("List before reversing :\n"+l1);
        System.out.println("List after reversing :\n"+l2);
    }
}
```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	5 Red Green Orange White Black	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	✓
✓	2	4 CSE AIML AIDS CYBER	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	✓

Passed all tests! ✓

## Set, Map

PROGRAM 11.1:

AIM:

IMPLEMENT A JAVA PROGRAM WITH HASH FEATURES.

CODE:

```
import java.util.HashSet; import
java.util.Scanner; class prog { public
static void main(String[] args) {
Scanner sc= new Scanner(System.in);
int n = sc.nextInt();

    // Create a HashSet object called numbers
    HashSet<Integer> numbers = new HashSet<Integer>();

    // Add values to the set
    for(int i=0;i<n;i++)
        numbers.add(sc.nextInt());

    int skey=sc.nextInt();

    // Show which numbers between 1 and 10 are in the set
    if(numbers.contains(skey)){
        System.out.println( skey+ " was found in the set.");
    } else {
        System.out.println(skey + " was not found in the set.");
    }
}
```

}OUTPUT:

CHANDRU K 231501031

Test	Input	Expected	Got	
✓	1 5 90 56 45 78 25 78	78 was found in the set.	78 was found in the set.	✓
✓	2 3 -1 2 4 5	5 was not found in the set.	5 was not found in the set.	✓

Passed all tests! ✓

## PROGRAM 11.2:

AIM: Write a Java program to compare two sets and retain elements that are the same.

### CODE:

```
import java.util.Scanner; import
java.util.HashSet; import
java.util.Set; import
java.util.Iterator; class prog{

    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);    int
n1=s.nextInt();

        HashSet<String> h1=new HashSet<>();

        for(int i=0;i<n1;i++)
        {
            h1.add(s.next());
        }

        int n2=s.nextInt();

        HashSet<String> h2=new HashSet<>();

        for(int i=0;i<n2;i++)
        {
```

```

        h2.add(s.next());

    }

    h1.retainAll(h2);

    Iterator iter=h1.iterator();

    while(iter.hasNext())

    {

        System.out.println(iter.next());

    }

}

```

OUTPUT:

	Test	Input	Expected	Got	
✓	1	5 Football Hockey Cricket Volleyball Basketball 7 Golf Cricket Badminton Football Hockey Volleyball Throwball	Cricket Hockey Volleyball Football	Cricket Hockey Volleyball Football	✓
✓	2	4 Toy Bus Car Auto 3 Car Bus Lorry	Bus Car	Bus Car	✓

Passed all tests! ✓

PROGRAM 11.3:

AIM : Java HashMap Methods

```

import java.util.HashMap;

import java.util.Map.Entry;

import java.util.Set; import
java.util.Scanner;

```

CHANDRU K 231501031

```

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);

    // Number of entries to add to the map
    int n = sc.nextInt();

    // Adding entries to 'map'
    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); // Corrected: pass 'map' as an argument

// 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'
Integer value = map.get("TWO"); // Use Integer to avoid NullPointerException
if (value != null) {
    System.out.println(value);
} else {
    System.out.println("Key 'TWO' not found.");
}

// Checking whether key 'ONE' exists in map
System.out.println( map.containsKey("ONE"));

```

```

// Checking whether value '3' exists in map
System.out.println(map.containsValue(3));

// Retrieving the number of key-value pairs present in map
System.out.println(map.size());

sc.close(); // Close scanner to prevent resource leak
}
}OUTPUT:

```

	Test	Input	Expected	Got	
✓	1	3	ONE : 1	ONE : 1	✓
		ONE	TWO : 2	TWO : 2	
		1	THREE : 3	THREE : 3	
		TWO	-----	-----	
		2	SIX : 6	SIX : 6	
		THREE	ONE : 1	ONE : 1	
		3	TWO : 2	TWO : 2	
			SEVEN : 7	SEVEN : 7	
			THREE : 3	THREE : 3	
		2	2	2	
		true	true	true	
		true	true	true	
		4	4	4	

Passed all tests! ✓

## Introduction to 1/0, 1/0 Operations, Object Serialization

PROGRAM 12.1:

AIM:

You are provided with a string which has a sequence of 1's and 0's.

This sequence is the encoded version of a English word. You are supposed write a program to decode the provided string and find the original word.

Each alphabet is represented by a sequence of 0s.

CODE:

```

import java.util.*; public class main{

public static void main(String[] args){

```



```

Scanner in = new Scanner(System.in);

String str = in.nextLine();

List<Integer> li= new ArrayList<Integer>();

int count=0;    for(char i: str.toCharArray()){

    if(i!='1'){

count++;

    }    else{

li.add(count);

count=0;

    }    }    for (int i : li) {

if (i >= 1 && i <= 26) {        char

letter = (char) ('Z' - (i - 1));

System.out.print(letter);

    }

}

}

```

OUTPUT:

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	0000100000000000000000001000000000001000000000100000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

## PROGRAM 12.2:

AIM:

Given two char arrays input1[] and input2[] containing only lower case alphabets, extracts the alphabets which are present in both arrays (common alphabets).

Get the ASCII values of all the extracted alphabets.

CHANDRU K 231501031

Calculate sum of those ASCII values. Lets call it sum1 and calculate single digit sum of sum1, i.e., keep adding the digits of sum1 until you arrive at a single digit.

Return that single digit as output.

CODE:

```
import java.util.*; public class Main {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        int sum = 0, rem = 0, sum1 = 0;

        String s1 = in.nextLine();

        String s2 = in.nextLine();        for (int i
= 0; i < s1.length(); i++) {            for
(int j = 0; j < s2.length(); j++) {
        if (s1.charAt(i) == s2.charAt(j) &&
s1.charAt(i) != ' ') {                sum
+= (int) s1.charAt(i);

            }

        }

    }

    String str = Integer.toString(sum);

    while (str.length() > 1) {        sum1
= 0;        int num =
Integer.parseInt(str);        while
(num != 0) {            rem = num %
10;            sum1 += rem;

            num /= 10;

        }

    }
```

```

        str = Integer.toString(sum1);
    }

    System.out.println(sum1);
}
}

```

OUTPUT:

	Input	Expected	Got	
✓	a b c b c	8	8	✓

Passed all tests! ✓

PROGRAM 12.3:

AIM:

Write a function that takes an input String (sentence) and generates a new String (modified sentence) by reversing the words in the original String, maintaining the words position.

CODE:

```
import java.util.Scanner;
```

```
public class ReverseWords {
```

```
    public static String reverseSentence(String sentence, int caseOption) {
```

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

```
        String[] reversedWords = new String[words.length];
```

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

```
            reversedWords[i] = reverseWord(words[i]);
```

```
        }
```

```

String reversedSentence = String.join(" ", reversedWords);

    if (caseOption == 1) {        reversedSentence =
applyCaseOption1(sentence, reversedSentence);
    }

    return reversedSentence;
}

private static String reverseWord(String word) {
    StringBuilder reversed = new StringBuilder();
    for (int i = word.length() - 1; i >= 0; i--) {
        reversed.append(word.charAt(i));
    }
    return reversed.toString();
}

private static String applyCaseOption1(String original, String reversed) {    StringBuilder
result = new StringBuilder(reversed);    for (int i = 0; i < original.length(); i++) {
    char originalChar = original.charAt(i);        char reversedChar = result.charAt(i);
    if (Character.isUpperCase(originalChar) && Character.isLetter(reversedChar)) {
        result.setCharAt(i, Character.toUpperCase(reversedChar));
    } else if (Character.isLowerCase(originalChar) && Character.isLetter(reversedChar)) {
        result.setCharAt(i, Character.toLowerCase(reversedChar));
    }
}

    return result.toString();
}

```

```

    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String inputSentence = scanner.nextLine();        int
        caseOption = scanner.nextInt();        while
        (caseOption != 0 && caseOption != 1) {
            caseOption = scanner.nextInt();

        }

        String result = reverseSentence(inputSentence, caseOption);
        System.out.println(result);        scanner.close();
    }
}

```

OUTPUT:

	Input	Expected	Got	
✓	Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB 0	orpiW seigolonhceT erolagnaB 0	✓
✓	Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB 0	orpiW ,seigolonhceT erolagnaB 0	✓
✓	Wipro Technologies Bangalore 1	Orpiw SeigolonhceT Erolagnab 1	Orpiw SeigolonhceT Erolagnab 1	✓
✓	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab 1	Orpiw ,seigolonhceT Erolagnab 1	✓

Passed all tests! ✓